This is the post peer-review accepted manuscript of:

F. Conti, R. Schilling, P. D. Schiavone, A. Pullini, D. Rossi, F. K. Gürkaynak, M. Muehlberghuber, M. Gautschi, I. Loi, G. Haugou, S. Mangard and L. Benini, "An IoT Endpoint System-on-Chip for Secure and Energy-Efficient Near-Sensor Analytics," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 9, pp. 2481-2494, Sept. 2017. doi: 10.1109/TCSI.2017.2698019

The published version is available online at: https://doi.org/10.1109/TCSI.2017.2698019

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

An IoT Endpoint System-on-Chip for Secure and Energy-Efficient Near-Sensor Analytics

Francesco Conti, Robert Schilling, Pasquale D. Schiavone, Antonio Pullini, Davide Rossi, Frank K. Gürkaynak, Michael Muehlberghuber, Michael Gautschi, Igor Loi, Germain Haugou, Stefan Mangard, Luca Benini

***Abstract*—Near-sensor data analytics is a promising direction for IoT endpoints, as it minimizes energy spent on communication and reduces network load - but it also poses security concerns, as valuable data is stored or sent over the network at various stages of the analytics pipeline. Using encryption to protect sensitive data at the boundary of the on-chip analytics engine is a way to address data security issues. To cope with the combined workload of analytics and encryption in a tight power envelope, we propose *Fulmine*, a System-on-Chip based on a tightly-coupled multi-core cluster augmented with specialized blocks for compute-intensive data processing and encryption functions, supporting software programmability for regular computing tasks. The *Fulmine* SoC, fabricated in 65 nm technology, consumes less than 20 mW on average at 0.8 V achieving an efficiency of up to 70 pJ/B in encryption, 50 pJ/px in convolution, or up to 25 MIPS/mW in software. As a strong argument for real-life flexible application of our platform, we show experimental results for three secure analytics use cases: secure autonomous aerial surveillance with a state-of-the-art deep CNN consuming 3.16 pJ per equivalent RISC op; local CNN-based face detection with secured remote recognition in 5.74 pJ/op; and seizure detection with encrypted data collection from EEG within 12.7 pJ/op.**

I. Introduction

The key driver for the development of the Internet-of-Things (IoT) is collecting rich and diverse information streams from sensors, which can then be fed to state-of-the-art learning-based data analytics algorithms. The information distilled by data analytics on such a rich input set can be used in a virtually unlimited set of applications, such as healthcare or home automation, which have the possibility to change the life of any person for the better [1]. However, in practice, the possibility to seamlessly tap into this rich stream of data is limited by two equally important factors. First, the amount of data an IoT end-node can extract from sensors and send over the network for analytics is essentially defined by the energy necessary for data transfer itself. Since IoT end-nodes must work within a tiny power envelope, this fact introduces a significant limit on the volume of data that can be transferred, e.g. the size of captured images, therefore curtailing their usefulness. Second, due to the ubiquitous nature of IoT devices, they often deal with private or safety critical input data even beyond the predictions of their designers; not only devices such as healthcare wearables acquire potentially safety-critical data, but also seemingly innocuous devices (such as cameras) can potentially acquire highly sensitive information [2]. To ensure practicality of IoT-based applications, it is imperative that data transmission from end-nodes to the network is protected from data theft or malicious tampering.

To address the first limiting factor, near-sensor smart data analytics is a promising direction; IoT end-nodes must evolve from simple data collectors and brokers into analytics devices, able to perform a pre-selection of potentially interesting data and/or to transform it into a more abstract, higher *information density* form such as a classification tag. With the burden of *sensemaking* partially shifted from centralized servers to distributed end-nodes, the energy spent on communication and the network load can be minimized effectively and more information can be extracted, making the IoT truly scalable. However, performing analytics such as feature extraction or classification directly on end-nodes does not address the security concerns. It worsens them: *distilled* data that is stored or sent over the network at several stages of the analytics pipeline is even more privacy-sensitive than the raw data stream [3], [4]. Protecting sensitive data at the boundary of the on-chip analytics engine is a way to address these security issues; however, cryptographic algorithms come with a significant workload, which can easily be of 100-1000s of processor instructions per encrypted byte [5].

This security workload is added to the computational effort imposed by leading feature extraction and classification algorithms, such as deep Convolutional Neural Networks (CNNs). CNNs are extremely powerful in terms of data analytics, and state-of-the-art results in fields such as computer vision (e.g. object detection [6], scene parsing [7], and semantic segmentation tasks [8]) and audio signal analytics [9] have been demonstrated. While effective, deep CNNs usually necessitate many billions of multiply-accumulate operations, as well as storage of millions of bytes of pre-trained *weights* [10]. The combined workload necessary to tackle these two limitations to the development of smarter IoT - namely, the necessity for *near-sensor analytics* and that for *security* - is formidable, especially under the limited available power envelope and the tight memory and computational constraints of deeply embedded devices. One possible solution is to augment IoT end-nodes with specialized blocks for compute-intensive data processing and encryption functions while retaining full software programmability to cope with lower computational-intensity tasks. Specialized processing engines should be tightly integrated both with the

Francesco Conti, Pasquale Schiavone, Antonio Pullini, Frank Gürkaynak, Michael Muehlberghuber, Michael Gautschi, Germain Haugou and Luca Benini are with ETH Zurich, Switzerland. Robert Schilling and Stefan Mangard are with the Graz University of Technology, Austria. Davide Rossi, Igor Loi, Francesco Conti and Luca Benini are also with the University of Bologna, Italy. Robert Schilling is also with the Know-Center, Graz, Austria.

This work was supported by EU projects MULTITHERMAN (FP7-ERC-291125), ExaNoDe (H2020-671578), OPRECOMP (H2020-732631), and SOPHIA (H2020-ERC-681402).

software-programmable cores and with one another, streamlining the process of data exchange between the different actors as much as possible to minimize the time and energy spent in data exchange; at the same time, to simplify their usage from the developer's perspective, it should be possible to abstract them, integrating them in standard programming models used in software development for IoT-aware platforms.

In this work, we propose the $65\,\mathrm{nm}$ *Fulmine secure data analytics* System-on-Chip (SoC), which tackles the two main limiting factors of IoT end-nodes while providing full programmability, low-effort data exchange between processing engines, (sufficiently) high speed, and low energy. The SoC is based on the architectural paradigm of tightly-coupled heterogeneous shared-memory clusters [11], where several engines (which can be either programmable cores or specialized hardware accelerators) share the same first-level scratchpad via a low-latency interconnect. In *Fulmine*, the engines are four enhanced 32-bit OpenRISC cores, one highly efficient cryptographic engine for AES-128 and KECCAK-based encryption, and one multi-precision convolution engine specialized for CNN computations. Due to their memory sharing mechanism, cores and accelerators can exchange data in a flexible and efficient way, removing the need for continuous copies between cores and accelerators. The proposed SoC performs computationally intensive data analytics workloads with no compromise in terms of security and privacy, thanks to the embedded encryption engine. At the same time, *Fulmine* executes full complex pipelines including CNN-based analytics, encryption, and other arbitrary tasks executed on the processors.

This claim is exemplified in three practical use cases: secure autonomous aerial surveillance in a nano-Unmanned Aerial Vehicle (nano-UAV) consuming $3.16\,\mathrm{pJ}$ per equivalent RISC operation; on-device CNN-based face detection (as part of a recognition pipeline) with $5.74\,\mathrm{pJ}$ per operation, including image encryption for external face recognition; and seizure detection with secure data collection within $12.7\,\mathrm{pJ}$ per operation. We show that on a workload consisting of balanced contributions from CNNs, AES, and other SW-implementable filters, *Fulmine* provides the best result in terms of pJ-per-equivalent-RISC-operation, with the nearest state-of-the-art platform in terms of efficiency needing $89\times$ more time to execute the workload.

The rest of this paper is organized as follows: in Section II we compare *Fulmine* with the state-of-the-art in low-power IoT computing devices. Section III describes the architecture of the SoC; cluster-coupled HW coprocessors are detailed in Section IV. Section V evaluates the implementation results and overall performance, while Section VI focuses on real-world use cases. Section VII concludes the paper.

II. State-of-the-Art and Related Work

A. Low-Power Encryption Hardware IPs

Authenticated encryption is a hot topic in the cryptographic community since it adds additional services on top of data confidentiality. AES in the Galois Counter Mode [12] (AES-GCM) is one of the most used authenticated encryption schemes today. For example, Intel added a dedicated finite field multiplication to the AES-NI extension, with a throughput up to 1.03 cpb [13]. However, solutions of this kind are clearly targeting a different scenario from small, low-power IoT devices.

Only a few IoT-oriented commercial AES controllers are available; an example is the Maxim MAXQ1061 [14], claiming up to 20 Mbit/s (power consumption data is not currently disclosed). Research AES accelerators in the sub-100 mW range for the IoT domain have been proposed by Mathew et al. [15] in Intel 22nm technology, Zhang et al. [16] in TSMC 40 nm and Zhao et al. [17] in 65 nm; the latter reaches efficiency up to 620 Gbit/s/W thanks to efficient body biasing and a statistical design flow targeted at reducing worst-case guard bands. A device consuming as little as $0.25\,\mu\mathrm{W}$ for passive RFID encryption has been proposed by Hocquet et al. [18]. The main differentiating point between our contribution and these hardware encryption techniques is the tightly coupled integration within a bigger low-power system.

B. Low-Power CNN Hardware IPs

The most common way to accelerate CNNs is to rely on powerful GP-GPUs [7][19] or on FPGAs [20][21]. Some programmable embedded platforms such as ODROID-XU [22], or Movidius Myriad 2 [23] improve the energy efficiency of software CNN implementations to up to 120 Gop/s/W within a few Watts of power envelope, targeting embedded systems such as smartphones or UAVs as well as the booming autonomous car business [24]. While these platforms are typically powerful enough for embedded scenarios that are not significantly power-constrained (e.g. deep-learning driven autonomous driving), we do not consider them directly comparable to our proposal, since they cannot be used in low-power endnodes: their efficiency is relatively low (up to tens of GMAC/s/W for most GPU and FPGA implementations) and their peak power envelope is typically too high, up to $\sim 10\,\mathrm{W}$ - $100\times$ the typical envelope considered for endnodes. To the best of our knowledge, the only deep neural network commercial solution specifically designed for IoT end-nodes is WiseEye, to be presented by CEVA at CES 2017 [25].

Most research architectures for acceleration of CNNs have focused on specialized architectures to accelerate convolutional layers (e.g. Origami [26]), or convolutional and pooling layers (e.g. ShiDianNao [27] and Eyeriss [28]). These accelerators reach efficiencies in the order of a few hundreds of equivalent Gop/s/W. However, they all rely on highly specialized architectures, their flexibility is limited, and most of them are not capable of implementing the other functionality required by IoT end-nodes, including security and general-purpose signal processing tasks.

One big differentiating point between these platforms are their assumptions in terms of algorithmic and arithmetic accuracy. Jaehyeong et al. [29] rely on 24bit fixed-point arithmetics, but they approximate weights using a low-dimensional representation based on PCA. Most other works use either 16 bits [30], [31] or 12 bits [26]. However, recent algorithmic developments such as BinaryConnect [32] suggest that it is

possible to reduce CNN weight precision down to a single bit with limited accuracy losses. This has been exploited in platforms such as YodaNN [33] to reach efficiency in the order of tens of equivalent Top/s/W. Another promising approach to improve energy efficiency in classification tasks are extreme learning machines (ELM), based on single-hidden layer feedforward neural networks. Although they have been proven to consume as little as 0.47 pJ/MAC [34][35], their applicability to real-life applications is still restricted to very simple problems.

In this work a flexible approach has been adopted, where the precision of images is fixed to 16 bits, while that of weights can be scaled from 16 to 4 bits. This approach avoids the requirement of specific training for binary connected networks, while weight precision can be scaled according to the specific application requirements in terms of accuracy, throughput and energy efficiency.

C. End-Node Architectures

Traditional end-node architectures for the IoT leverage tiny microprocessors, often Cortex-M0 class, to deal with the extreme low-power consumption requirements of applications. Several commercial solutions have been proposed, among the others, by TI [36], STMicroelectronics [37], NXP [38], and Ambiq [39], leveraging aggressive duty-cycling and sub-$10\,\mu W$ deep-sleep modes to provide extremely low power consumption on average. Other recent research platforms optimize also the active state, exploiting near-threshold or sub-threshold operation to improve energy efficiency and reduce power consumption during computation [40][41][42][43].

Some commercial architectures leverage lightweight SW acceleration and optimized DSP libraries to improve performance. The NXP LPC54100 [38] is a commercial platform where a *big* Cortex-M4F core acts as an accelerator for a *little* ultra-low-power Cortex-M0 targeted at always-on applications. From a software viewpoint, some optimized libraries have been developed to efficiently implement crypto algorithms on Cortex-M3 and M4 architectures, given the criticality of this task for IoT applications. Examples of these libraries are SharkSSL [44] and FELICS [5], able to encrypt one block of AES-128-ECB in 1066 cycles and 1816 cycles respectively, both targeting a Cortex-M3. On the other hand, CMSIS [45] is a well-known set of libraries to optimize DSP performance on Cortex-M architectures.

However, even with software-optimized libraries, these tiny micro-controllers are unfortunately not suitable for secure near-sensor analytics applications using state-of-the-art techniques, which typically involve workloads in the orders of billions of operations per second. For this reason, a few recent SoCs couple programmable processors with hardwired accelerators, to improve execution speed and energy efficiency in cryptography and other performance-critical tasks. In the field of embedded vison, heterogeneous SoCs of this kind include the one recently proposed by Renesas [46], coupling a general purpose processor with an FPU, a DSP, and a signal processing accelerator. Intel [47] proposed a 14 nm SoC where a small core with light signal processing acceleration

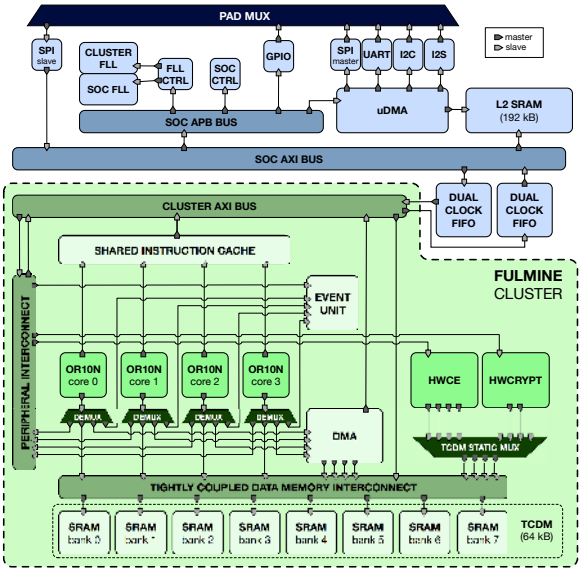


Fig. 1: *Fulmine* SoC architecture. The SOC domain is shown in shades of blue, the CLUSTER domain in shades of green.

cooperates with a vision processing engine for CNN-based feature extraction and a light encryption engine, within a $22\,\mathrm{mW}$ power budget. Pullini et al. proposed Mia Wallace, a heterogeneous SoC [48] coupling four general purpose processors with a convolutional accelerator. In the field of bio-signals processing, Konijnenburg et al. [49] proposed a multichannel acquisition system for biosensors, integrating a Cortex-M0 processor and accelerators for digital filtering, sample rate conversion, and sensor timestamping. Lee et al. [50] presented a custom bio-signals processor that integrates configurable accelerators for discriminative machine-learning functions (i.e. SVM and active learning) improving energy by up to 145x over execution on CPU.

Similarly to the presented designs, *Fulmine* is a low-power, heterogeneous MPSoC. In contrast to the other architectures presented here, it tackles at the architectural level the challenge of efficient and secure data analytics for IoT end-nodes, while also providing full programmability with sufficient high performance and low power to sustain the requirements of several near-sensor processing applications.

III. SoC Architecture

The *Fulmine* multi-core System-on-Chip (Figure 1) implements a secure near-sensor data analytics architecture, which leverages highly efficient processors for software programmable signal processing and control, flexible hardware acceleration for cryptographic functions, convolutional neural networks, and a highly optimized subsystem implementing power management and efficient communication and synchronization among cluster resources. The architecture, based on the PULP platform [51], is organized in two distinct voltage and frequency domains, CLUSTER and SOC, communicating through an AXI4 interconnect and separated by dual-clock FIFOs and level shifters. Two frequency-locked loops (FLLs) are used to generate clocks for the two domains, which rely on separate external voltage regulators for their supply and can be

independently power-gated. The FLLs work with a 100 kHz external reference clock and support fast switching between different operating modes (less than 10 reference cycles in the worst case).

The CLUSTER domain is built around six *processing elements* (four general-purpose processors and two flexible accelerators) that share 64 kB of level 1 Tightly-Coupled Data Memory (TCDM), organized in eight word-interleaved SRAM banks. A low-latency logarithmic interconnect [52] connects all processing elements to the TCDM memory, enabling fast and efficient communication among the resources of the cluster. The TCDM interconnect supports single-cycle access from multiple processing elements to the TCDM banks; if two masters attempt to access the same bank in the same clock cycle, one of them is stalled using a starvation-free round-robin arbitration policy. The two hardware accelerators, *Hardware Cryptography Engine* (HWCRYPT) and *Hardware Convolution Engine* (HWCE), can directly access the same TCDM used by the cores. This architecture allows data to be seamlessly exchanged between cores and accelerators, without requiring explicit copies and/or point-to-point connections. To avoid a dramatic increase in the area of the TCDM interconnect, as well as to keep the maximum power envelope in check, the two accelerators share the same set of four physical ports on the interconnect. The two accelerators are used in a time-interleaved fashion, allowing one accelerator full access to the TCDM at a time, which is suitable for data analytics applications where computation can be divided into several separate stages.

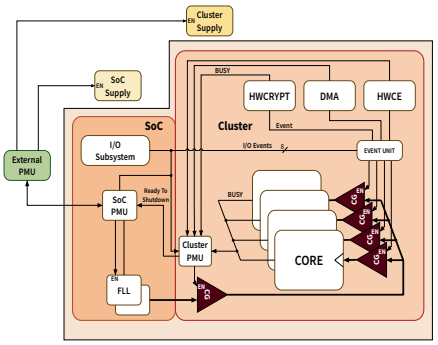
The four OR10N cores are based on an in-order, single-issue, four stage pipeline, implementing the OpenRISC [53] instruction set architecture (ISA), improved with extensions for higher throughput and energy efficiency in parallel signal processing workloads [54]. GCC 4.9 and LLVM 3.7 toolchains are available for the cores, while OpenMP 3.0 is supported on top of the bare-metal parallel runtime. The cores share a single instruction cache of 4 kB of Standard Cell Memory (SCM) [55] that can increase energy efficiency by up to 30% compared to an SRAM-based private instruction cache on parallel workloads [56]. The ISA extensions of the core include general-purpose enhancements (automatically inferred by the compiler), such as zero-overhead hardware loops and load and store operations embedding pointer arithmetic, and other DSP extensions that can be explicitly included by means of *intrinsic* calls. For example, to increase the number of effective operations per cycle, the core includes single instruction multiple data (SIMD) instructions working on 8 bit and 16 bit data, which exploit 32 bit registers as vectors. Furthermore, the core is enhanced with a native dot-product instruction to accelerate computation-intensive classification and signal-processing algorithms. This single-cycle operation supports both 8 bit and 16 bit vectors using two separate datapaths to reduce the timing pressure on the critical path. Fixed point numbers are often used for embedded analytics and signal processing applications; for this reason, the core has also been extended with single-cycle fixed point instructions including rounded additions, subtractions, multiplications with normalization, and clipping instructions.

The cluster features a set of peripherals including a direct memory access (DMA) engine, an event unit, and a timer. The processors can access the control registers of the hardware accelerators and of the other peripherals through a memory mapped interface implemented as a set of private, per-core demultiplexers (DEMUX), and a peripheral interconnect shared among all cores. The peripheral interconnect implements the same architecture of the TCDM interconnect, featuring a different addressing scheme to provide 4 kB of address map for each peripheral.

The DMA controller available in the cluster is an evolution of the one presented in [57], and enables fast and flexible communication between the TCDM and the L2 memory trough four dedicated ports on the TCDM interconnect and an AXI4 plug on the cluster bus. In contrast to traditional memory mapped interfaces, access to the internal DMA programming registers is implemented through a sequence of control words sent to the same address, significantly reducing DMA programming overheads (i.e. less then 10 cycles to initiate a transfer, on average). The DMA supports up to 16 outstanding 1D or 2D transfers to hide L2 memory latency and allows 256 byte bursts on the 64-bit AXI4 interface to guarantee high bandwidth. Once a transfer is completed, the DMA generates an event to the cores that can independently synchronize on any of the enqueued transfers by checking the related transfer ID on the DMA control registers. Synchronization of DMA transfers and hardware accelerated tasks is hardware-assisted by the event unit. The event unit can also be used to accelerate the typical parallelization patterns of the OpenMP programming model, requiring, for example, only 2 cycles to implement a *barrier*, 8 cycles to open a *critical section*, and 70 cycles to open a *parallel section*. These features are all essential to guarantee high computational efficiency during execution of complex tasks such as CNNs in *Fulmine*, as detailed in Section IV.

The SOC domain contains 192 kB of L2 memory for data and instructions, a 4 kB ROM, a set of peripherals, and a power management unit. Furthermore, the SOC includes a (quad) SPI master, I2C, I2S, UART, GPIOs, a JTAG port for debug, and a (quad) SPI slave that can be used to access all the SoC internal resources. An I/O DMA subsystem (uDMA) allows to autonomously copy data between the L2 memory and the external interfaces, even when the cluster is in sleep mode. This mechanism allows us to relieve cores from the frequent control of peripherals necessary in many microcontrollers, and to implement a double buffering mechanism both between IOs and L2 memory and between L2 memory and TCDM. Therefore, I/O transfers, L2 memory to TCDM transfers, and computation phases can be fully overlapped.

A sophisticated power management architecture distributed between the SOC and CLUSTER domains can completely clock-gate all the resources when idle, as shown in Figure 2 (*idle* mode with FLL on). The power manager can also be programmed to put the system in a low power retentive state by switching down the FLLs and relying on the low-frequency reference clock (*low freq* and *idle* mode). Finally, it can be used to program the external DC/DC converter to fully power-gate the CLUSTER domain. The event unit is responsible for



CLUSTER @0.8V	CLOCK	WAKEUP	POWER
ACTIVE	50 MHz	-	~10 mW
LOW-FREQ (FLL off)	100 kHz	300 us	230 uW
IDLE (FLL on)	-	20 ns	600 uW
IDLE (FLL off)	-	300 us	210 uW
DEEP SLEEP	-	DC/DC[1]	~10 nW[2]

SOC @0.8V	CLOCK	WAKEUP	POWER
ACTIVE	50 MHz	-	~2 mW
LOW-FREQ (FLL off)	100 kHz	300 us	130 uW
IDLE (FLL on)	-	20 ns	510 uW
IDLE (FLL off)	-	300 us	120 uW
DEEP SLEEP	-	300 us	120 uW

[1]Depends on DC/DC settling time.
[2]Cluster is power-gated in **DEEP SLEEP** mode.

Fig. 2: *Fulmine* power management architecture and power modes.

automatically managing the transitions of the cores between the active and idle state. To execute a *wait-for-event* instruction, the cores try to read a special register in the event unit; this load is kept stalled until the event comes so that the core pipeline is stalled in a known state. After pending transactions and cache refills are complete, the event unit gates the core clock. The clock gating manager gates the cluster clock if the *idle* mode is selected and no engine is busy, or it activates the handshaking mechanism with the external regulator to power gate the cluster if the *deep-sleep* mode is selected. Once the wake-up event reaches the power management unit, the latter reactivates the cluster, then it forwards the event notification to the event unit, waking up the destination core on turn. Figure 2 reports all the power modes along with their average wakeup time and power consumption, divided between the CLUSTER and SOC domains. As the CLUSTER and SOC power domains are managed independently, it is possible to transparently put the CLUSTER in *idle*, where it consumes less than $1\,\mathrm{mW}$, when waiting for an event such as the end of an I/O transfer to L2 or an external interrupt that is expected to arrive often. It is possible to partially trade off wakeup time versus power by deciding whether to keep the FLLs active in *idle* mode: by paying a $\sim 400\,\mu\mathrm{W}$ cost, wakeup time is reduced to essentially a single clock cycle ($20\,\mathrm{ns}$), versus a maximum of 10 reference cycles ($\sim 320\,\mu\mathrm{s}$) if the FLL is off. The *deep sleep* mode instead enables efficient duty cycling in the case computing bursts are relatively rare, by completing power-gating the CLUSTER domain and keeping the SOC domain in a clock-gated, retentive state.

IV. Cluster-coupled accelerator engines

In this Section we describe in detail the architecture of the two cluster-coupled accelerator engines, HWCRYPT and HWCE. The main purpose of these engines is to provide a performance and efficiency boost on computations, and they were designed to minimize active power, e.g. by using aggressive clock gating on time-multiplexed sub-modules and by making use of latches in place of regular flip-flops to implement most of the internal buffering stages.

The shared-memory nature of the HWCRYPT and HWCE accelerators enables efficient zero-copy data exchange with the cores and the DMA engine, orchestrated by the cluster event unit. This architecture enables complex computation patterns with frequeny transfers of data set tiles from/to memory. A typical application running on the *Fulmine* SoC operates conceptually in the following way. First, the input set (e.g. a camera frame) is loaded into the L2 memory from an external I/O interface using the uDMA. The cluster can be left in sleep mode during this phase and woken up only at its conclusion. The input set is then divided into tiles of appropriate dimension so that they can fit in the L1 shared TCDM; one tile is loaded into the cluster, where a set of operations are applied to it either by the SW cores or the HW accelerators. These operations can include en-/decryption and convolutions (in HW), plus any SW-implementable filter. The output tiles are then stored back to L2 memory using DMA transfers, and computation continues with the next tile. Operations such as DMA transfers can typically be overlapped with computation by using double buffering to reduce the overall execution time.

A. Hardware Encryption Engine

The Hardware Encryption Engine (HWCRYPT), as shown in Figure 3, implements a dedicated acceleration unit for a variety of cryptographic primitive operations, exploiting the advantages of the shared memory architecture of the SoC. The HWCRYPT is based on two parallel cryptographic engines, one implementing the AES-128 [58] block cipher and the other one implementing the KECCAK-f[400] [59] permutation (a smaller version of the SHA-3 permutation) used in a flexible sponge construction. The AES-128 engine includes two instances of a round-based AES-128 design with a shared on-the-fly round-key computation module. Each of the two AES-128 instances is based on two cipher rounds supporting both encryption and decryption. The round-key generator keeps track of the last round-key during encryption operations, which acts as the starting point to generate round-keys for a decryption operation. The AES-128 engine of the HWCRYPT implements the Electronic-Code-Book (ECB) mode as well as the XEX-based tweaked-codebook mode with ciphertext stealing (XTS) [60]. XTS uses two different encryption keys, one to derive the initial tweak and the other one to encrypt the data. When using the same key for deriving the initial tweak

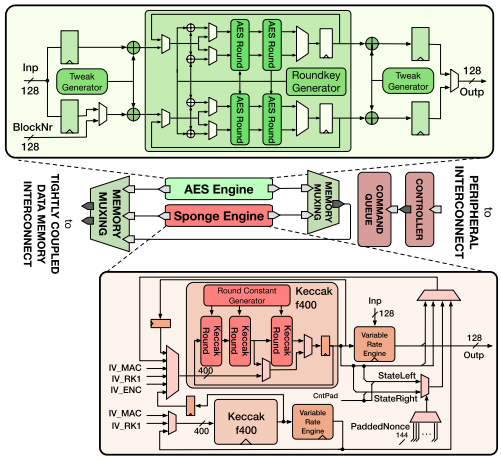


Fig. 3: HWCRYPT datapath overview, with details of the AES-128 and the sponge engine.

and encrypting the data, the encryption scheme is changed to XEX [61] without implications to the overall security. Furthermore, the accelerator supports the individual execution of a cipher round similar to the Intel AES-NI instructions [62] to boost the software performance of other new AES round-based algorithms [63], [64].

Although AES-128-ECB is a fast encryption mode, it is not recommended to use it to encrypt larger blocks of data. Since every block is encrypted independently using the same key, the same plaintext always yields the same ciphertext, which reveals patterns of the plaintext in the ciphertext. To overcome this issue, the AES-128-XTS mode uses a so-called tweak T to modify the encryption for each block. The encryption function E in the block diagram denotes an AES-128-ECB encryption with the key K_1 and K_2 respectively. The tweak is XORed to the plaintext and to the resulting ciphertext of the AES-128-ECB encryption. Since the tweak is different for each block of plaintext, it is denoted as T_i for the i-th block of data. The initial tweak T_0 is computed by encrypting the sector number SN, derived from the address of the data, using the encryption key K_1 and multiplying it with α^i with $i = 0$. The multiplication with α^i ensures that the tweak is different for each block. The XTS mode is defined by Equation 1:

$$\begin{aligned} T_i &= E_{K_1}(SN) \otimes \alpha^i \\ C_i &= E_{K_2}(P_i \oplus T_i) \oplus T_i \end{aligned} \tag{1}$$

The address-dependent tweak T_i is derived by a multiplication between the initial tweak and α^i. The multiplication is performed in the finite Galois field[1] $\mathrm{GF}(2^{128})$ defined by the irreducible polynomial $x^{128} + x^7 + x^2 + x + 1$. AES-128-XTS requires a 128-bit finite field multiplier and exponentiator, which is rather complex in terms of VLSI implementation. To reduce this complexity, we first observe that α is constant with the recommended value $\alpha = 2$. T_i is derived from T_{i-1} as a one-bit shift with a conditional XOR with the irreducible polynomial, i.e. $T_i = T_{i-1} \otimes 2$, which implements the multiplication by 2 in the Galois field $GF(2^{128})$.

The sponge engine implements two instances of the KECCAK-f[400] permutation, each based on three permutation rounds. KECCAK-f[400]'s architecture is optimized to match the length of the critical path of the AES-128 engine. Permutations support a flexible configuration of the rate and round parameters. The rate defines how many bits are processed within one permutation operation, and it can be configured from 1 bit to 128 bits in powers of two. This parameter supports a trade-off between security and throughput. The more bits are processed in one permutation call, the higher the throughput - but with a cost regarding the security margin of the permutation. The round parameter configures the number of KECCAK-f[400] rounds applied to the internal state. It can be set up as a multiple of three or for 20 rounds as defined by the specification of KECCAK-f[400]. The two instances of permutations are combined to implement an authenticated encryption scheme based on a sponge construction with a prefix message authentication code that additionally provides integrity and authenticity on top of confidentiality. In the sponge construction for encryption, the initial state of the sponge is filled with the key K and the initial vector IV. After executing the KECCAK-f[400] permutation p, we sequentially squeeze an encryption pad and apply the permutation function to encrypt all plaintext blocks P_i via an XOR operation. Apart from this favorable mode of operation, the sponge engine also provides encryption without authentication and direct access to the permutations to allow the software to accelerate any KECCAK-f[400]-based algorithm.

The HWCRYPT utilizes two 32 bit memory ports of the TCDM interconnect, while an internal interface performs the conversion from 32 bit to the 128 bit format used by the encryption engines. The system is designed so that memory interface bandwidth matches the requirements of all cipher engines. The HWCRYPT is programmed and started through a dedicated set of configuration registers, which allows the reconfiguration of a new encryption operation while the HWCRYPT is busy by using a command queue that supports up to four pending operations. The current state of the HWCRYPT can be monitored via status registers. The accelerator supports a flexible event and interrupt system to indicate when one or all operations have finished.

B. Hardware Convolution Engine

The Hardware Convolution Engine (HWCE) is based on a precision-scalable extension of the design proposed by Conti and Benini [30] to accelerate convolutional layers in deep CNNs. These layers, which constitute the overwhelming majority of computation time in most CNNs [7], are composed of a linear transformation that maps N_{if} input feature maps into N_{of} output feature maps by means of a set of convolutional filters; and a pointwise non-linear activation function, often a rectifier (ReLU) . The linear part of convolutional layers is usually the dominant operation by far; its typical form for $k_{\mathrm{of}} \in 0 \cdots N_{\mathrm{of}} - 1$ is the following:

$$\mathbf{y}(k_{\mathrm{of}}) = \mathbf{b}(k_{\mathrm{of}}) + \sum_{k_{\mathrm{if}}=0}^{N_{\mathrm{if}}-1} \Big(\mathbf{W}(k_{\mathrm{of}}, k_{\mathrm{if}}) * \mathbf{x}(k_{\mathrm{if}}) \Big). \tag{2}$$

The specific task executed by the HWCE is the acceleration of the *accumulation of convolutions* that are at the core of Equation 2. To represent input/output pixels and weights, a fixed-point data representation with 16 bits is used by default. The number of fractional bits is configurable at run time. HWCE can natively perform 5×5 and 3×3 convolutions, and any arbitrary convolution by combining these two in software. The key novelty in the HWCE design with respect to [30] is that the *Fulmine* HWCE can exploit the relative insensitivity of CNNs to weight approximation [65][32] by reducing the arithmetic precision of the convolution weights to 8 or 4 bit. In that case, the internal datapath is reconfigured so that two or four convolutions respectively (on different output k_{of} feature maps) are computed simultaneously, while feature map pixels still use the full 16 bit representation. In these scaled precision modes, a similar level of accuracy to the 16 bit full precision CNNs can be maintained by proper training, with access to

[1] In the following, $\otimes$ denotes the 128-bit finite field multiplication in which also the exponentiation is performed.

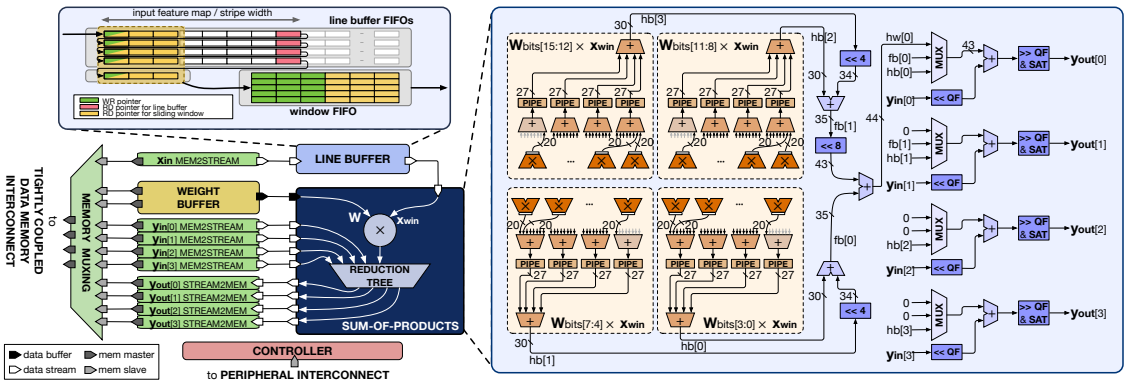


Fig. 4: *Fulmine* HWCE architecture, with the *controller* shaded in red, the *wrapper* in green, and the *datapath* in blue. The diagram also shows details of the line buffer and sum-of-products submodules microarchitecture.

significantly improved performance, memory footprint, and energy efficiency as is shown in Section V.

Figure 4 depicts the HWCE architecture, which can be divided into three main components: a *datapath* performing the main part of the data plane computation in a purely streaming fashion, relying on an AXIStream-like handshake for back-pressure; a *wrapper* that connects and decouples the datapath streaming domain from the memory-based cluster; and a *controller* that provides a control interface for the accelerator. In the full-precision 16 bit mode, the sum-of-products datapath is used to perform a convolution between a preloaded filter $\mathbf{W}$ (stored in a weight buffer) and a 5×5 $\mathbf{x}_{\text{win}}$ window extracted from a linear $\mathbf{x}$ input feature map stream. Window extraction is performed by a line buffer, which is realized with latch-based SCMs for optimized energy efficiency. The line buffer is composed by a set of FIFO queues with two read pointers: one to implement the mechanism to pass the oldest pixel to the next FIFO and the other to extract the 5×5 sliding window. The output of the sum-of-products is summed to an input pre-accumulated $\mathbf{y}_{\text{in}}$ value; in other words, the accelerator needs no internal memory to perform the feature map accumulation component of Equation 2 but uses directly the shared memory of the cluster. The wrapper, shaded in green in Figure 4 is responsible for generating memory accesses through four memory ports to the TCDM to feed the accelerator $\mathbf{x}$,$\mathbf{y}_{\text{in}}$ streams and write back $\mathbf{y}_{\text{out}}$ (partial) results. The controller (red in Figure 4) contains a register file which can host a queue of two jobs, each consisting of pointers to $\mathbf{x}$, $\mathbf{W}$, $\mathbf{y}$, strides for the wrapper address generators, and other configuration such as the number of fractional bits to use. The controller is mapped in the cluster peripheral interconnect.

To support three different possible sizes for weights, the HWCE sum-of-products datapath must be able to perform 16 bit × 16 bit products as well as 8 bit × 16 bit and 4 bit × 16 bit ones. The two or four filters hosted in the weight buffer in scaled precision modes are not consecutive, but they are interleaved: in full precision mode a location represents a single 16 bit weight; in the scaled precision modes, it represents two 8 bit or four 4 bit weights. The sum-of-products datapath is designed in a hierarchical way to maximize its reuse between the three configurations. Four submodules (shown in orange in Figure 4) compute the sum-of-products of $\mathbf{x}_{\text{win}}$ with a 4 bit slice of $\mathbf{W}$ each, using a set of signed multipliers and a first-stage reduction tree. A second-stage reduction tree and a set of multiplexers are used to combine these four partial sums-of-products to produce one, two or four concurrent $\mathbf{y}_{\text{out}}$ outputs; fractional part normalization and saturation are also performed at this stage. As multiple accumulations of convolutions are performed concurrently, the $\mathbf{y}_{\text{in}}$ and $\mathbf{y}_{\text{out}}$ streamers are replicated four times. All HWCE blocks are aggressively clock gated so that each component consumes power only when in active use.

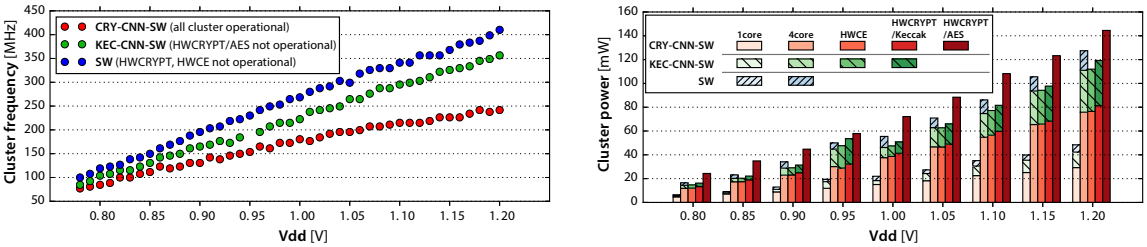
V. Experimental Evaluation

In this Section, we analyze measured performance and efficiency of our platform on the manufactured *Fulmine* prototype chips, fabricated in UMC 65 nm LL 1P8M technology, in a 2.62 mm×2.62 mm die.

A. System-on-Chip Operating Modes

An important constraint for the design of small, deeply embedded systems such as the *Fulmine* SoC is the maximum supported power envelope. This parameter is important to select the system battery and the external DC/DC converter. To maximize energy efficiency, the worst case for the DC/DC converter (i.e. the peak power) should not be too far from the average working power to be delivered. However, a SoC like *Fulmine* can operate in many different conditions: in pure software, with part of the accelerator functionality available, or with both accelerators available. These modes are characterized by very different average switching activities and active power consumption. In pure software mode, it is often desirable to push frequency as much as possible, while when using accelerators it can be convenient to relax it to improve power consumption. Moreover, some of the internal accelerator datapaths are not easily pipelined, as adding pipeline stages severely hits throughput - this is the case of the HWCRYPT sponge engine (Section IV-A), which relies on tight loops of KECCAK-f[400] rounds as visible in the datapath in Figure 3. Relaxing these paths can improve the overall synthesis results for the rest of the circuit.

Multi-corner multi-mode synthesis and place & route were used to define three operating modes that the developer can



(a) Cluster maximum frequency in three operating modes.

(b) Cluster power at maximum frequency in three operating modes.

Fig. 5: Cluster maximum operating frequency and power in the CRY-CNN-SW, KEC-CNN-SW, and SW operating modes. Each set of power bars, from left to right, indicates activity in a different subset of the cluster. KEC-CNN-SW and SW bars show the additional power overhead from running at the higher frequency allowed by these modes.

		Technology	*Area* [mm^2]	*Power*[a] [mW]	*Conv. Perf.*[b] [GMAC/s]	*Conv. Eff.*[b] [GMAC/s/W]	*Enc. Perf.*[c] [Gbit/s]	*Enc. Eff.*[c] [Gbit/s/W]	*SW Perf.* [MIPS]	*SW Eff.* [MIPS/mW]	*Eq. Eff.*[d] [pJ/op]
AES	Mathew et al. [15] @ 0.43V, 324MHz	Intel 22nm	2.74×10^{-3}	0.43	-	-	0.124	289	-	-	0.19[g]
	Zhang et al. [16] @ 0.9V, 1.3GHz	TSMC 40nm	4.29×10^{-3}	4.39	-	-	0.446	113	-	-	0.49[g]
	Zhao et al. [17] @ 0.5V, 34MHz	65nm LL	0.013	0.05	-	-	0.027	574	-	-	0.10[g]
	Hocquet et al. [18] @ 0.36V, 0.32MHz	65nm LP	0.018	2.5×10^{-4}	-	-	3.6×10^{-7}	144	-	-	0.39[g]
CNN	Origami [26] @ 0.8V, 190MHz	UMC 65nm	3.09	93	37	402	-	-	-	-	0.69[g]
	ShiDianNao [27]	65nm	4.86	320	64	200	-	-	-	-	1.39[g]
	Eyeriss [28] @ 1V, 200MHz	TSMC 65nm LP	12.25	278	23	83	-	-	-	-	3.35[g]
	Jaehyeong et al. [29] @ 1.2V, 125MHz	65nm	16.00	45[e]	32	710[e]	-	-	-	-	0.39[g]
	Park et al. [31] @ 1.2V, 200MHz	65nm	10.00	37[f]	41	1108[f]	-	-	-	-	0.25[g]
IoT	SleepWalker [41] @ 0.4V, 25MHz	65nm	0.42	0.175	-	-	-	-	25	143	6.99
	Myers et al. [40] @ 0.4V, 0.7MHz	65nm	3.76	0.008	-	-	-	-	0.7	88	11.4
	Konijnenburg et al. [49] @ 1.2V, 10MHz	180nm	37.7	0.52	-	-	-	-	10.4	20	50.0
	Mia Wallace [48] @ 0.65V, 68MHz	UMC 65nm	7.4	9.2	2.41	261	-	-	270	29	22.5
Fulmine	**CRY-CNN-SW @ 0.8V, 85MHz**			**24**	**4.64**	**309**	**1.78**	**67**	**333**	**14**	
	KEC-CNN-SW @ 0.8V, 104MHz	**UMC 65nm LL**	**6.86**	**13**	**6.35**	**465**	**1.6**	**100**	**408**	**31**	**5.74**
	SW @ 0.8V, 120MHz			**12**	-	-	-	-	**470**	**39**	

[a] Power and efficiency numbers refer to core power, excluding I/Os.
[b] Considering 1 MAC = 2 ops where Gop/s are reported. *Fulmine* numbers refer to the 4bit weights mode.
[c] Refers to AES-128-{ECB,XTS} for *Fulmine* in CRY-CNN-SW; KECCAK-f[400] for *Fulmine* in KEC-CNN-SW; else AES-128-ECB.
[d] Considering the local face detection workload of Section VI-B. 1op = 1 OpenRISC equivalent instruction from the set defined in [53].
[e] Weights produced on-chip from a small set of PCA bases to save area/power. No evaluation on the general validity of this approach is presented in [29].
[f] Performance & power of inference engines only, estimating they are responsible for 20% of total power.
[g] ASIC equivalent efficiency refers to an AES-only or CNN-only equivalent workload.

TABLE I: Comparison between *Fulmine* and several platforms representative of the state-of-the-art in encryption, data analytics, and IoT end-nodes.

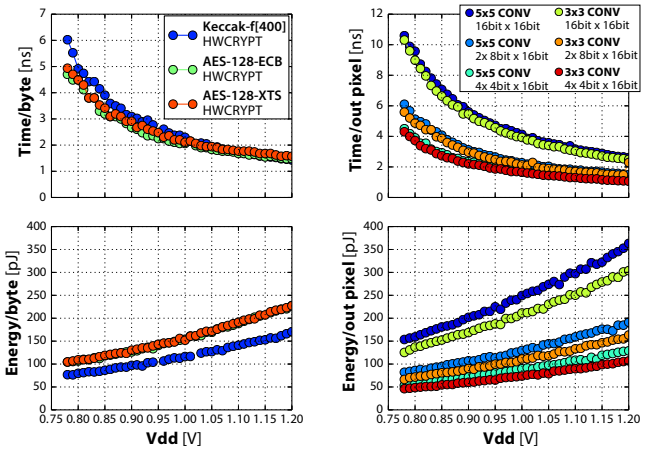
statically select for the target application: in the CRY-CNN-SW mode, all accelerators and cores can be used. In the KEC-CNN-SW mode, cores and part of the accelerators can be used: the HWCE fully, the HWCRYPT limited to KECCAK-f[400] primitives. In this mode, the frequency can be pushed significantly further than in the CRY-CNN-SW mode. Finally, in the SW mode, only the cores are active, and the operating frequency can be maximized. Figure 5 shows frequency scaling in the three operating modes while varying the cluster operating voltage V_{DD}. The three modes were designed so that at V_{DD} =1.2 V, current consumption under full load is close to 100 mA (i.e. 120 mW of power consumption), as can be seen in Figure 5b.

B. HWCRYPT Performance and Power Evaluation

Due to a throughput oriented hardware implementation, HWCRYPT achieves a significant acceleration compared to an optimized software implementation running on the OpenRISC cores. To encrypt one 8 kB block of data using the AES-128-ECB mode, HWCRYPT requires ∼3100 clock cycles including the initial configuration of the accelerator. This is a 450× speedup compared to a software implementation on one core. When parallelizing the software implementation to all four cores, the hardware accelerator still reaches a speedup of 120×. The throughput of HWCRYPT in AES-128-ECB mode is 0.38 cycles per byte (cpb).

The performance of the AES-128-XTS mode is the same with respect to the ECB mode, thanks to parallel tweak computation and encryption. When comparing that to an optimized software implementation on a single core, this speeds up the throughput by a factor of 495× and by a factor 287× when running on four cores. It is important to note that, contrarily to the ECB mode, XTS encryption cannot be efficiently parallelized in software due to a data dependency during the tweak computation step.

The authenticated encryption scheme based on KECCAK-f[400] achieves a throughput of 0.51 cpb by utilizing both permutation instances in parallel. The first permutation encrypts the data and the second one is used to compute the message authentication code to provide integrity and authenticity. This



(a) HWCRYPT time/energy per byte. (b) HWCE time/energy per pixel.

Fig. 6: Performance and efficiency of the HWCRYPT and HWCE accelerators in terms of time/energy for elementary output.

performance is achieved in a maximum-rate configuration of 128 bit per permutation call and 20 rounds as specified by KECCAK-f[400]. Reducing the rate and/or increasing the number of invoked permuations decreases the throughput while increasing the security margin.

In Figure 6a, we present the performance of HWCRYPT in terms of time and energy per byte, while scaling the V_{DD} operating voltage of the cluster. When normalizing these values to the power consumption, we reach a performance of 67 Gbit/s/W for AES-128-XTS and 100 Gbit/s/W for KECCAK-f[400]-based authenticated encryption respectively.

C. HWCE Performance and Power Evaluation

The *Fulmine* SoC includes many distinct ways to perform the basic operation of CNNs, i.e. 2D convolutions. In software, a naïve single core implementation of a 5×5 convolution filter has a throughput of 94 cycles per pixel. Parallel execution on four cores can provide almost ideal speedup reaching 24 cycles/px. Thanks to the SIMD extensions described in Section III, an optimized multi-core version can be sped up by almost 2× down to 13 cycles/px on average.

With respect to this baseline, the HWCE can provide a significant additional speedup by employing its parallel datapath, the line buffer (which saves input data fetch memory bandwidth), and weight precision scaling. We measured average throughput by running a full-platform benchmark, which therefore takes into account the overheads for real world usage: line buffer fill time, memory contention from cores, self-contention by HWCE inputs/outputs trying to access the same TCDM bank in a given cycle. Considering the full precision 16 bit mode for the weights, we measured an average inverse throughput of 1.14 cycles per output pixel for 5×5 convolutions and 1.07 cycles per output pixel for 3×3 convolutions - the two sizes directly supported by the internal datapath of the HWCE. This is equivalent to a 82× speedup with respect to the naïve single core baseline, or 11× with respect to a fully optimized 4-core version.

As described in Section IV-B, the HWCE datapath enables application-driven scaling of arithmetic precision in exchange for higher throughput and energy efficiency. In the 8 bit precision mode, average inverse throughput is scaled to 0.61 cycles/px and 0.58 cycles/px for the 5×5 and 3×3 filters, respectively; in 4bit mode, this is further improved to 0.45 cycles/px and 0.43 cycles/px, respectively. In the 4 bit precision mode, the HWCE is fully using its 4-port memory bandwidth towards the TCDM in order to load 4 $\mathbf{y}_{\text{in}}$ partial results and store back 4 $\mathbf{y}_{\text{out}}$ ones. Further performance scaling would therefore require an increase in memory bandwidth.Figure 6b reports time and energy per pixel, running the same set of filters in the KEC-CNN-SW operating mode while scaling the V_{DD} operating voltage. At 0.8 V, the energy to spend for an output pixel can be as low as 50 pJ per pixel, equivalent to 465 GMAC/s/W for a 5×5 filter.

D. Comparison with State-of-the-Art

Table I compares *Fulmine* with the architectures that define the boundaries of the secure data analytics application space described in Section II. Apart from area, power and performance, we also use an *equivalent energy efficiency* metric defined as the energy that a platform has to spend to perform an elementary RISC operation[2]. *Fulmine* achieves the highest result on this metric, 5.74 pJ per operation, thanks to the cooperation between its three kinds of processing engines. The second-best result is of SleepWalker (6.99 pJ) - but in an operating point where execution takes 89× more time than in the case of *Fulmine*.

Moreover, *Fulmine* provides better area efficiency than what is available in other IoT end-nodes: 32 SleepWalker chips would be needed to achieve the same performance as Fulmine in the workload of Section VI-B. On the other hand, coupling an efficient IoT microcontroller with external accelerators can theoretically provide an effective solution, but it requires continuous high-bandwidth data exchange from chip-to-chip, which is typically not practical in low-power systems. Conversely, in Fulmine HW accelerators are coupled to the cluster cores via the shared L1 memory, and no copy at all is required - only a simple pointer exchange.

For IoT endnodes, the smaller footprint of a System-on-Chip solution can also provide an advantage with respect to a traditional system on board, which is heavier and bulkier. Taking this reasoning one step further, while it is not always possible to place sensors and computing logic on the same die, the system we propose could be coupled to a sensor in a System-on-Package solution, requiring only a single die-to-die connection. Competing systems listed in Table I would require the integration of more than two dies on the same package, resulting in a more complex and expensive design.

VI. USE CASES

To evaluate the *Fulmine* SoC in full end-to-end applications, we propose three distinct use cases, which represent

[2]This is computed as the total energy per instruction on the workload presented in Section VI-B, which provides a balanced mix of encryption, convolution, other SW-based filters.

a necessarily incomplete selection of possible security- and performance-critical IoT sensor analytics applications. The first use case represents deep-learning based sensor analytics workloads that are predominantly executed locally on the endnode, but require security to access unsafe external memory (secure autonomous aerial surveillance, Section VI-A); the second one represents workloads executed only in part on the endnode, which therefore require secured connectivity with an external server (local face detection and remote recognition, Section VI-B). Finally, the third use case represents workloads in which, while analytics is performed online, data must also be collected for longer term monitoring (seizure detection and monitoring, Section VI-C).

For our evaluation, we consider the system shown in Figure 7. We use two banks (16 MB) of Microchip SST26VF064 bit quad-SPI flash memory to host the weights for a deep CNN as *ResNet-20*; each bank consumes down to $15\,\mu$A in standby and a maximum of $15\,\mathrm{mA}@3.6\,\mathrm{V}$ in QPI mode. Moreover, we use 2 MB of non-volatile Cypress CY15B104Q ferroelectric RAM (FRAM) as a temporary memory for partial results. Four banks are connected in a bit-interleaved fashion to allow access with quad-SPI bandwidth. Both the FRAM and the flash, as well as a camera and an ADC input, are connected to the *Fulmine* uDMA, which can be used to transfer data to/from the SoC L2 memory. The cluster then transfers tiles of the input data to operate on and writes results back to L2 via DMA transfers. We focus on the power spent for actual computation rather than on system power, i.e. we include power spent in transfers from memory used during the computation, but exclude data acquisition/transmission[3].

A. Secure Autonomous Aerial Surveillance

For the secure autonomous aerial surveillance use case, we consider deploying the system of Figure 7 on a low-power nano-UAV such as a CrazyFlie nano quadcopter [66]. Storms of tens or hundreds of these devices could provide diffused, fully autonomous, and low energy footprint aerial surveillance. In these vehicles the power budget for computing is extremely limited (more than 90% of the battery must be dedicated to the quadrotor engines), and continuous wireless data transmission from cameras is not an option due to its power overhead. Local elaboration and transmission of high-level labeling information provides a more efficient usage of the available power, while also granting greater availability in situations like disaster relief, where wireless propagation might be non-ideal and enable only low-bandwidth communication.

[3] We measured performance on each kernel composing the three applications and for SPI and DMA transfers via RTL simulation, and the related power consumption by direct measurement using an Advantest SoCV93000 integrated circuit tester, encapsulating the target kernel within an infinite loop. Power is measured at two distinct frequencies to obtain leakage and dynamic power density via linear regression. For external memories, we used publicly available data from their datasheets, considering always the worst case.

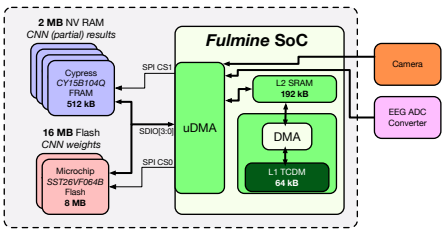


Fig. 7: A *Fulmine* SoC connected to 16 MB of Flash, 2 MB of FRAM, and sensors (the grey area is taken into account for power estimations).

Deployment of state-of-the-art deep CNNs on these devices naturally requires external memory for the storage of weights and partial results. These memories cannot be considered to be secure, as the weights deployed in the Flashare an important intellectual property and UAVs are fully autonomous, therefore vulnerable to malicious physical hijacking . Partial results stored in the FRAM and SPI traffic could be monitored or modified by an external agent, with the purpose of changing the final result classified by the UAV. Strong encryption for weights and partial results can significantly alleviate this issue, at the cost of a huge overhead on top of the pure data analytics workload.

Here we consider a deep *ResNet-20* CNN [10] to classify scenes captured from a low power sensor producing a 224×224 input image. *ResNet-20* has been shown to be effective on CIFAR-10 classification but can also be trained for other complex tasks, and it is in general a good representative of state-of-the-art CNNs of medium size. It consists of more than 1.35×10^9 operations, a considerable workload for a low power end-node. External memories are required for both weights (with a footprint of 8.9 MB considering 16 bits of precision) and partial results (with a maximum footprint of 1.5 MB for the output of the first layer). All weights and partial results are en-/decrypted with AES-128-XTS; the *Fulmine* cluster is considered the only secure enclave in which decrypted data can reside.

Figure 8 shows execution time and energy spent at $0.8\,\mathrm{V}$ for this compound workload. We exploit the fast frequency switching capabilities of *Fulmine* to dynamically switch from the CRY-CNN-SW operating mode (at $85\,\mathrm{MHz}$) when executing AES to the KEC-CNN-SW operating mode (at $104\,\mathrm{MHz}$) when executing other kernels. The figure also shows a breakdown of energy consumption regarding kernels (convolution CONV, encryption AES), densely connected CNN layers (DENSE), DMA transfers and other parts of the CNN (DMA, OTHER), and external memories and I/O (FRAM, FLASH, SPI I/O). In the baseline, where all the workload is run in software on a single core, energy consumption is entirely dominated by convolutions and encryption, with a 4-to-1 ratio between the two. When more features of the *Fulmine* SoC are progressively activated, execution time is decreased by $114\times$ and energy consumption by $45\times$, down to $27\,\mathrm{mJ}$ in total - $3.16\,\mathrm{pJ}$ per equivalent operation (defined as an equivalent OpenRISC instruction from [53]). When CNNs use the HWCE with 4 bit weights and AES-128-XTS uses the HWCRYPT, the overall energy breakdown shows that cluster computation is no longer largely dominant, counting for only slightly more than 50% of the total energy. Additional acceleration would likely require expensive hardware (e.g. more sum-of-products units or more ports in the HWCE) and would yield diminishing returns in terms of energy efficiency.

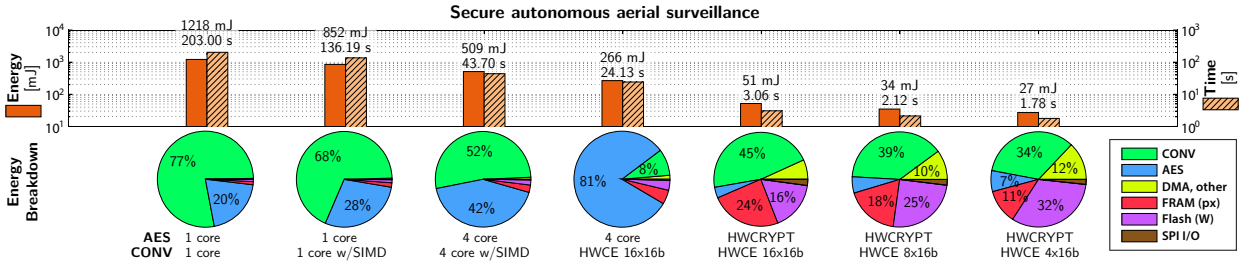


Fig. 8: Secure autonomous aerial surveillance use case based on a *ResNet-20* CNN [10] with AES-128-XTS encryption for all weights and partial results. KEC-CNN-SW and CRY-CNN-SW operating modes at V_{DD} =0.8 V.

To concretely estimate whether the results make it feasible to deploy a *ResNet-20* on a nano-UAV, consider that a CrazyFlie UAV [66] can fly for up to 7 minutes. Continuous execution of secure *ResNet-20* during this flight time corresponds to a total of 235 iterations in the operating point considered here. This would consume a total of 6.4 J of energy - less than 0.25% of the 2590 J available in the onboard battery - and the low peak power of 24 mW makes this concretely achievable in an autonomous device.

B. Local Face Detection with Secured Remote Recognition

Complete on-device computation might not be the most advantageous approach for all applications, particularly for those that can be clearly divided in a lower effort *triggering* stage and a higher effort one that is only seldom executed. A good example is the problem of face recognition. While state-of-the-art face recognition requires a significant workload in the order of billions of operations (e.g. FaceNet [68]), the problem can be easily decomposed in two stages: one where the input image is scanned to detect the presence of a face, and another where the detected faces are recognized. The first stage could be run continuously on a low-power wearable device such as a smartwatch, using an external device (e.g. a smartphone, the cloud) to compute the much rarer and much more complex second stage.

We envision *Fulmine* to be integrated into an ultra-low power (ULP) smartwatch platform similar to that presented in Conti et al. [69]. We consider a similar camera with the one used in Section VI-A producing a 224×224 input image. Face detection is performed locally, using the first two stages (12-net and 24-net) of the multi-stage CNN proposed by Li et al. [67]. If faces are detected by this two-stage CNN, the full input image is encrypted and transferred to a coupled smartphone for the recognition phase. The networks are applied to small separate 24×24 windows extracted from the input image; partial results need not be saved from one window to the next. Therefore the CNN does not use any external memory and can rely exclusively on the internal L2.

Figure 9 reports the experimental results for the local face detection use case in terms of energy and execution time. Baseline energy is almost evenly spent between convolutions, AES-128-XTS encryption, and densely connected CNN layers. Software optimizations such as parallelization, SIMD extensions are much more effective on convolutional and dense layers than they are on AES, due to XTS internal data dependencies in the tweak computation. Using hardware accelerators essentially reduces the energy cost of convolution and on AES-128-XTS to less than 10% of the total, and leads to a 24× speedup and a 13× reduction in energy with respect to the baseline. With all optimizations, face detection takes 0.57 mJ or 5.74 pJ per elementary operation. This face detection could be performed with no interruption for roughly 1.6 days before exhausting the battery charge, if we consider a small 4 V 150 mA h lithium-ion polymer battery. Duty cycling, taking advantage of the power management features of the SoC described in Section III, can prolong this time considerably.

C. Seizure Detection and Secure Long-Term Monitoring

Extraction of semantically relevant information out of biosignals such as electromyogram (EMG), electrocardiogram (ECG), and electroencephalogram (EEG) is a potentially huge market for low-power footprint IoT devices. Here we consider a seizure detection application based on a support vector machine (SVM) trained on energy coefficients extracted from the principal components analysis (PCA) of a multi-channel EEG signal [70][71]. The sampling frequency is 256 Hz with 50% overlapped windows, i.e. seizure detection is performed every 0.5 s. Starting from a 256-sample window of 23 input EEG channels (represented as 32 bit fixed-point numbers), PCA is applied to extract 9 components, that are then transformed by a digital wavelet transform (DWT) to extract energy coefficients, which are classified by an SVM. For long-term monitoring, the components produced by the PCA have to be collected and sent to the network to be stored or analyzed, which requires encryption due to the sensitivity of this data.

Figure 10 shows the results in terms of energy (split down between the various kernels) and execution time. Several components of PCA, like diagonalization, are not amenable to parallelization. Nonetheless, we observe a 2.6× speedup with four cores excluding AES encryption. Using the HWCRYPT, encryption becomes a *transparent* step of the algorithm and essentially disappears from the overall energy breakdown. Therefore, with combined SW parallelization and accelerated encryption, an overall 4.3× speedup and 2.1× energy reduction can be achieved. More importantly, the absolute energy consumption of 0.18 mJ (12.7 pJ per operation) means that a typical 2 A h@3.3 V pacemaker battery [72] would suffice for more than 130 million iterations, and more than 750 days if used continuously - as for most of the time the *Fulmine* SoC can be in *deep sleep* mode.

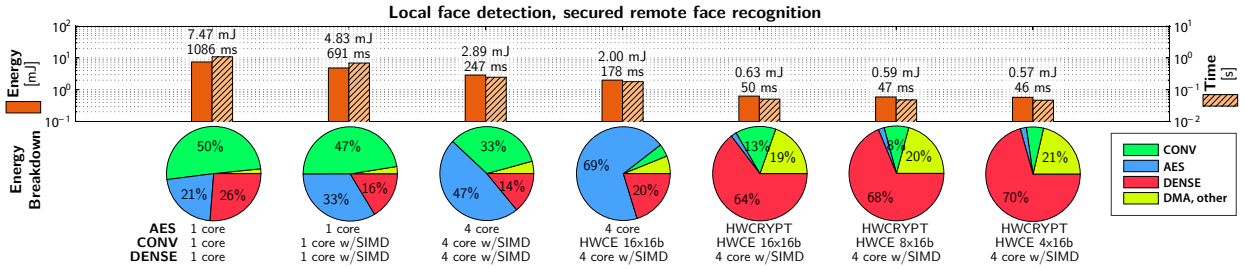


Fig. 9: Local face detection, secured remote recognition use case based on the 12-net and 24-net CNNs from Li et al. [67] on a 224×224 input image, with full AES-128-XTS encryption of the image if a potential face is detected. CRY-CNN-SW operating mode at V_{DD} =0.8 V. We consider that the first stage 12-net classifies 10% of the input image as containing faces, and that the second stage 24-net is applied only to that fraction.

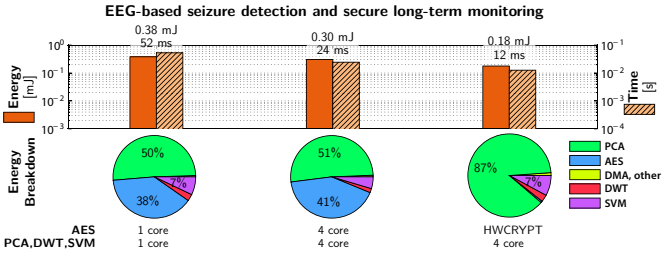


Fig. 10: EEG-based seizure detection and secure data collection. CRY-CNN-SW operating mode at V_{DD} =0.8 V.

VII. Conclusion

This work presented *Fulmine*, a 65 nm System-on-Chip targeting the emerging class of smart secure near-sensor data analytics for IoT end-nodes. We achieve this without using aggressive technology or voltage scaling, but through the architectural solution of combining cores and accelerators within a single tightly-coupled cluster. The use cases we have proposed show that this approach leads to improvements of more than one order of magnitude in time and energy with respect to a pure software based solution, with no sacrifice in terms of flexibility. The *Fulmine* SoC enables secure, integrated and low-power *secure data analytics* directly within the IoT node. Without any compromise in terms of security, the proposed SoC enables sensemaking in a budget of a few pJ/op - down to 3.16 pJ/op in one case, or 315 Gop/s/W.

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342513

[2] H. Kumarage, I. Khalil, A. Alabdulatif, Z. Tari, and X. Yi, "Secure Data Analytics for Cloud-Integrated Internet of Things Applications," *IEEE Cloud Computing*, vol. 3, no. 2, pp. 46–56, Mar 2016.

[3] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in *2012 10th International Conference on Frontiers of Information Technology*, Dec. 2012, pp. 257–260.

[4] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen, and S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities," in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, Nov. 2014, pp. 230–234.

[5] D. Dinu, Y. Le Corre, D. Khovratovich, L. Perrin, J. Grossschädl, and A. Biryukov, "Triathlon of Lightweight Block Ciphers for the Internet of Things." *IACR Cryptology ePrint Archive*, vol. 2015, p. 209, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[7] L. Cavigelli, M. Magno, and L. Benini, "Accelerating Real-time Embedded Scene Labeling with Convolutional Networks," in *Proceedings of the 52Nd Annual Design Automation Conference*, ser. DAC '15. New York, NY, USA: ACM, 2015, pp. 108:1–108:6.

[8] R. Girshick, J. Donahue, T. Darrell, J. Malik, and U. C. Berkeley, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[9] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015.

[11] F. Conti, C. Pilkington, A. Marongiu, and L. Benini, "He-P2012 : Architectural Heterogeneity Exploration on a Scalable Many-Core Platform," in *Proceedings of 25th IEEE Conference on Application-Specific Architectures and Processors*, 2014.

[12] D. A. McGrew and J. Viega, "The Security and Performance of the Galois/Counter Mode (GCM) of Operation," in *International Conference on Cryptology in India*. Springer, 2004, pp. 343–355.

[13] S. Gueron, "AES-GCM software performance on the current high end CPUs as a performance baseline for CAESAR competition," *Directions in Authenticated Ciphers (DIAC)*, 2013.

[14] "Maxim Integrated MAXQ1061 DeepCover Cryptographic Controller for Embedded Devices." [Online]. Available: https://www.maximintegrated.com

[15] 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 22nm Tri-Gate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 1048–1058, 2015.

[16] 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 *VLSI Circuits (VLSI-Circuits), 2016 IEEE Symposium on*. IEEE, 2016, pp. 1–2.

[17] W. Zhao, Y. Ha, and M. Alioto, "Novel Self-Body-Biasing and Statistical Design for Near-Threshold Circuits With Ultra Energy-Efficient AES as Case Study," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 8, pp. 1390–1401, Aug. 2015.

[18] C. Hocquet, D. Kamel, F. Regazzoni, J.-D. Legat, D. Flandre, D. Bol, and F.-X. Standaert, "Harvesting the Potential of Nano-CMOS for Lightweight Cryptography: An Ultra-Low-Voltage 65 Nm AES Coprocessor for Passive RFID Tags," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 79–86, 2011.

[19] S. Chintala, "convnet-benchmarks," 2016. [Online]. Available: https://github.com/soumith/convnet-benchmarks

[20] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks," in *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '15. New York, NY, USA: ACM, 2015, pp. 161–170.

[21] P. Meloni, G. Deriu, F. Conti, I. Loi, L. Raffo, and L. Benini, "A high-efficiency runtime reconfigurable IP for CNN acceleration on a mid-range all-programmable SoC," in *2016 International Conference on*

ReConFigurable Computing and FPGAs (ReConFig), Nov. 2016, pp. 1–8.

[22] F. Conti, A. Pullini, and L. Benini, "Brain-inspired Classroom Occupancy Monitoring on a Low-Power Mobile Platform," in *CVPR 2014 Workshops*, 2014.

[23] Movidius, "INS-03510-C1 Datasheet," 2014, datasheet of Myriad 2 Vision Processor. [Online]. Available: http://uploads.movidius.com/1441734401-Myriad-2-product-brief.pdf

[24] "Mobileye 560." [Online]. Available: http://www.mobileye.com/products/mobileye-5-series/mobileye-560

[25] "Himax, emza and CEVA Partner to Create Ultra-Low Power, Always-On Vision Sensor for IoT." [Online]. Available: http://www.ceva-dsp.com/

[26] L. Cavigelli and L. Benini, "Origami: A 803 GOp/s/W Convolutional Network Accelerator," *arXiv:1512.04295 [cs]*, Dec. 2015.

[27] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "ShiDianNao: Shifting Vision Processing Closer to the Sensor," in *ISCA '15*, ser. ISCA '15. New York, NY, USA: ACM, 2015, pp. 92–104. [Online]. Available: http://doi.acm.org/10.1145/2749469.2750389

[28] Y. H. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *ISSCC-16*, Jan 2016, pp. 262–263.

[29] J. Sim, J. Park, M. Kim, D. Bae, Y. Choi, and L. Kim, "A 1.42TOPS/W deep convolutional neural network recognition processor for intelligent IoE systems," in *ISSCC 2016*, Jan 2016, pp. 264–265.

[30] F. Conti and L. Benini, "A Ultra-Low-Energy Convolution Engine for Fast Brain-Inspired Vision in Multicore Clusters," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, ser. DATE '15. San Jose, CA, USA: EDA Consortium, 2015, pp. 683–688.

[31] S. Park, K. Bong, D. Shin, J. Lee, S. Choi, and H.-J. Yoo, "A 1.93TOPS/W Scalable Deep Learning/Inference Processor with Tetra-Parallel MIMD Architecture for Big-Data Applications," in *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*, 2016.

[32] M. Courbariaux, Y. Bengio, and J. P. David, "BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 3123–3131.

[33] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "YodaNN: An Ultra-Low Power Convolutional Neural Network Accelerator Based on Binary Weights," *arXiv:1606.05487 [cs]*, Jun. 2016.

[34] Y. Chen, E. Yao, and A. Basu, "A 128-Channel Extreme Learning Machine-Based Neural Decoder for Brain Machine Interfaces," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 679–692, June 2016.

[35] E. Yao and A. Basu, "VLSI Extreme Learning Machine: A Design Space Exploration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–15, 2016.

[36] "Texas Instruments MSP430 Low-Power MCUs."

[37] "STMicroelectronics STM32L476xx Datasheet."

[38] "LPC54000 Series: Low Power Microcontollers (MCUs) based on ARM Cortex-M4 Cores with optional Cortex-M0+ co-processor." [Online]. Available: www.nxp.com/LPC54000

[39] "Apollo Ultra-Low-Power Microcontrollers." [Online]. Available: http://ambiqmicro.com/apollo-ultra-low-power-mcu/

[40] J. Myers, A. Savanth, D. Howard, R. Gaddh, P. Prabhat, and D. Flynn, "An 80nW retention 11.7pJ/cycle active subthreshold ARM Cortex-M0 subsystem in 65nm CMOS for WSN applications," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb 2015, pp. 1–3.

[41] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J.-D. Legat, "SleepWalker: A 25-MHz 0.4-V Sub-mm2 7-uW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 20–32, Jan. 2013.

[42] S. Paul, V. Honkote, R. Kim, T. Majumder, P. Aseron, V. Grossnickle, R. Sankman, D. Mallik, S. Jain, S. Vangal, J. Tschanz, and V. De, "An energy harvesting wireless sensor node for IoT systems featuring a near-threshold voltage IA-32 microcontroller in 14nm tri-gate CMOS," in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, June 2016, pp. 1–2.

[43] A. Roy, P. J. Grossmann, S. A. Vitale, and B. H. Calhoun, "A 1.3 uW, 5pJ/cycle sub-threshold MSP430 processor in 90nm xLP FDSOI for energy-efficient IoT applications," in *2016 17th International Symposium on Quality Electronic Design (ISQED)*, March 2016, pp. 158–162.

[44] "RealTimeLogic. SharkSSL/RayCrypto v2.4 crypto library benchmarks with ARM Cortex-M3." [Online]. Available: https://realtimelogic.com/products/sharkssl/Cortex-M3/

[45] "CMSIS - Cortex Microcontroller Software Interface Standard." [Online]. Available: https://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php

[46] M. Nakajima, I. Naka, F. Matsushima, and T. Yamauchi, "A 20uA/MHz at 200MHz microcontroller with low power memory access scheme for small sensing nodes," in *2016 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS XIX)*, April 2016, pp. 1–3.

[47] M. Wu, R. Iyer, Y. Hoskote, S. Zhang, J. Zamora, G. Fabila, I. Klotchkov, and M. Bhartiya, "Design of a low power SoC testchip for wearables and IoTs," in *2015 IEEE Hot Chips 27 Symposium (HCS)*, Aug 2015, pp. 1–27.

[48] A. Pullini, F. Conti, D. Rossi, I. Loi, M. Gautschi, and L. Benini, "A Heterogeneous Multi-Core System-on-Chip for Energy Efficient Brain Inspired Vision," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 2910–2910.

[49] M. Konijnenburg, S. Stanzione, L. Yan, D.-W. Jee, J. Pettine, R. van Wegberg, H. Kim, C. van Liempd, R. Fish, J. Schuessler, and others, "A Multi (bio) sensor Acquisition System With Integrated Processor, Power Management, 8 x 8 LED Drivers, and Simultaneously Synchronized ECG, BIO-Z, GSR, and Two PPG Readouts," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 11, pp. 2584–2595, 2016.

[50] K. H. Lee and N. Verma, "A Low-Power Processor With Configurable Embedded Machine-Learning Accelerators for High-Order and Adaptive Analysis of Medical-Sensor Signals," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, July 2013.

[51] 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*. IEEE, 2015, pp. 1–39.

[52] 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*. IEEE, Mar. 2011, pp. 1–6.

[53] *OpenRISC 1000 Architecture Manual*, 2012.

[54] M. Gautschi, P. D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gürkaynak, and L. Benini, "Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–14, 2017.

[55] A. Teman, D. Rossi, P. Meinerzhagen, L. Benini, and A. Burg, "Power, Area, and Performance Optimization of Standard Cell Memory Arrays Through Controlled Placement," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 21, no. 4, pp. 59:1–59:25, May 2016.

[56] D. Rossi, A. Pullini, I. Loi, M. Gautschi, F. K. Gürkaynak, A. Teman, J. Constantin, A. Burg, I. M. Panades, E. Beigné, F. Clermidy, F. Abouzeid, P. Flatresse, and L. Benini, "193 MOPS/mW 162 MOPS, 0.32V to 1.15V Voltage Range Multi-Core Accelerator for Energy-Efficient Parallel and Sequential Digital Processing," 2016.

[57] D. Rossi, I. Loi, G. Haugou, and L. Benini, "Ultra-Low-Latency Lightweight DMA for Tightly Coupled Multi-Core Clusters," in *Proceedings of the 11th ACM Conference on Computing Frontiers - CF '14*. New York, New York, USA: ACM Press, 2014, pp. 1–10.

[58] N. FIPS, "Advanced encryption standard (AES)," *Federal Information Processing Standards Publication*, vol. 197, pp. 441–0311, 2001.

[59] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Keccak Sponge Function Family Main Document," *Submission to NIST (Round 2)*, vol. 3, p. 30, 2009.

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

[61] P. Rogaway, "Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2004, pp. 16–31.

[62] S. Gueron, "Intel® Advanced Encryption Standard (AES) New Instructions Set," *Intel Corporation*, 2010.

[63] H. Wu and B. Preneel, "AEGIS: A Fast Authenticated Encryption Algorithm," in *International Conference on Selected Areas in Cryptography*. Springer, 2013, pp. 185–201.

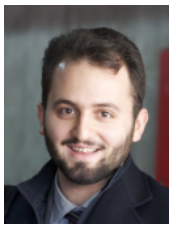
[64] V. T. Hoang, T. Krovetz, and P. Rogaway, "Robust Authenticated-Encryption AEZ and the Problem That It Solves," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 15–44.

[65] M. Courbariaux, Y. Bengio, and J.-P. David, "Training Deep Neural Networks with Low Precision Multiplications," *arXiv:1412.7024 [cs]*, Dec. 2014.

[66] "The Crazyflie Nano Quadcopter — Bitcraze."

[67] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A Convolutional Neural Network Cascade for Face Detection," 2015, pp. 5325–5334.

[68] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," 2015, pp. 815–823.
[69] F. Conti, D. Palossi, R. Andri, M. Magno, and L. Benini, "Accelerated Visual Context Classification on a Low-Power Smartwatch," *IEEE Transactions on Human-Machine Systems*, vol. PP, no. 99, pp. 1–12, 2016.
[70] S. Benatti, F. Casamassima, B. Milosevic, E. Farella, P. Schönle, S. Fateh, T. Burger, Q. Huang, and L. Benini, "A Versatile Embedded Platform for EMG Acquisition and Gesture Recognition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 5, pp. 620–630, 2015.
[71] S. Benatti, F. Montagna, D. Rossi, and L. Benini, "Scalable EEG Seizure Detection on an Ultra Low Power Multi-Core Architecture," in *Proceedings of IEEE BIOCAS 2016*, 2016.
[72] V. S. Mallela, V. Ilankumaran, and N. Rao, "Trends in Cardiac Pacemaker Batteries," *Indian Pacing and Electrophysiology Journal*, vol. 4, no. 4, pp. 201–212, Oct. 2004.



Francesco Conti received the Ph.D. degree from University of Bologna in 2016 and is currently a post-doctoral researcher at the Integrated Systems Laboratory, ETH Zürich, Switzerland and the Energy-Efficient Embedded Systems laboratory, University of Bologna, Italy. He has co-authored more than 20 papers on international conferences and journals. His research focuses on energy-efficient multicore architectures and applications of deep learning to low power digital systems.

Robert Schilling received has B.Sc. and M.Sc. degrees in information and computer engineering from Graz University of Technology, where he is currently pursuing a Ph.D. degree. His current research interests include countermeasures against fault attacks, security enhanced processors and compilers, and software security.

Pasquale Davide Schiavone received his B.Sc. (2013) and M.Sc. (2016) in computer engineering from Polytechnic of Turin. Since 2016 he has started his PhD studies at the Integrated Systems Laboratory, ETH Zurich. His research interests include low-power microprocessors design in multi-core systems and deep-learning architectures for energy-efficient systems.

Antonio Pullini received the M.S. degree in electrical engineering from Bologna University, Italy, and he is currently pursuing a Ph.D. degree from the Integrated Systems Laboratory, ETH Zürich, Switzerland. He was a Senior Engineer at iNoCs S.à.r.l., Lausanne, Switzerland. His research interests include low-power digital design and networks on chip.

Davide Rossi is an assistant professor at the Energy Efficient Embedded Systems Laboratory at the University of Bologna. His current research interests include ultra-low-power multicore SoC design and applications. He has published more than 50 papers on international conferences and journals.

Frank Kağan Gürkaynak obtained his B.Sc. and M.Sc. degrees from Electrical and Electronical Engineering Department of the Istanbul Technical University and his Ph.D. degree from ETH Zürich. He is employed by the Microelectronics Design Center of ETH Zrich and his research interests include design of VLSI systems, cryptography, and energy efficient processing systems.

Michael Muehlberghuber received the MSc degree from Graz University of Technology in 2011 and the Ph.D. degree from ETH Zurich in 2017. His research interests include hardware Trojans and the development of VLSI architectures of cryptographic primitives, targeting resource-constrained environments and high-performance applications.

Michael Gautschi received the M.Sc. degree in electrical engineering and information technology from ETH Zürich, Switzerland, in 2012. Since then he has been with the Integrated Systems Laboratory, ETH Zürich, pursuing a Ph.D. degree. His current research interests include energy-efficient systems, multi-core SoC design, mobile communication, and low-power integrated circuits.

Igor Loi received the B.Sc. degree in electrical engineering from the University of Cagliari, Italy, in 2005, and the Ph.D. degree from the Department of Electronics and Computer Science, University of Bologna, Italy, in 2010. He currently holds a researcher position in electronics engineering with the University of Bologna. His research interests include ultra-low power multicore systems, memory system hierarchies, and ultra low-power on-chip interconnects.

Germain Haugou received the Engineering Degree in Telecommunication from the University of Grenoble (INP Ensimag), in 2004. He worked for 10 years in STMicroelectronics as a research engineer. He is currently working at the Swiss Federal Institute of Technology in Zurich, Switzerland, as a research assistant. His research interests include virtual platforms, run-time systems, compilers and programming models for many-core embedded architectures.

Stefan Mangard is a professor at TU Graz since 2013 and heads the Secure Systems Group. His research focuses on device and system security. This in particular includes hardware/software security architectures, physical attacks and countermeasures, microarchitectural attacks and countermeasures, cryptography, as well as secure and efficient hardware and software implementations of cryptography.

Luca Benini holds the chair of Digital Circuits and Systems at ETH Zürich and is Full Professor at the Università di Bologna. Dr. Benini's research interests are in energy-efficient system design for embedded and high-performance computing. He has published more than 800 papers, five books and several book chapters. He is a Fellow of the ACM and a member of the Academia Europaea. He is the recipient of the 2016 IEEE CAS Mac Van Valkenburg award.