Microcontrollers for IoT: Optimizations, Computing Paradigms, and Future Directions

Rym Chéour[1], Sabrine Khriji[2,3], Mohamed abid [1], Olfa Kanoun [2]
[1] *National School of Engineers of Sfax, Computer and Embedded System Laboratory, University of Sfax, Tunisia*
[2] *Chemnitz University of Technology, Measurement and Sensor Technology, Chemnitz, Germany*
[3] *National School of Engineers of Sfax, LETI Laboratory, University of Sfax, Tunisia*

***Abstract*—Internet of Things (IoT) is a paradigm covering almost every area of life, from education to health, from services to products, and from home to smart cities invading any area in which IoT is applicable, like automotive, infrastructure and business. But many challenges remain to be met, ranging from sensor energy autonomy, to data security, to interoperability of numerous objects. The high computation capability is a crucial contributor to the progress of IoT solutions. So maximizing the processing resources in a single CPU. This paper reviews the recent microcontrollers architectures employed in the context of wireless sensor networks and IoT, and outlines many of the low-power techniques allowing a wide range of applications. An overview of the foremost recent research topics and future trends are also discussed.**

***Index Terms*—Internet of Things (IoT), Microcontrollers, Wireless sensor networks, Energy Efficiency, Performance, Edge computing, Big Data, Processor Design, IoT Processor Features, Energy harvesting.**

I. Introduction

The advent of Big Data technologies is reshaping the market, with the ability to store on a regular rack hundreds of terabytes of data and process them with response times that are revolutionizing the storage world [1]. The rush to vast warehouses full of servers and storage systems, powered primarily by Intel's high-end microprocessors or to sophisticated processors to support an operating system, remotely upgradable firmware, high programmability and embedded device management agents define the capabilities of a microprocessor [2]. Therefore, the need to set up an appropriate processor that could process this upstream data in a network in a given time is challenging.

In recent years, new computing paradigms have been introduced whereby microprocessors have received much less attention at the expense of communication. The challenges of computing on Internet of Thing (IoT) devices are particularly high as the microcontroller (MCU) must comply with specific additional requirements. The wide variety of applications and platforms available on Wireless Sensor Networks (WSNs) accentuates the crucial need to choose the appropriate architectures. Combining complex features with a wireless sensor node such as security and video processing increases significantly the size of the embedded software program stored in the program memory [3]. Therefore, there is a simultaneous rise in the time to handle the massive quantity of information by the processing unit and the wireless network traffic throughout the information exchanges. [4]. Similarly, this overload has a substantial effect on the energy consumption, which is strongly related to the sensor node lifetime.

Despite the tight battery lifetime and power constraints, CPUs capabilities continue an upward trend. Thus, the edge node must be well provisioned in terms of energy to perform the required calculations while respecting the design constraints of the nodes. The current processor architecture allows fairly high computing performance to be achieved, but at the cost of high energy consumption of around 100 Watts [5]. Trading-off the overall energy consumption with the performance and security capabilities of the low-budget wireless devices is quite challenging, as these requirements usually point to opposite directions. Numerous intelligent energy management methods policies emerged from a wide range of research studies like Maximum Power Point Tracking (MPPT), Dynamic Voltage/Frequency scaling (DVFS) on loads and the implementation of advanced task scheduling for the microsystem.

Sensor network technologies will rise to meet this new challenge. This paper describes a baseline understanding of the fundamental concepts of MCUs including the key concepts and the main advantages. It gives a deep overview about 19 microcontrollers in the field of WSN, IoT and Big Data. Moreover, this study surveys some potential microprocessor optimizations and computing paradigms, which enable the design of right-provisioned microprocessor architectures.

The reminder of the paper is as follows. In the section II, the requirements of the MCU and the limitations of current resource management approaches are identified. In the following section III, some energy autonomy opportunities are investigated besides of some criteria for evaluating solutions and, on the basis of these criteria, carry out a state of the art of some research work. A discussion is carried-out in section IV. Future trends of microcontrollers platforms for IoT are investigated in section V. Finally, a concluion is given.

II. Challenges for microcontrollers working on the edge of IoT

The quantification of the performance of an MCU is measured by its tendency to handle intense computations for artificial intelligence, hash, deep neural networks applications where tons of raw data are collected and further processed automatically in a short period of time. Hence, the processing

978-1-7281-5503-6/20/$31.00 ©2020 IEEE

time needs to be optimized for a better performance. If overloaded, the MCU must keep the system operable and adjust to variations in run time and data characteristics. Also, the design of MCU faces the battery size constraint: power-hungry applications such as security and cryptography can, dramatically, drain the totality of the available CPU power. For example, in the field of e-health, images for Magnetic Resonance Imaging (MRI) are bulky, highly complex and require high resolution, which leads to bandwidth overload, leading to delays in data arrival and violations of real-time constraints [6]. When computations become more complex for image processing for example, the demand for memory increases thus rendering the use of other platforms more powerful and better adapted to the application necessary.

However, IoT's heterogeneous architectures pose multiple challenges [7]. Graphics processing units (GPUs) or digital signal processing DSPs as hardware accelerators enable parallelism, thereby allowing increasingly complex and larger tasks (millions of runs) to be processed in ever faster times associated with higher and higher data volumes. Programming models and associated tools are at the center of this evolution. However, the energy consumption needs to be monitored throughout the lifetime of the application. So, mixing different hardware has many bottlenecks including memory bandwidth limitation, memory access latency, performance instabilities due to many interactions between components causing faulty data sharing.

To conclude, a microcontroller for IoT must be cost-efficient, area-efficient, meet the real-time constraints of the applications, with a maximum lifetime. MCU tasks also consume a higher amount of power in the WSN devices behind the communication unit. Time-consuming processing tasks such as data analysis, security and encryption increase energy consumption considerably. For this reason, managing active and inactive states will reduce the microcontroller's power consumption.

III. Energy Autonomy Optimizations Techniques

Generally, measuring the power consumption of an edge node is too hard because juggling two things, the dynamic current versus the static current (less available), is difficult [8]. The clock speed has to be increased, which increases the dynamic current and it may not be linear. So the more steps in a shorter window of time means more power from dynamic current. With larger circuits clocked at lower speeds static current from stuff like leakage ends up dominating here [9].

A. Variable CPU speed

Depending on workload, changing the task's frequency can significantly reduce energy consumption. So, operating a CPU at its maximum frequency without taking into account the variation of the workload induced to a loss of energy. As a result, a novel technology called Dynamic Voltage and Frequency Scaling (DVFS) was incorporated into processors by hardware manufacturers where voltage and frequency are scaled independently [10]. DVFS method aims to minimize energy dissipation when tasks are performed by dynamically adjusting the voltage and operating frequency according to its load [11]. At low CPU loads, the frequency and voltage of the CPU are dropped progressively to avoid compromising execution latency. The CPU operates at its maximum frequency (and voltage), active state, when it is heavily stressed or has a load above a defined threshold.

B. Power-mode variation

Typically, CPU energy saving policies focus on the state switching. They intended to put the CPU in a standby state or to change the runtime speed. For an efficient energy use in idle mode, the most recent CPU architectures rely on a specific power-down state known as sleep state, which has several progressive levels: sleep, deep sleep and deeper sleep. Cache data loss and zero power consumption arise in the sleep state, but the transition back to the active state requires an extended time delay. As the sleep state deepens, the time to go back to the active state is longer, which affects the task execution as shown in table I. After a long idle period, the CPU enters a sleep state and can then go through each of these sleep states sequentially [12].

TABLE I
Transition states and latency supported by ATmega128L

Transition	Delay
idle to run	10 μs
idle to sleep	0
sleep to run	160 ms
sleep to idle	0
run to sleep	90 ms
run to idle	10 μs

Another effective technique for minimizing power consumption is Dynamic Power Management (DPM), where the level of power consumption is tuned according to the power-mode by automatically shutting down the node when idle and waking it up with a software command [11]. Although DPM consumes additional dynamic power and mode switching overheads, using it without the DVFS generates additional energy costs and latency penalty due to the elongated task execution time and the non-negligible transition delays. So, there is a great need of an optimized DPM scheme [13].

C. Batching

Request batching is defined as delaying the execution of a group of applications in batches and putting the processor to sleep between each batch [14]. This mechanism consists of grouping requests in memory for a predefined period (called batching timeout) and processing them once the deadline has been reached. Before the deadline, the CPU is idle and has almost no power consumption. When the timeout period expires, the CPU switches to the active state and operates

at full load to execute all the requests that have built up. In [14], author presented a request batching mechanism, where the network interface card accumulates the incoming requests in its memory, while the host processor of the server is kept in a low-power state. In case of a wait longer than a batch delay, the host CPU is woken up. However, request batching trades off system responsiveness to lower energy consumption. This is not appropriate for trade, since slow servers might drive away customers. In this design, each processing stage waits until the previous stage has finished the batch [15].

D. Computation level

Code optimization consists of decreasing external memory accesses since instructions are proportional to power consumption. Since manual optimization is challenging for the user, inlining, looping and cloning are among the most common automatic optimization functions. A function call is replaced by the body of the function with the inlining method. Instead of using a function call procedure, the loop method copies the code into a function. Whereas, the cloning technique creates specialized versions of functions that are called from different call sites with many arguments. Although, the two previous software power management techniques are not primarily intended to reduce the consumption at the software level but rather to reduce the execution time. However, once this goal is reached, a significant decrease in the energy consumption is noticed.

The operating system (OS) also contributes to reduce energy consumption [16]. TinyOS, an OS dedicated to WSN, uses an event-driven execution model. The OS turns off the CPU between events to when a processor is idle with fast wake-up times [17]. Tasks can be executed or post-processed according to an event manager or other tasks, those that are pending, are queued and run as FIFOs. The system goes into a low-power states as the queue is empty until the next interruption. Along with the CPU, other devices can also be on standby. So, in terms of operating systems, energy reduction is based on two methods. On the one hand, using both scheduling and Dynamic Voltage Scaling (DVS) techniques to allocate the CPU time to predefined tasks and to manage the CPU's power states [11], [18]. Parallel thread processing is cluster-based, where the clusterhead (CH) is the edge node responsible for data collection and processing [19].

E. Hardware Solutions

With the advent of multi-core architectures, many programmers use modern processors to maximize code performance by executing independent instructions in separate processors. This solution is called parallelism. Different cores execute in parallel code more efficiently, and a range of special-purpose accelerators are used to support the large computing cores.

Field Programmable Gate Array (FPGA), as heterogeneous architecture, is a key technical that provides flexibility since their inner circuits can be invoked individually and can communicate with each other without going through the CPU [36]. These internal circuits support massive volumes of data to meet the evolving computing needs. FPGAs are configured to run tasks in parallel in a single clock cycle. Large-scale FPGAs are available in servers in large data centers, also to accelerate certain functions (often related to machine learning or data mining). However, the use of an FPGA in IoT applications, has some limits since the need for processing speed remains predominant.

Each MCU has a particular architecture that may contain a queue, peripherals, GPIOs, etc. affecting its overall energy consumption. Some platforms rely on the use of a queue that groups together the tasks to be performed and passes them to the arithmetic and logical unit (ALU) once their execution time has come. This prefetching phase promotes the speed of execution but, consumes an extra energy. So a trade-off between speed and energy is necessary. Also, the width of the bus affects energy, for an 8 bits-bus, the transmission of operations on 10 bits would be done in 2 times. Also in contrast to 8 bits, a 16 bits microcontroller provides more accuracy and efficiency than 8 bits while performing the arithmetic and logic operations. Automatic control system with measurement, comparison, calculation and correction functions such as implantable medical devices and office machines are mainly based on 32-bit microcontrollers. Finally, the bus width, memory width and speed, etc have a great impact in the energy consumption.

IV. Discussion

Processors differ in their architectures, features and capabilities as shown in table II. Some processor configurations are adapted to a particular application requiring either high speed (72 MHz) therefore high frequency or an average speed at the expense of long lifetime or the cost as illustrated in figure 1. This figure presents the trade-off between cost and CPU speed for different nodes. Memories differ in their capacities, technologies and word sizes for memory optimization (8,16,32 bits). This diversity of computing resources brings new constraints that the runtime system has to handle. For a multiprocessor architecture, the ALUs' power and performance levels should be monitored tightly so that the power can be turned off if necessary.

At the architectural level, the target of the existing solutions is to impulse new dynamics based on a tight trade-off between performance and power issues of the wireless mote. The instruction decoding and execution logics belong to a single location, which allows better performance and power management compared to distributing instructions across multiple chips. Among the sources of power consumption and delays of a CPU is the variation of capacitive loads [5].

DPM delivers extra power at each power-mode transition, increasing the latency and the difficulty to model the idle interval at the dynamic level. However, DVFS has a high complexity level without reducing static power consumption. Moreover, most of the available works do not consider the overhead due to the changes in the voltage/frequency pair. These approaches reduce the time losses, but they require centralized control, which lead to additional calculations (call to

TABLE II
TYPICAL CONFIGURATION OF SOME IOT NODES

	Microcontrollers	Communication	Energy	Application	Frequency
Sprouts [20]	ARM Cortex M3	Bluetooth	Energy harvesting: Vibration	General Purpose	
[21]	MSP430	CC2530 antenna	ZigBeePRO Low power mode	Outdoor :sensing (CO2, temperature,lighting, room occupancy, and fire detection)	
[3]	MSP430	CC2500 antenna	-3 V battery -very low-power communications	Fire hazard detection	
[22]	Raspberry Pi+ Arduino Uno	ZigBee	High power consumption	RT urban mobility monitoring	
E2MWSN [23]	ATMEGA1281 AT91SAM7Sx	IEEE802.15.4 ZigBee	0.053 mJ	General Purpose	14MHz
CUTE [24]	ARM Cortex-M3 – 32-bit	CC2520 IEEE 802.15.4		IoT applications	
HaloMote [25]	Atmel ATmega 256RFR2	TI CC2530	30 mW	IoT applications	14MHz
HireCookie [26]	TI MSP430 Microblaze	ZigBee, WI-FI, Ethernet or 802.15.4		Cyber-Physical Systems	
SENTIOF [27]	AVR32UC3B	IEEE 802.15.4 CC2520	74.811 mJ	Industrial monitoring	
[28]	MSP430BT5190 16bits	EZ430- RF256x Bluetooth	260 nW	IoT, healthcare	
Waspmote [29]	ATmega1281 – 8-bit	802.15.4/ ZigBee	15 mA	IoT -Smart city smart-water,	14MHz
TSmoTe [30]	Cortex-M3 – 32-bit	ZigBee, Wi-Fi, IEEE802.15.4, GPRS, NFC/RFID	40 mA	IoT applications	72MHz
M3/A8 Open Node [2]	Cortex-M3 32-bits	AT86RF231: ZigBee, 6LoWPAN,	14 mA	IoT applications	72Mhz
Mica2/Z Motes [31]	ATmega128L – 8-bit	TR1000	89 mW	General purpose	14MHz
TelosB [32]	TI MSP430F1611 – 16-bit	CC2420	32/ 69mW	IoT Academic	8MHz
IRIS Mote [33]	ATmega1281 – 8-bit	Atmel ATRF 230	17 mA	IoT applications	14MHz
Preon32 [34]	Cortex-M3	Radio	28,3 mA	General purpose	
WiSense [35]	TI MSP430G2955 16-bit	CC2520 LORA, 2G/3G/4G, BLE ZigBee, WiFi,	200 mA	IoT applications	16MHz

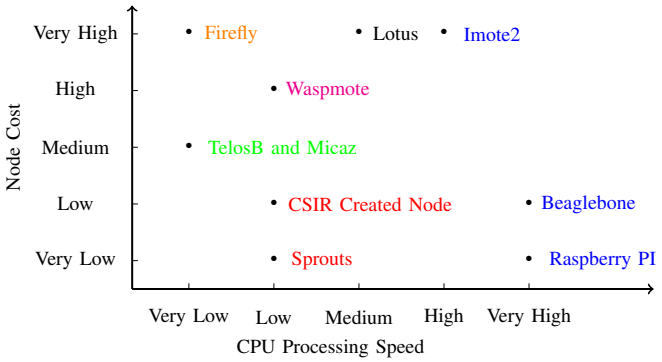

Fig. 1. Processing speed versus nodes cost [37]

the sequencer during the task, more frequent context changes, etc.) [37]. Using the DVFS technique at the expense of the batch technique is more advantageous because it is applied during both activity and inactivity periods. A novel approach both techniques independently depending on the system load. If no requests are received, the system is set to standby until the batch period expires. Thanks to the DVFS, the CPU runs at different energy-efficient frequency levels depending on its current load.

V. FUTURE DIRECTIONS OF IOT MICROCONTROLLERS

CPU design trends include hardware accelerators for machine learning, image recognition, compression, security, and high-frequency exchange systems with a low power consumption rate. Multiple instruction complex algorithms can be accelerated by a hardware function. Another key emerging trend motivated by the current approaches is multi-core architectures and multi-core embedded wireless sensor networks. The challenge then is the number and choice of cores, and scheduling of applications to the appropriate cores. Since the ultimate purpose of IoT is to act on connected intelligent objects based on collected data from machines, sensors, devices, IoT faces multiple challenges including transmission time, latency

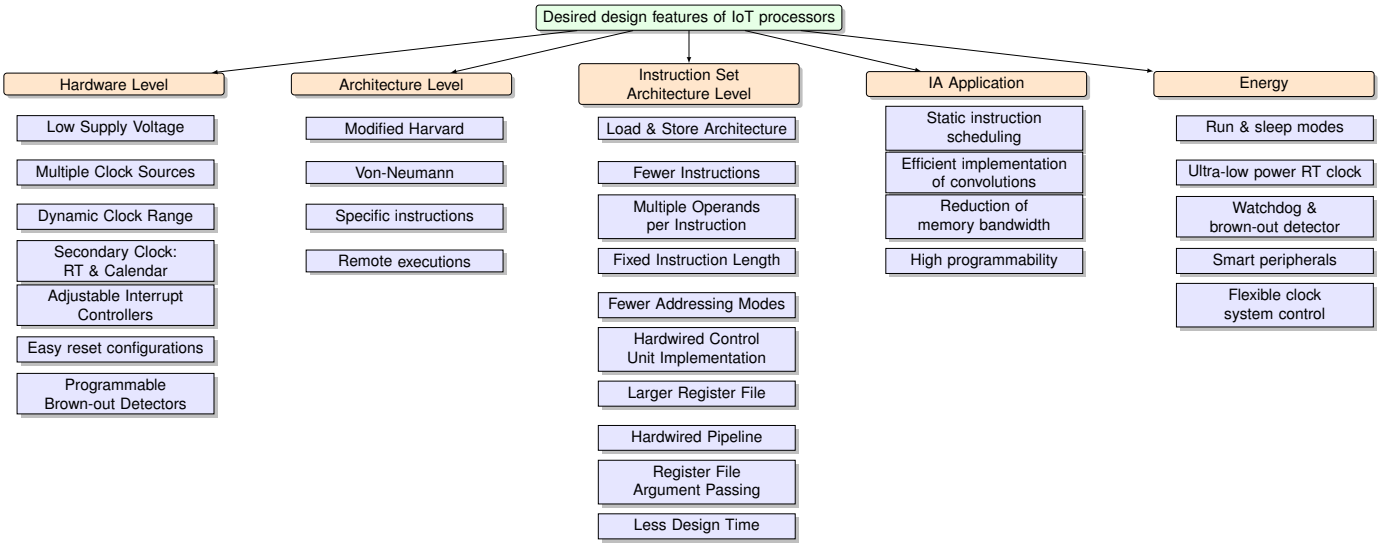

Fig. 2. Design features for IoT microprocessors

and energy efficiency [7]. Thus, processing data on the connected object requires a powerful processing unit. The figure 2 presents the features to choose the right microcontrollers depending in its characteristics. For example, the hardware design features sought in processors include: low power supply voltage, multiple clock sources and dynamic clock range, real-time and calendar running secondary clock, relocatable and programmable interrupt controllers, easy reset configurations, programmable power failure detectors. The IoT processors are built with Harvard or modified Harvard architecture, where the code memory can also be read out. So that, constant data can be stored therein over pure Von-Neumann architecture to accelerate the processing within constrained resources and leverage multiple types of memory such as cache memory [2]. Several semiconductor firms develop comprehensive IoT platforms around their core hardware component. The ARMs mbed IoT device platform offers modules as well as open source hardware and software blocks including an OS, cloud services and a development ecosystem with a complete layer of security for the management, core software encryption strategy and security of electronic communications [38]. IoT Pelion platform [39] supports all types of connections and IoT objects regardless of the type of data and cloud that can be public or private.

Edge computing performs real-time data processing before transmission directly on the node itself, relieving the processing load on the base station thanks to processors with specific features. With the emergence of new software solutions, Complex Event Processing (CEP) is an evolving paradigm for both IoT and Big Data. As event correlation solution built upon flexible and scalable Big Data techniques, CEP enables the calculation and processing of complex events [41]. It compares the data streams received from the sensors with matching data patterns. This type of solution allows the analysis, processing (reading, creation, transformation) and then correlation of numerous complex events in real time.

Artificial intelligence (AI) and digital applications involve sophisticated components of various technologies, including CPUs, GPUs and other variants for processors, given their high computational resource requirements.Among the proposed solutions, Google's Tensor Processing Unit (TPU) is a custom ASIC chip designed to speed up machine learning. It has the advantage of adapting its computing resources such as the speed of execution and memory resources according to the workload required for such algorithms [42]. This AI accelerator integrated circuit has prompted research to find other approaches to accelerate the speed of evolution and democratization of artificial intelligence-related uses, otherwise the challenges will be completely lost. ARM contributed to raise this challenge with processors dedicated to artificial intelligence. The architecture of these ARM'ML processors are designed to be efficient for large neural networks, particularly convolutional or recurrent ones. These deep learning oriented processors have been designed based on ARM Cortex-M, very low power chips with specific instructions for neural networks (one Cortex-M core per computing engine). The choice of a processor for this field of application need to consider four particular points: Static instruction scheduling, efficient implementation of convolutions, reduction of the bandwidth used in memory, and high programmability.

VI. Conclusion

The challenges for WSNs connected to the IoT are not only to aggregate data but also to ensure their security during the transfer, to process information within strict deadlines, to deliver precise decisions without human intervention. This requires a high-performance processor capable of processing intensive calculations with a limited energy budget that comes either from a low-capacity battery or harvesters whose solutions are not yet mature. Nowadays, designers work to make processors smaller, cheaper and more energy-efficient, to

incorporate multi-core technologies and to determine the best cores or configurations to incorporate into the microprocessor. Energy consumption rises to the forefront of system design challenges alongside performance specifications. This paper identifies the IoT microcontrollers reviewed regarding computing performance, computational complexity, time constraints, power requirements and fields of application in order to guide the researchers' choice towards the most suitable microprocessor configurations to achieve the optimization objectives and satisfy the design constraints for different applications. It investigated the energy techniques to prevent these networks from spraying power expenditures going from hardware to software optimization, to heterogeneous platforms. Potential core configurations and their characteristics, the future trend of IoT platform for particular applications such as IA are discussed.

REFERENCES

[1] B.-S. Kim, K.-I. Kim, B. Shah, F. Chow, and K. H. Kim, "Wireless sensor networks for big data systems," *Sensors*, vol. 19, no. 7, p. 1565, 2019.

[2] M. T. Banday, "A study of current trends in the design of processors for the internet of things," in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*. ACM, 2018, p. 21.

[3] Q. Huang and K. Rodriguez, "A software framework for heterogeneous wireless sensor network towards environmental monitoring," *Applied Sciences*, vol. 9, no. 5, p. 867, 2019.

[4] G. Selimis, L. Huang, F. Massé, I. Tsekoura, M. Ashouei, F. Catthoor, J. Huisken, J. Stuyt, G. Dolmans, J. Penders *et al.*, "A lightweight security scheme for wireless body area networks: design, energy evaluation and proposed microprocessor design," *Journal of medical systems*, vol. 35, no. 5, pp. 1289–1298, 2011.

[5] S. Furber, "Microprocessors: the engines of the digital age," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2199, p. 20160893, 2017.

[6] Y. Bhatt and C. Bhatt, "Internet of things in healthcare," in *Internet of things and big data technologies for next generation HealthCare*. Springer, 2017, pp. 13–33.

[7] O. Kanoun, T. Keutel, C. Viehweger, X. Zhao, S. Bradai, S. Naifar, C. Trigona, B. Kallel, I. Chaour, G. Bouattour *et al.*, "Next generation wireless energy aware sensors for internet of things: A review," in *2018 15th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2018, pp. 1–6.

[8] S. Khriji, D. El Houssaini, I. Kammoun, and O. Kanoun, "Energy-efficient techniques in wireless sensor networks," in *Energy Harvesting for Wireless Sensor Networks: Technologies, Components and System Design*. De Gruyter Oldenbourg, 2018.

[9] A. González, "Trends in processor architecture," in *Harnessing Performance Variability in Embedded and High-performance Many/Multi-core Platforms*. Springer, 2019, pp. 23–42.

[10] T. Guérout, T. Monteil, G. Da Costa, R. N. Calheiros, R. Buyya, and M. Alexandru, "Energy-aware simulation with dvfs," *Simulation Modelling Practice and Theory*, vol. 39, pp. 76–91, 2013.

[11] R. Chéour, M. W. Jmal, and M. Abid, "New combined method for low energy consumption in wireless sensor network applications," *SIMULATION*, vol. 94, no. 10, pp. 873–885, 2018.

[12] C. M. Kamga, "Cpu frequency emulation based on dvfs," *ACM SIGOPS Operating Systems Review*, vol. 47, no. 3, pp. 34–41, 2013.

[13] C.-H. Chou, L. N. Bhuyan, and D. Wong, "μdpm: Dynamic power management for the microsecond era," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2019, pp. 120–132.

[14] W. Lau, R. Cherukuri, Y. Firstenberg, D. Louie, G. Anand, and K. M. Woley, "Batching notifications to optimize for battery life," Mar. 24 2015, uS Patent 8,990,295.

[15] D. Wang, T. Abdelzaher, B. Priyantha, J. Liu, and F. Zhao, "Energy-optimal batching periods for asynchronous multistage data processing on sensor nodes: foundations and an mplatform case study," *Real-Time Systems*, vol. 48, no. 2, pp. 135–165, 2012.

[16] P. K. Dutta and D. E. Culler, "System software techniques for low-power operation in wireless sensor networks," in *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*. IEEE Computer Society, 2005, pp. 925–932.

[17] M. Arora, S. Manne, I. Paul, N. Jayasena, and D. M. Tullsen, "Understanding idle behavior and power gating mechanisms in the context of modern benchmarks on cpu-gpu integrated systems," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2015, pp. 366–377.

[18] C. You and K. Huang, "Exploiting non-causal cpu-state information for energy-efficient mobile cooperative computing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4104–4117, 2018.

[19] S. Khriji, D. El Houssaini, I. Kammoun, and O. Kanoun, "A fuzzy based energy aware unequal clustering for wireless sensor networks," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2018, pp. 126–131.

[20] A. El Kouche, A. Alma'aitah, H. Hassanein, and K. Obaia, "Monitoring operational mining equipment using sprouts wireless sensor network platform," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2013, pp. 1388–1393.

[21] H.-C. Lee, "Towards a general wireless sensor network platform for outdoor environment monitoring," in *SENSORS, 2012 IEEE*. IEEE, 2012, pp. 1–5.

[22] J. Lee, Z. Zhong, B. Du, S. Gutesa, and K. Kim, "Low-cost and energy-saving wireless sensor network for real-time urban mobility monitoring system," *Journal of Sensors*, vol. 2015, 2015.

[23] H.-L. Shi, "Development of an energy efficient, robust and modular multicore wireless sensor network," Ph.D. dissertation, Université Blaise Pascal-Clermont-Ferrand II, 2014.

[24] T. Gomes, F. Salgado, A. Tavares, and J. Cabral, "Cute mote, a customizable and trustable end-device for the internet of things," *IEEE Sensors Journal*, vol. 17, no. 20, pp. 6816–6824, 2017.

[25] A. Engel and A. Koch, "Heterogeneous wireless sensor nodes that target the internet of things," *IEEE Micro*, vol. 36, no. 6, pp. 8–15, 2016.

[26] J. Valverde Alcalá, A. Rodríguez Medina, J. Mora de Sambricio, C. Castañares Franco, J. Portilla Berrueco, E. d. l. Torre Arnanz, and T. Riesgo Alcaide, "A dynamically adaptable image processing application trading off between high performance, consumption and dependability in real time," pp. 1–2, 2014.

[27] K. Shahzad, "Energy efficient wireless sensor node architecture for data and computation intensive applications," Ph.D. dissertation, Mid Sweden University, 2014.

[28] C. J. Deepu, C.-H. Heng, and Y. Lian, "A hybrid data compression scheme for power reduction in wireless sensors for iot," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 2, pp. 245–254, 2017.

[29] L. Adnan, Y. Yussoff, H. Johar, and S. Baki, "Energy-saving street lighting system based on the waspmote mote," *Jurnal Teknologi*, vol. 76, no. 4, 2015.

[30] J. Wang, B. Yun, P. Huang, and Y.-A. Liu, "Applying threshold smote algoritwith attribute bagging to imbalanced datasets," in *International Conference on Rough Sets and Knowledge Technology*. Springer, 2013, pp. 221–228.

[31] crossbow, "Mica2, . https://www.eol.ucar.edu/isf/facilities/isa/internal/crossbow/datasheets/mica2.pdf," 2002.

[32] ——, "Crossbow, rev b, www.willow.co.uk/telosb_datasheet.pdf," 2005.

[33] MEMSIC, "Iris, rev a, www.memsic.com/userfiles/files/datasheets/wsn/iris_datasheet.pdf," 2007.

[34] "Virtenio. 2.4 ghz funkmodul "preon32" mit überlegener technik http://www.virtenio.com/de/produkte/hardware/preon32.html." 2011.

[35] S. overview, "Wisense, http://www.wisense.in/api/html/systemoverview.html," 2014.

[36] R. Chéour, S. Khriji, D. El Houssaini, M. Baklouti, M. Abid, and O. Kanoun, "Recent trends of fpga used for low-power wireless sensor network," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 10, pp. 28–38, 2019.

[37] R. Fisher, L. Ledwaba, G. Hancke, and C. Kruger, "Open hardware: A role to play in wireless sensor networks?" *Sensors*, vol. 15, no. 3, pp. 6818–6844, 2015.

[38] M. V. Shenoy, "Sensor information processing for wearable iot devices," in *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*. Springer, 2020, pp. 177–200.
[39] F. Update, "Arm update," 2018.
[40] T. Adegbija, A. Rogacs, C. Patel, and A. Gordon-Ross, "Microprocessor optimizations for the internet of things: A survey," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 7–20, 2017.
[41] C. Andrade, J. Correia, C. Costa, and M. Y. Santos, "Intelligent event broker: A complex event processing system in big data contexts," 2019.
[42] N. Jouppi, "Google supercharges machine learning tasks with tpu custom chip," *Google Blog, May*, vol. 18, 2016.