

Edge Computing: A Survey On the Hardware Requirements in the Internet of Things World

Original

Edge Computing: A Survey On the Hardware Requirements in the Internet of Things World / Capra, Maurizio; Peloso, Riccardo; Masera, Guido; Roch, Massimo Ruo; Martina, Maurizio. - In: FUTURE INTERNET. - ISSN 1999-5903. - ELETTRONICO. - 11:4(2019), pp. 100-124. [10.3390/fi11040100]

Availability:

This version is available at: 11583/2731661 since: 2019-04-29T10:26:29Z

Publisher:

MDPI

Published

DOI:10.3390/fi11040100

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Review

Edge Computing: a survey on the hardware requirements in the Internet Of Things world

Maurizio Capra, Riccardo Peloso, Guido Masera, Massimo Ruo Roch and Maurizio Martina

M. Capra, R. Peloso, G. Masera, M. Ruo Roch and M. Martina are with the Department of Electrical, Electronics and Telecommunication Engineering, Politecnico di Torino, Torino, TO, 10100 Italy

* Correspondence: maurizio.capra@polito.it riccardo.peloso@polito.it guido.masera@polito.it massimo.ruoroch@polito.it maurizio.martina@polito.it

Version April 12, 2019 submitted to Future Internet

Abstract: In today's world, ruled by a great amount of data and mobile devices, cloud-based systems are spreading all-over. Such phenomenon increases the number of connected devices, broadcast bandwidth, and information exchange. These fine-grained interconnected systems, which enable the Internet connectivity for an extremely large number of facilities (far beyond the current number of devices) go by the name of Internet of Things (IoT). In this scenario, mobile devices have an operating time which is proportional to the battery capacity, the number of operations performed per cycle and the amount of exchanged data. Since the transmission of data to a central cloud represents a very energy-hungry operation, new computational paradigms have been implemented. The computation is not completely performed in the cloud, distributing the power load among the nodes of the system, and data is compressed to reduce the transmitted power requirements. In the edge-computing paradigm, part of the computational power is moved toward data collection sources, and only after a first elaboration collected data are sent to the central cloud server. Indeed, the "edge" term refers to the extremities of systems represented by IoT devices.

This survey paper presents the hardware architectures of typical IoT devices and sums up many of the low power techniques which make them appealing for a large scale of applications. An overview of the newest research topics is discussed, besides a final example of a complete functioning system, embedding all the introduced features.

Keywords: Edge computing; Internet of Things(IoT); MicroController(MCU); low power; embedded system

1. Introduction

The Internet of Things (IoT), term coined in 1999 by Kevin Ashton, is gaining more and more attention in these years due to the increasing amount of connected devices and consequently to the amount of data. In the big data era, recording data from several environments and users is extremely valuable from a statistic as well as a business and economic point of view. Nowadays, almost every device present in everyday life presents some embedded electronics, which turns it into a potential IoT node. Indeed, IoT nodes are able to sense information and transmit it, thanks to a communication interface.

So far the IoT paradigm had a huge impact on both consumers lives and business models, due to the decreasing cost of implementation of these devices and the increasing demand. The trend is expected to rapidly increase, as shown in Figure 1. Gartner [1] (world's leading company in research and advisory fields) states that 23.14 billion of connected devices have been produced in the past year (2018), up to 30.73 billion are expected for 2020. This represents a great opportunity for investors, producers and companies to collect big data. In fact, companies are expected to spend around 5 trillion dollars in 2021

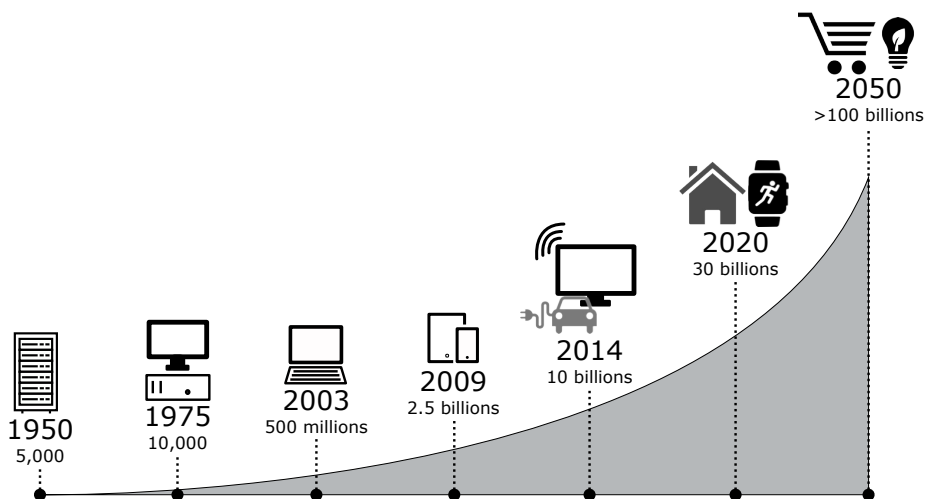


Figure 1. Expected adoption growth of IoT devices

to expand the market [2] and introduce new applications, embedded in everyday gadgets.

In a highly dynamic scenario, as depicted above, the opportunities to diversify the possible solutions and applications are many. Besides, there still exists a main common factor: the hardware implementation. Indeed, hardware architectures are quite similar regardless of the final use since their organization relies very often on microcontroller-based platforms. Most part of IoT devices leans on batteries or energy harvesters. Given that their energy budget is limited, even the power that can be consumed will need smart energy management, driving the hardware engineering toward an ultra-low power approach. Limited power represents a huge constraint to many components of the architecture, especially the energy-hungry ones, like wireless transmitters. In such a case, communication and data to be broadcast must be reduced to the essential, which translates to low energy technology, such as Ultra-Wide Band (UWB) [3–8], and transmit only useful features by exploiting state of the art techniques, like Compressing Sensing.

IoT nodes have to sense and collect data with respect to their specific task. The tasks could be many, ranging from smart household appliance, sleep monitoring, physical activity tracking, caretaker condition monitoring etc. Such information, collected over thousands of individuals, must ensure absolute privacy for final users. Indeed, such personal data must be kept safe from outside attacks, avoiding any hacking attempt. Even localization-based services are affected by this problem. The position is part of a privacy policy that must be prevented from threats [9]. In order to protect data, the common procedure is to encrypt the communication so that nobody can steal precious info. Device security, together with encryption techniques are a matter of discussion and research widely spread around the globe.

Many of the aforementioned tasks, including the last one described above, can be performed directly on the platform without accessing the cloud or a remote hosting service. This considerably reduces the power needed to transmit and receive data before and after elaboration, relieving much of the effort from servers. This change of paradigm is called Edge Computing: part of the workload is decentralized and distributed among the IoT nodes, turning them from simple sensors into more powerful and smart embedded systems, capable of several new features.

Thanks to this innovative and effective approach, measured information can be further analyzed directly on the field, allowing for a more responsive application and a faster post-processing operation once data has been transmitted. Edge Computing paradigm is considered, from many, an environmentally friendlier alternative to the Cloud Computing one, due to its ability to restrain the volume of data to be moved, consequently cutting down the energy cost.

As described above, the decentralization of the workload is the focus of many works of research. The

67 aim is to reduce the latency by offloading some of the tasks on surrounding servers [10,11]. In fact
68 in urban environments is possible to rely on such infrastructure to enhance performances. However,
69 in order to correctly manage the workload, there is often the need to split it homogeneously and to
70 synchronize all the different duties. To achieve this goal, something more is required, such as an
71 Operating System (OS). The use of a complete OS instead of a limited embedded firmware becomes a
72 powerful tool when handling complex tasks. In order to manage multiple users and different scenarios,
73 required for IoT, the main OS can run virtualized operating systems (which emulates the entire
74 hardware resources required by an OS) or it can exploit the containerization paradigm, which let the
75 user have multiple instances of the OS running at the same time while sharing the kernel for resources
76 allocation. The containerization of tasks makes the organization and the allocation of workloads more
77 efficient, by exploiting the bandwidth of the system at its maximum, without exceeding constraints
78 [12]. For sake of clarity, since this current survey is focusing on hardware aspects, OS and software
79 related technologies are not discussed in the following sections; however, the reader can refer to the
80 provided references for further details.

81 This survey is intended to be an overview of the key aspects of IoT hardware platforms, designed
82 for Edge Computing. Several state of the art techniques, suitable for low power applications, are
83 introduced and discussed through real examples. Section 2 presents the motivations that drive the
84 continuous development of new architectures and techniques, with a special focus on the Edge
85 Computing approach, by describing and emphasizing its importance. Section 3 depicts the landscape
86 of a typical IoT node system, comprehensive of all its main features, ranging from the "brain"
87 (Central Processing Unit) to its peripherals. First, Input/Output (IO) systems are presented as the
88 main communication channels of the system (3.1); a detailed analysis of several IO types is listed,
89 providing pros and cons for every possible choice. Then, memories are introduced as fundamental
90 elements for data retention. Even in this case, several solutions are presented, with a special focus on
91 non-volatile memories (3.3) (memory elements in which data are retained even in case of power failure
92 or interruption).

93 Power management (Section 3.4) is responsible for deciding which parts of the system should be turned
94 off, when they are detected as not useful for the current task, or even for adjusting the local power
95 supply parameters, such as the voltage level. After this unit, near-threshold behavior is described (3.5)
96 in order to better understand what are the consequences of power supply parameters tuning.

97 Since Edge Computing heavily relies on data processing, an entire section is dedicated to this topic
98 (Section 3.6). State of the art algorithms and techniques able to reduce significantly the amount of data
99 to be transmitted are discussed.

100 Section 3.8 is completely devoted to the Central Processing Unit (CPU), explaining the evolution
101 of single-core architectures and the reason for adopting the multi-core paradigm. Indeed, most
102 part of modern IoT nodes is equipped with powerful processors, able to effectively distribute the
103 computational load among the cores and, consequently, level out the power dissipation.

104 Finally, a real platform, which sums up all the above-listed features, is presented in Section 4, while
105 Section 5 concludes the paper by providing new challenges and future perspectives.

106 *1.1. Methodology and Organization*

107 This section is intended to explain the review methodology, which this paper is based on.
108 Researchers have at their disposal several different approaches [13] to collect and summarize the
109 state of art literature on a specific topic. A review paper is the result of five generic steps:

- 110 1. topic and objectives definition;
- 111 2. primary search;
- 112 3. information refinement and secondary search;
- 113 4. data retrieving;
- 114 5. analyzed data presentation.

115 Namely, once the main subject of the manuscript is defined, primary search is fundamental in order
116 to create a pool of articles from which the topic is clearly presented and the reader can figure out
117 the principal aspects. Then, a refining process is necessary in order to discard loosely related articles
118 or misleading essays. Secondary search is intended to integrate the current information pool and to
119 deepen some important points. Once the ensemble of articles has been consolidated, data and topic
120 are extracted, reworked and presented.

121 All the aforementioned points seem to be sequential and to be followed in a linear path, but the
122 review process is strongly iterative, going back and forth through stages many times, before reaching a
123 satisfactory result [14].

124 The methodology chosen for this paperwork is the narrative review, that is a very traditional way of
125 reviewing contemporary literature. It consists of a summary of the found material, which depicts a
126 quasi-general overview of the topic. Indeed, authors can decide to focus more on a certain aspect than
127 others. As a matter of fact, this paper deals with IoT and edge computing, with special attention to
128 hardware features.

129 In particular, several efforts have been spent to ensure a rigorous approach. The one pursued in the
130 following sections relies on the one developed by Levy et al. [15], which includes these three steps:

- 131 1. literature collection and screening;
- 132 2. data extrapolation and examination;
- 133 3. composition of the review script.

134 Thus, this current overview paper presents a narrow and focused spectrum, as allowed by the flexibility
135 of the narrative review approach.

136 2. Definitions and Motivations

137 The number of IoT devices is constantly increasing for several different reasons: low production
138 cost, pervasive electronics, and technology in daily life, availability of wireless and wired
139 communication networks, etc..

140 Since these devices are spreading all over, ethic and practical questions had been posed related to
141 energy footprint and sustainability. Indeed IoT systems are often battery powered devices able to
142 acquire and send data via wireless transmission. This means, from an architectural point of view, that
143 these devices must be energy efficient, thus requiring low power solutions.

144 In order to meet all these constraints, IoT systems must be composed of low power sensors, such
145 as MicroElectroMechanical Systems (MEMS), while data are elaborated by means of low power
146 MicroController Units (MCU) and then transmitted. Generally, data are sent as once-in-a-while
147 packets, by exploiting low energy radio transmitters/receivers based on new technologies, such as the
148 UWB communication.

149 Since transmission is the most power consuming task among the three steps described above, many
150 techniques have been developed in the past years to make it as low-power as possible. One of the
151 most promising techniques, compressive sensing (described in section 3.7), represents a very efficient
152 solution, which allows to directly reduce the amount of data collected by sensors. As a consequence,
153 the power needed to transfer data to the cloud for post-processing is much smaller than conventional
154 Nyquist-rate sampling. The energy budget for IoT nodes usually consists of small battery sources or
155 energy harvesters; thus, the ability to reduce the amount of data to be transmitted (and consequently
156 the power) plays a key role in today's systems. Alternative approaches have been proposed in the
157 literature to reduce the rate of data to be transmitted, by relying on bio-inspired techniques.

158 Many of modern MCU architectures cannot deal with the power budget imposed by IoT nodes, but
159 even in this strict scenario, researchers developed a broad spectrum of energy saving approaches,
160 such as drastic voltage and frequency scaling. Near-threshold operation of embedded transistors
161 introduces limitations such as performance degradations but still, for certain applications, it represents
162 a very promising solution. Sometimes, for some particular cases, a dedicated hardware approach

163 such as Application Specific Integrated Circuits (ASICs) 2 could be preferred in order to avoid the
 164 power consumption of the general purpose MCUs. The main drawback is the loss of flexibility of
 165 the architectures. This is overcome by the use of semi-specific processors, called Application Specific
 166 Integrated Processors (ASIPs [16]) 2, that are designed so that recurrent application-specific operations
 167 (for example convolution or bit-wise operations) are accelerated with dedicated hardware. The overall
 168 processor is not as specifically tailored on the whole application as in ASICs in order to remain flexible
 169 and future-ready.

170

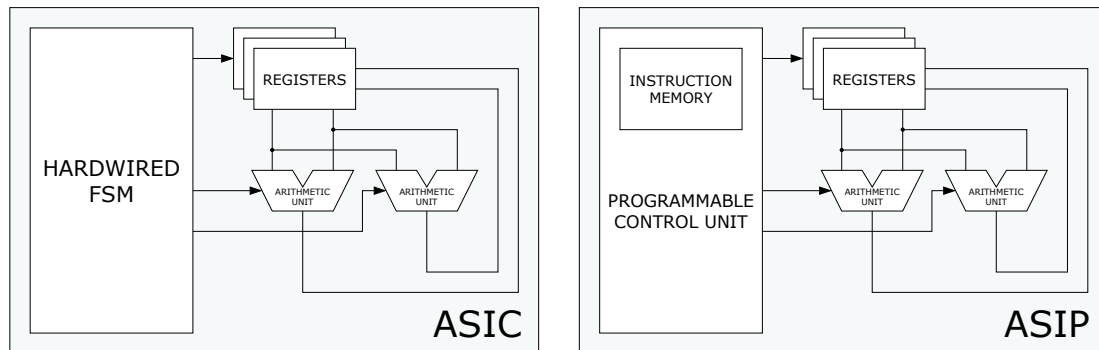


Figure 2. Comparison between ASIC and ASIP architecture.

171 This approach leads to processors which usually require higher power than ASICs but are able
 172 to adapt to the evolution of standards and technologies without re-designing and producing a new
 173 integrated circuit, which cost is a major limitation for the IoT. Ideally, IoT hardware should be at the
 174 same time optimized but not too application-specific in order to cut the non-recurrent manufacturing
 175 expenses by producing millions of small and cheap integrated circuits. This is a very difficult task, as
 176 explained in the following sections. These constraints drive IoT manufacturers towards older silicon
 177 technology processes, from 200 nm to 130 nm. These particular nodes are very effective in this field
 178 as they have been proven to be mature, cheap, mixed-signal capable, embedded-flash capable, low
 179 power and, thus, they are considered a low-risk option. Smaller technological nodes are still needed
 180 for high-performance applications, where very fast execution is required.

181 3. Ultra-Low-Power MCU Architectures

182 This section introduces the main hardware components that compose a typical IoT node, or more
 183 in general, a typical electronic system. Taking into account the power and energy requirements and
 184 constraints introduced in Section 2, the following description is oriented to low power solutions.
 185 The main component is the Central Processing Unit (CPU), typically on board of an MCU, equipped
 186 with some memory. These two component are tightly coupled since the former elaborates data coming
 187 from the latter. In order to make things easier and faster, the memory is organized in a hierarchical
 188 way, in which smaller and faster components (such as flip-flops) are directly integrated into the CPU,
 189 while larger and slower components (such as flash memories) transfer data towards the faster ones.
 190 Usually, data to be saved into the memory come from peripherals that are represented by sensors,
 191 transceiver modules for communication or, more in general, by connectors.
 192 The steps executed in an embedded application can be roughly represented by fetching data
 193 from sensors, elaborate them via the MCU and finally, transmit the results by adopting wireless
 194 communications, like Bluetooth [17], Mobile Networks, Wi-Fi [18], Zigbee [19], Z-Wave [20], LoRaWAN
 195 [21], custom transmission protocols (like IR-UWB [22]) and Light-based systems as [23,24]. Since this
 196 sequence of tasks is performed periodically, energy savings can be obtained by switching off key
 197 electronics components when not needed. This process is called duty cycling and consists of waking

198 up the device only when it has to perform a task, while the rest of the time it operates in a deep sleep
199 mode. During the wake-up phase, generally triggered by an external event, the MCU status is restored,
200 together with the power supply and the clock signal. Once the task is performed, the MCU saves its
201 current state before transitioning again in the low power state.

202 This behavior is predictable and so the main system can wake up and put to sleep the various actors,
203 including itself, by resorting to well-defined power management techniques.

204 A general system architecture and organization has been depicted above; now, in order to better
205 understand the details, the following sections will analyze each component listed above. First of all,
206 peripherals will be explained by giving examples of the newest technology in use today. Then the
207 paper will focus on memory types, power management techniques, and data processing, respectively.
208 A final section is devoted to CPU architecture and its evolution through the years.

209 *3.1. IO Architecture*

210 In IoT devices, peripherals are fundamental in order to connect several sensors or external
211 devices. MCUs are equipped with some serial interfaces such as UART (Universal Asynchronous
212 Receiver/Transmitter) [25], SPI (Serial Peripheral Interface) [26] and I2C (Inter-Integrated Circuit) [27].
213 However, currently, also high bandwidth connections are required, like USB (Universal Serial Bus).
214 Generally, the CPU has the control of the peripherals, handling associated events and data transfer,
215 but since systems are becoming more and more complex, peripheral subsystems have become smarter,
216 having, sometimes, their own control unit. Peripherals can run even while the CPU is in deep sleep
217 mode and they are able to manage their own power status independently of the CPU, by switching off
218 unused parts. A peripheral can be woken up by events, thus, data transfer can take place without any
219 action from the CPU, thanks to dedicated memory management units.

220 A not negligible amount of energy is required by off-chip communications, especially for high-speed
221 interfaces, like Double Data Rate (DDR) dynamic RAM memories, wired data transfer (like USB and
222 Ethernet) and video interfaces. IoT nodes tend to be tailored to their application, optimization is indeed
223 the key to create systems able to survive for decades with an extremely low power budget. Nevertheless,
224 there are still applications where high performances are required. This translates in the necessity of
225 low power standard interfaces. The key player in the definition of these standards is the MIPI
226 (Mobile Industry Processor Interface) Alliance [28], founded in 2003 by Samsung, Nokia, Intel,
227 Texas Instruments, STMicroelectronics, and ARM. MIPI interfaces are optimized for low power,
228 high bandwidth and low electromagnetic interferences. The MIPI Alliance works on defining
229 standards for the physical layer (PHY), protocols for multimedia (cameras, displays, audio and
230 touch peripherals), chip-to-chip and inter-processor communications, management of low-speed
231 devices, power management, debugging tools and software integration. As an example, the latest
232 MIPI Display Serial Interface (MIPI-DSI-2) is able to handle very high resolution displays, also thanks
233 to video compression, by reducing the power spent for the transmission of data to the screen and so
234 off-loading the reconstruction of the video stream to the display controller. Another example of MIPI
235 low power, high throughput design is the I3C interface, presented as the successor of the I2C. I3C
236 features a high clock speed and can work at double data rate regime; moreover, it features high power
237 efficiency with respect to its predecessor, as shown in figure 3.

238 If no inter-chip communications are required, the only possible power saving technique for peripherals
239 comes from the optimization of the peripheral itself. In particular mixed signal circuits, analog
240 to digital converters (ADC) and digital to analog converters (DAC), are required in almost every
241 embedded application to translate the physical world measure (usually analog) to the digital domain
242 and vice-versa. While digital circuits are inherently robust against continuous time noise, analog
243 circuits suffer from voltage and transistor size scaling. These aspects limit the maximum excursion
244 of the input signal and affect the linearity of active analog parts, like operational amplifiers. Several
245 researchers have struggled to improve the performances, even with the IoT constraints, by reaching a
246 sub-femtoJoule-per-conversion-step Figure of Merit (FOM) [29–34]. Absolute FOM alone is not always

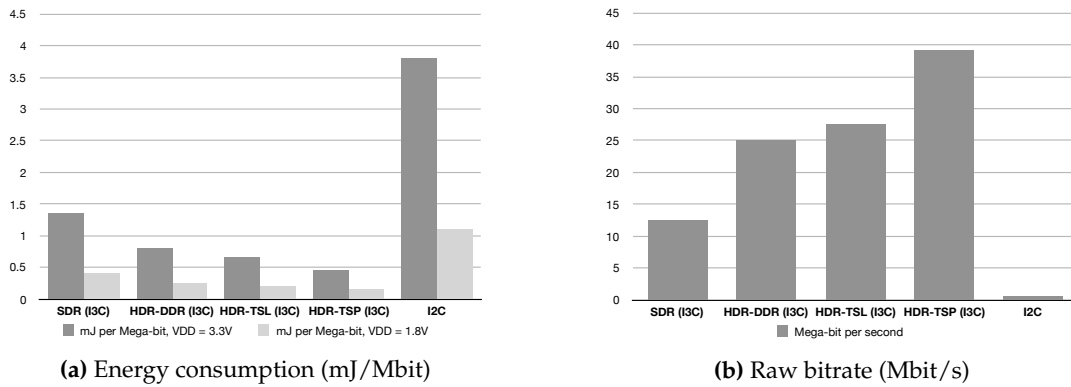


Figure 3. MIPI I3C bus efficiency and throughput

247 the best indicator for low power conversion systems, as not all input signals require the same precision
 248 or speed (this aspect is addressed in Section 3.7). Furthermore, it is important to tailor a converter
 249 around the specific signal requirements to achieve the best power performances. As stated by Alioto
 250 [35], only a few physical signal types require more than 16 bits and, usually, 8 bits are sufficient for
 251 low resolution applications. Also, the converter speed can vary a lot depending on the application:
 252 from the low speeds of heart rate and temperature readings to the high speeds required by imaging
 253 peripherals. A huge amount of information redundancy (spatial and temporal) is required for classic
 254 video stream-based algorithms. Indeed, sensor-level compressive sensing for video capture will be an
 255 important step towards low power video applications [36–41].

256 3.2. Communication and security

257 This section deals with Internet communication systems and security issues arising from the IoT
 258 paradigm. Many standards have emerged in recent years to cope with the low power requirements
 259 of IoT nodes. Depending on the application, in particular on the required data rate and data range,
 260 it is possible to select the best transmission technology. Sometimes, it is possible to create a local
 261 network and to use an aggregator for sending the data to the cloud for further processing or for
 262 being stored. However, future IoT nodes will heavily depend on cellular communication, especially
 263 with the forthcoming 5G, as the prominent player [42,43]. Indeed, 5G aims to be a revolution for
 264 machine-to-machine communication, in particular in situations which were not optimally handled by
 265 previous cellular communication systems. In particular, 5G communications can be tailored around
 266 the application to improve reliability, reduce latency and energy consumption, and to increase device
 267 density. The adoption of 5G in new IoT platforms promises to enable a true pervasive fully-connected
 268 era. However, always connected devices through wireless connections can be prone to external hacking
 269 attempts, which is a major issue when dealing with sensible data and/or dangerous situations [44–49].
 270 Attacks can involve sensors nodes to collect precious data from users, which would otherwise be
 271 unavailable for privacy or secrecy reasons [50]. These data could be used for analysis purposes or to
 272 profile users. On the other hand, attacks can involve actuator-based systems, like Autonomous Electric
 273 Vehicles [51–54], or Microgrids (small electrical sources able to better distribute the electrical power
 274 through the load) [55,56] or healthcare devices (like implantable cardiac devices) or literally every
 275 electrical item that will potentially be equipped with a network access. For this reason, a substantial
 276 amount of research is devoted to ensure connection security and to avoid eavesdropper stealing
 277 precious data. IoT devices will be built with hardware cryptographic accelerators for optimizing power
 278 consumption and latency of data transmission. These hardware accelerators can (and should) also
 279 be used for anti-tamper protection and to circumvent IP stealing attempts, encrypting the data in
 280 inter-chip communications to null reverse engineering efforts.

281 3.3. Non-volatile Memories

282 A big problem in edge computing is data retention
 283 during the idle mode. Current technologies do not cope
 284 well with the low voltages used during power gating and
 285 voltage scaling as the memories used are usually volatile,
 286 like Dynamic and Static RAM (DRAM and SRAM) and
 287 internal Flip-Flops (FFs) or Latch-based registers. These
 288 traditional technologies can work at high speed and are
 289 relatively easy to manufacture and integrate, but they
 290 have the disadvantage that information is lost under a
 291 certain supply voltage value. Actually, DRAM cells tend
 292 to lose the information even when fully powered due to
 293 the leakage through the cell capacitance, thus requiring cell
 294 refresh mechanism which is energy and time consuming.
 295 Even though non-volatile integrated memories exist, like
 296 Erasable Programmable ROM (EPROM) and Flash memories, these technologies are usually slow,
 297 especially during the writing stage. This can be a serious problem, indeed, with volatile memories, part
 298 of the computation core cannot be shut down as the information has to be retained. This bottleneck is
 299 driving a lot of research towards new non-volatile memories, which have to be easily integrable, fast,
 300 small, reliable and cheap. The main new technological approaches are:

- 301 • Resistive RAMs (ReRAMs) [57–63], which store the information as the variation of resistivity
 302 of a thin oxide film. A current is injected in the oxide to change its structure and to modify its
 303 resistance value. It is possible to program one cell to high or low resistance and so to assign a
 304 logic value to each of these two states. This technology is compatible with the current CMOS
 305 process. Moreover, it can achieve switching speeds of up to 10 ns and it features multilevel
 306 capability. However, the current required to reset the oxide state is high, usually being difficult
 307 to integrate in the circuit.
- 308 • Ferroelectric RAMs (FeRAMs) [64–66], which work like Dynamic RAMs but store the information
 309 in a ferroelectric layer instead of a dielectric one. The technology can be compatible with DRAM
 310 process, but it is usually built on old processes (350 to 130 nm). Besides, this type of memory
 311 consumes power only to read or write the memory cell, which drastically lowers the consumption
 312 with respect to DRAMs. The technology is intrinsically fast, it takes about 1 ns to modify the state
 313 of the layer, indeed. Usually, the bottleneck is the electronic control, which is rather complex, like
 314 in DRAMs.
- 315 • Phase-Changing RAMs (PCRAMs) [67–72], in which a chalcogenide glass can change phase from
 316 amorphous to crystalline. Moreover, chalcogenide glass can hold also an intermediate state,
 317 allowing for multilevel storage. However, the cells are difficult to program, so their use is still
 318 limited. These devices are faster than Flash-based memories, in particular for writing operations
 319 as PCRAMs are feature the possibility to modify each cell individually. The drawbacks are that
 320 the cells are prone to aging (even if they are better than Flash memories) and they are susceptible
 321 to temperature variations.
- 322 • Magnetic RAMs (MRAMs) [73–81], which use electron spin to store information. Currently,
 323 MRAMs look a very promising solution and several researchers envision that they might replace
 324 both the main memory and the storage memory in future architectures. When reading an MRAM,
 325 a current is forced to flow near the magnetic material, and the reading operation is accomplished
 326 by sensing the polarization of the magnetic field. When writing, an external current needs to
 327 overcome the stored field to impose a new value. As a consequence, writing requires more power
 328 consumption than reading. This technology can compete with Static RAM cells speed, while
 329 presenting a much lower area utilization.

330 Nowadays, when dealing with voltage scaling and power gating, the information contained in
 331 a Flip-Flop (FF) is retained thanks to the use of a Non-Volatile Flip-Flop (NVFF) cell [81–95], which

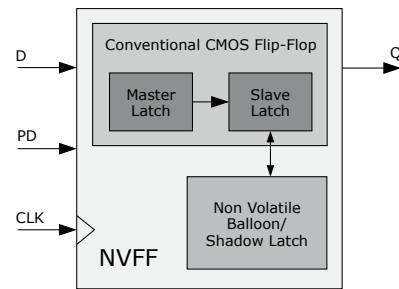


Figure 4. Non-Volatile Flip-Flop with shadow latch

332 consists on a FF helped by a balloon latch (sometimes called shadow latch) circuit that works with
 333 true ground and power supply to retain the logic level inside the FF, as shown in figure 4. Despite
 334 this approach works as required, it also increases leakage with respect to true non-volatile memories
 335 and it still requires the availability of the power supply. Aside from mass storage applications, the
 336 new technologies listed above can be integrated into the balloon part of the NVFF to make it truly
 337 non-volatile, as reported by [96].

338 3.4. Power Management

339 Power management is a feature of many CPUs and it consists in turning off or switching to a low
 340 energy regime parts of the core, peripherals or even sections of the memory hierarchy.
 341 There are many reasons to perform such optimization:

- 342 • to reduce the power consumption by excluding elements that are not involved in the current
 343 task;
- 344 • to enhance the lifetime of the battery and consequently of the embedded system;
- 345 • to tone down the noise produced by all the components forming the system;
- 346 • to reduce the effort and requirements of the cooling apparatus.

347 Since usually IoT nodes are represented by mobile, wearable battery-based devices, having an onboard
 348 unit able to dynamically control the energy consumption is extremely precious. Furthermore, since
 349 algorithms and tasks are performed in a sequential fashion, not all the units will be used at the same
 350 time. As a consequence, switching off these parts becomes essential for the above-listed reasons.

351 MCUs are already equipped with some low power modes, that consist in turning the entire system in a
 352 suspended state, in which peripherals can work independently, while no operation is performed by the
 353 core. This situation is usually referred to as deep-sleep status. MCUs can enter this state when no task
 354 has to be performed, and, typically, the wake-up signal is produced by an internal timer (deep-sleep
 355 for a known period) or by an external interrupt (event wake up).

356 The aforementioned mode has become very popular in the embedded systems community, despite it is
 357 not always effective with some of nowadays benchmarks. Indeed, entering and exiting the deep-sleep
 358 mode comes at certain power cost, so, depending on the application, it is not always advantageous.

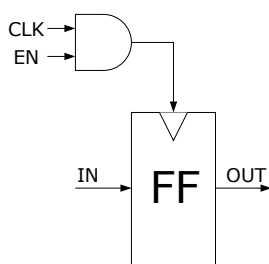


Figure 5. Clock gating: instead of using an enable signal, the clock is directly disabled

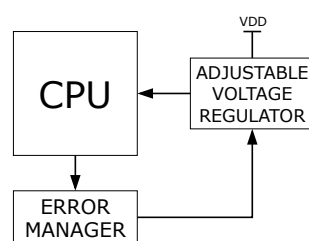


Figure 6. Dynamic voltage scaling: a closed loop decides the core voltage power supply

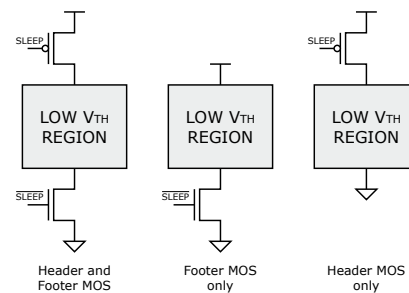


Figure 7. Power gating: regions of the circuit are decoupled by transistors from the true supply rails

359

360 Many other low power techniques exist, such as clock gating 5, that consists in stopping the clock
 361 from the part of the circuit that is not necessary to the current task. Since the dynamic power is related
 362 to the internal activity of combinational circuitry, no transition occurs by disabling the clock. However,

363 leakage current still persists, but, since in general the dynamic power is greater than the static one, the
364 energy saving will be consistent. Furthermore, clock gating intrinsically retains the state of the circuit,
365 allowing to restore normal operation by just reasserting the clock. The hardware overhead needed to
366 control the clock signal is negligible, and thanks to its fine granularity this solution turns out to be
367 very effective. Besides, leakage power can be reduced by combining clock gating and dynamic voltage
368 scaling, as explained in Section 3.5. However, despite a small overhead is required in the power supply
369 management unit, the time needed to restore the normal state of the circuit is greater as a stable supply
370 voltage is required to have the circuit behaving correctly.

371 The best approach to reduce power consumption is power gating, in which the supply voltage is
372 disconnected from the circuit. In this case, there is no dynamic power consumption and only a small
373 leakage power is present in current CMOS technology. While the hardware overhead is negligible, as
374 in clock gating, the time to restore the previous status of the circuit could be significant. This aspect
375 must be kept into account since it may affect system performance. In particular, the power gating
376 mode must be entered and exited in a safe way, in order to avoid damaging the circuit and to ensure a
377 correct behavior.

378 As depicted in figure 7, the hardware overhead of power gating consists of an MOS transistor to be
379 applied between the logic circuit and the supply line. Generally, a header and a footer are applied in
380 order to completely insulate the circuit. MOS transistors behave like switches, namely, when the MOS
381 transistor is open, the voltage is no longer applied to the circuit. As a consequence, the circuit is in a
382 frozen state and no power dissipation occurs.

383 3.5. Near-Threshold MCU Architectures

384 Generally, MCUs work with a power supply voltage well above the threshold voltage of transistors.
385 Nevertheless, the power supply voltage can be scaled during deep sleep mode in order to reach
386 sub-threshold condition, as shown in Fig. 6. Indeed, smart voltage scaling to make transistors working
387 in the near-threshold region, leads to a new low-power era. However, making a circuit working in the
388 near-threshold region is a complex task as reliability problems and performance degradation can arise
389 due to several factors, including fabrication process variations.

390 Indeed, the behavior of a circuit can degrade due to sensitivity to process variation (i.e. channel
391 length, doping concentration, ecc.), voltage and temperature (PVT). PVT compensation requires special
392 circuits to work correctly.

393 It is worth noting that special circuits are embedded within the main circuit, so they are exposed
394 to the same conditions of the main circuit, i.e. aging, high temperature and current. In order to be
395 effective, these special circuits must probe the current and provide feedback, by adjusting the power
396 supply to prevent unpleasant problems such as meta-stabilities. Common implementations of probing
397 subsystems are canary circuits and razor flip-flops.

398 Canary circuit is a replica of the critical path that is monitored in order to adjust the power supply.
399 Though being a simple approach, it only provides information about global process variations, while
400 no local information can be evinced as well as local PVT conditions, this is due to the fact that the real
401 critical path is placed somewhere else.

402 Razor flip-flop approach relies on lowering the supply voltage until a critical point. Working so near to
403 the limit, errors can occur due to time constraints violation. However, no error is propagated thanks to
404 shadow flip-flops that can restore the correct value. Shadow flip-flops are scanned using a delayed
405 clock signal that preserves their integrity and errors are detected by comparing these value to the one
406 in the real critical path. Global and local variations are both considered in this solution since the device
407 works in borderline conditions. In order to correctly apply razor flip-flops feedback, the designer
408 must have access to the low-level circuit and this is not always possible, especially when dealing with
409 externally engineered cores with no additional information than the top level interface protocol.

410 In contrast to voltage supply control, transistor body biasing represents an alternative to PVT
411 compensation. The main advantage of this solution is that adjusting transistors threshold only modifies

412 the leakage component, whereas the above approaches impact both on leakage and dynamic power.
 413 Moreover, body biasing is very effective when working near-threshold, as it features a more efficient
 414 and simpler circuit for polarization of p-well and n-well regions than DC-DC regulators for the supply
 415 voltage.

416 3.6. Data Processing

417 The term Data Processing is generally used to indicate the collection and then the manipulation of
 418 data in order to extrapolate meaningful information. Sometimes it can also indicate the transformation
 419 of data in an easy to handle format.

420 When related to hardware systems, data processing involves the CPU, as it must fetch data from
 421 memory or sensors, process them (generally through an application specific algorithm) and finally
 422 store the results in memory.

423 This data processing technique is important for IoT as it can significantly reduce the amount of data to
 424 store and transmit. IoT nodes can perform a pre-processing in-loco, discarding useless information,
 425 releasing part of the burden from the central unit that is in charge of performing the complete
 426 elaboration.

427 From a data processing point of view, energy saving can be obtained by increasing the parallelism,
 428 namely the amount of data per cycle that can be handled by the CPU or by optimizing the architecture
 429 with respect to the per-cycle power consumption. MCUs moved from initial 8 bit to 32 bit of today's
 430 most common devices (like ARM Cortex-M), but still engineers are focusing on reducing the power
 431 per instruction metric since in many application is not required to handle multiple data concurrently.
 432 In MCUs it is common to find optimizations for single cycle multiplication, like Multiply and
 433 Accumulate (MAC) dedicated instructions and arithmetic control support. From the instruction
 434 side, some improvements have been introduced as well, such as defining a set of instructions with
 435 reduced size (such as from 32 to 16 bit), saving energy in the storage and read process from memory.
 436 A very effective way to save power by processing data is through neuromorphic or quasi-digital
 437 approaches. As an example in [97] an Address Event Representation (AER) is adopted. This example
 438 relies on a neuromorphic event-based approach in which the only information to be transmitted is the
 439 source ID when a new event occurs. As a consequence, only when a change in the physical quantity
 440 is sensed, a certain amount of data is produced and sent. Indeed, AER [98,99] allows representing
 441 information very efficiently, by exploiting the neuromorphic approach to reduce the switching activity
 442 and thus the power needed to process and send acquired data is significantly lowered. Stemming from
 443 this concept, [97] proposes an architecture based on the openMSP430 processor that compared with its
 444 regular implementation can save up to 50% of power.

445 3.7. Compressive Sensing

446 As briefly introduced in Section 1, not only
 447 architectural optimizations but also algorithmic
 448 optimizations are required to efficiently exploit the
 449 available energy. Usually, the energy bottleneck in
 450 ultra-low power systems is the communication subsystem,
 451 which can be in certain applications a very power-hungry
 452 block, depending on the communication standard and
 453 the hardware architecture used. Many algorithms have
 454 been proposed to manage meshes of a large number of
 455 small interconnected sensor nodes in order to limit the
 456 radio range [100–108] but even in this scenario, it is not
 457 possible to just send the whole measured data to the cloud
 458 for post-processing. This is energetically too expensive and
 459 the burden of data to be processed is (and increasingly will

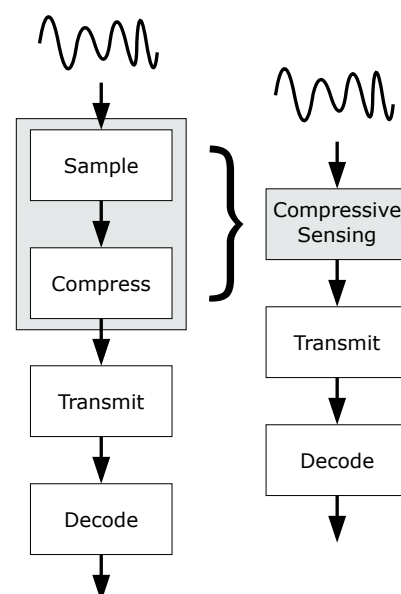


Figure 8. Compressed data can be directly obtained while acquiring the signal

460 be) a major problem in the IoT context. To cope with this
461 problem, a large research effort has been dedicated to the
462 compressive sensing paradigm. Indeed, it is possible, given
463 that one can find a domain where the signal representation
464 is sparse, to undersample the signal, i.e. not work at the
465 Nyquist frequency. By approximating the signal shape it is
466 possible to retain the relevant information and send it instead of the whole ADC reading, or even
467 work directly with signals compressed in the analog domain as in Analog to Information Converters
468 (AIC) [109–117]. In particular, this is possible only when dealing with signals, which waveform can be
469 efficiently approximated resorting to a change of basis and/or by working on the transformed signal
470 (Discrete Fourier Transform, Discrete Sine and Cosine Transforms, Wavelet Transforms and others). If
471 in one domain the signal, or a faithful representation of it, can be compactly encoded, it is possible
472 to send only its representation. For example, a sinusoidal signal in the time domain requires many
473 samples to be represented correctly while in the frequency domain it is fully characterized by a single
474 complex number representation. Even if periodic signals are easy to compress, it is also possible to
475 resort to dictionaries shared by the node and the cloud as proposed in [118]. Indeed, in [118] authors
476 cleverly set up a dictionary with redundancies to form a custom basis set for the signal representation,
477 which adapts its content with respect to the signal shape.

478 If there is no need to reconstruct the signal but only to extract some features, a different approach can
479 be exploited in order to isolate and send features. As an example in certain bio-inspired applications, it
480 is important to detect only some changes in the amplitude of the signal. This detection can be obtained
481 by a thresholding mechanism, so that when the signal crosses the threshold, the system generates a
482 pulse or an event. In systems where only the event is important, it would be a waste of resources
483 to send the whole signal. In [119] an event-based bio-inspired pseudo-neuromorphic approach is
484 proposed. This system aims to mimic the behavior of neurons, by sending a pulse only when the
485 signal crosses a defined threshold. [120–123] shows how this approach can be effectively coupled with
486 an impulse radio UWB communication system. Indeed, in such a system pulses behave as a spread
487 spectrum signal that can be effectively sent with a very low power consumption and without further
488 processing. This approach works well with low frequency signals, which information can be predicted
489 and where an event is an anomalous deviation from the expected behavior, which is conceptually
490 similar to an asynchronous delta-sigma modulation.

491 In [124] different types of compressive sensing and reconstruction algorithms are listed, along
492 with a list of successful compressive sensing applications in the fields of imaging, biomedical,
493 communications, pattern recognition, audio and video processing, dimensionality reduction, and
494 very-large-scale integrated (VLSI) systems.

495 *3.8. From Single Core to Multi Core*

496 Energy per instruction combined with increasing frequency are pushing designers toward the
497 multi-core domain. Even though in many applications single core MCUs would be enough, the
498 IoT trend is demanding more and more computational power. By splitting the workload on more
499 low-frequency simple cores, it is possible to increase the parallel computation maintaining a low power
500 budget. Since multi-core solutions are more complex than traditional MCUs, they introduce several
501 new problems regarding data and connection management. Cache coherency is fundamental in such
502 systems: when a core produces a new write operation, all the caches that contain this variable must be
503 invalidated. Such coherency is maintained by a software locking mechanism that synchronizes data,
504 preventing the invalidation of caches. In fact, every time a cache must be trashed, a lot of energy is
505 wasted in order to flush the pipe of the core, restore cache and recompute previous instructions.

506 Data transfer between cores is power demanding and requires a very complex connection infrastructure.
507 A complex architecture, such as the multi-core one, translates to expensive silicon and power resources;
508 however, many applications cannot be supported nowadays by a normal MCU. That is the reason why

509 the trend is moving from single core to multi-core, still maintaining low power budget by limiting the
510 memory hierarchy.

511 Common solutions present heterogeneous cores, in which one core is in charge of the heavy
512 computation, while another core, usually smaller than the previous one, handles the peripheral
513 requests. Since the two cores handle different tasks, they are decoupled, allowing for better energy
514 management since each of them can be turned off completely.

Figure 9 shows the trade-off between single core and multi-core architectures from a

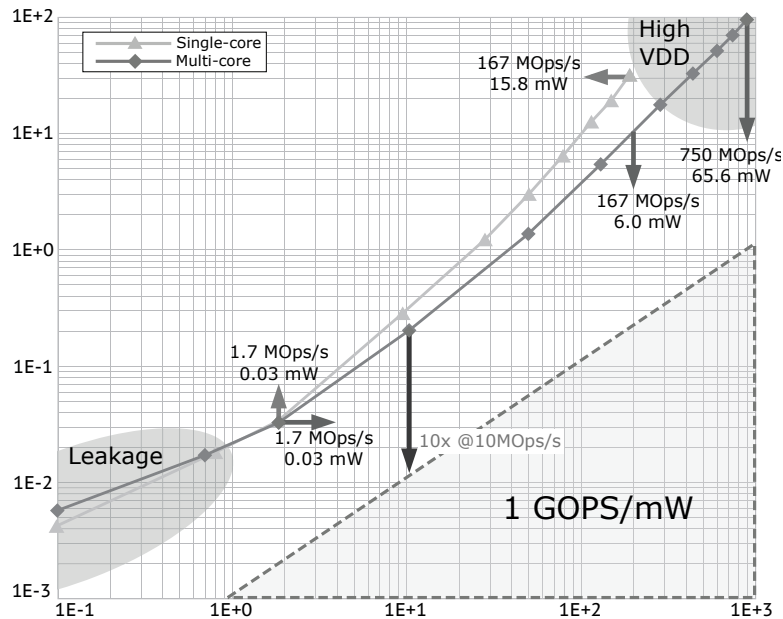


Figure 9. Comparison between single core and multi-core architectures

515 power-per-operation perspective. For high workload, a multi-core approach is more efficient. On the
516 other hand, in near-threshold or deep-sleep cases, power leakage is dominant and multi-core, being
517 composed roughly by twice the number of transistors (in case of dual-core), it is less energy efficient.
518 However, a multi-core approach is convenient to reduce the active period by a factor equal to the
519 number of cores, when working with a duty cycling behavior in which the MCU is active for a certain
520 period and in deep-sleep for the rest of the time. In this way, dynamic power is reduced by increasing
521 the deep-sleep period.
522

523 Introducing a more efficient power management system it is possible to reduce both dynamic and
524 static power by handling each core separately. Dynamic Voltage and Frequency Scaling (DVFS) can
525 be performed on each core, enhancing energy savings and decoupling their working points. Being
526 able to operate each core independently, as well as shutting down them selectively, represents the best
527 low power configuration. However, this needs a hardware overhead and an increased complexity to
528 handle data that cross different frequency and/or voltage domains, requiring handshaking operations.
529 Such cost is not always affordable so, sometimes, simpler solutions could be adopted like power gating.
530 Cores would be subjected to the same voltage and frequency domain, but still capable to be switched
531 off independently. This solution requires a ring of p-type MOS transistor all around the core, and being
532 p-type MOS transistors larger than n-type MOS ones, the area overhead is not negligible. Furthermore,
533 this ring introduces some performance degradation, affecting pull up resistance and consequently the
534 current drain. However, power leakage is strongly reduced, still considering that the core requires a
535 non-negligible time to be restored.

536 3.9. Memory Hierarchy for Multi-Core Domain

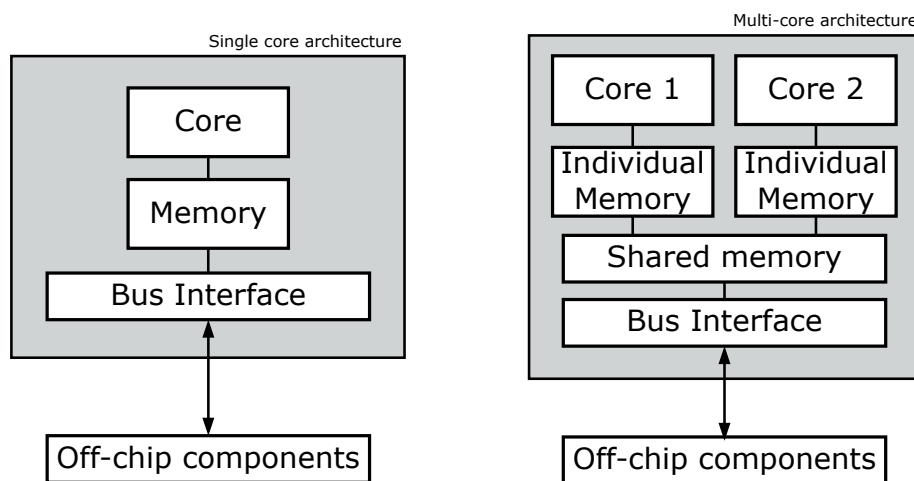
537 Multi-core systems introduced a new problem: parallel memory hierarchy and management.
 538 Since CPU computational power started increasing, memory always represented a bottleneck for both
 539 speed and energy consumption. Indeed, CPUs can process data at a higher rate than memories. As
 540 aforementioned, multi-core domain requires smart memory management in order to avoid cache
 541 invalidation. A schematic view of a typical multi-core system is shown in Figure 10. This type of
 542 coherency represents a limit to the number of cores.

543 IoT is a heterogeneous environment, so it is not possible to refer to a specific architecture. The device
 544 architecture is based on the type of application and data dependency. So the memory hierarchy must
 545 be designed taking into account the final application and dataflow.

546 The common multi-core memory hierarchy is composed of two levels, the first one is private and
 547 is placed inside the single core, while the second layer is common to all the cores. The first one is
 548 smaller and is very energy efficiency, leading to a lower hit latency, but it presents coherency problems.
 549 Moreover, when the cores are working on the same dataset, there is the need for data replication that
 550 translates to energy wasting. The second level is bigger and stores data in a fixed position in order to
 551 maintain integrity. Due to this characteristic, the hitting process is slower, but no data replication is
 552 required.

553 In order to remove coherency problems due to local storage, different types of L2 caches have been
 554 developed. Loi et al. [125] present a model of L2 cache shared among different cores that behaves as a
 555 final memory stage. The final implementation of the innovative architecture on a 28 nm technology
 556 showed a very high bandwidth and low power consumption.

Researchers conducted many studies on memory hierarchy performance for multi-core platforms,



557 **Figure 10.** Memory management of single-core and multi-core systems

558 presenting some innovative solutions. Farimah R. Poursafae et al. [126] proposed a different approach
 559 in which Non-Volatile Memories (NVM) are used instead of conventional static or dynamic RAM for
 560 producing a low power hierarchy storage. They exploited the NVM property of consuming less static
 561 power while presenting a higher density. Based on the fact that NVM presents a limited lifetime, they
 562 proposed a memory management-aware method able to allocate data based on the access patterns
 563 defined at compilation time. This approach reduces power consumption but presents low adaptability
 564 to the final application and a lower throughput with respect to RAM technology.

565 Johannes Ax et al. [127] compared different architectures in their work, presenting a new tightly
 566 coupled shared data storage with respect to each core cluster. The result shows an improvement of
 567 roughly 20% respect to other solutions, tested on 10 different applications.

568

569 4. Example of a Many-Core Low-Power processor: PULP

570 In this section, a practical ultra-low power many-core architecture is presented as an example
 571 of all the topics discussed above. The device under examination is the so-called Parallel-processing
 572 Ultra-Low Power Platform (PULP)[128] developed by the Department of Electrical, Electronic and
 573 Information Engineering of University of Bologna and the Integrated System Laboratory of ETH
 574 Zurich. Figure 11 shows the internal structure of the system, including peripherals, several bus types,
 575 memory hierarchy and computational logic. The structure is defined as a System on Chip (SoC)
 576 composed by clusters of cores featuring a lot of different functionalities. Each core represents a single
 577 Processing Element (PE) of the cluster.

In order to avoid cache coherency problems, no local data storage is included in the single PE, instead,

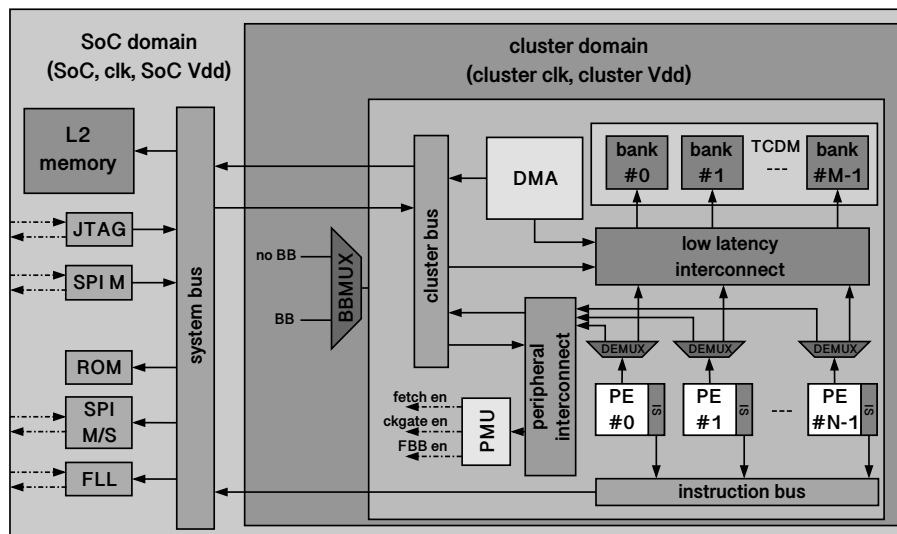


Figure 11. PULP SoC architecture

578 a tightly coupled data memory (TCDM) is included in the cluster and connected to all cores by means
 579 of multiple parallel ports. The TCDM is split in banks, providing concurrent access through as many
 580 ports as the number of banks. A DMA allows to move data from L2 memory to TCDM with minimal
 581 energy cost and in the meantime, it is on charge of bridging with other clusters or peripherals.
 582 Outside of cluster domains, an L2 memory provides processing data to clusters and peripherals
 583 allowing the SoC to interface with the rest of the world. As exposed by Conti et al. in [128], PULP
 584 can operate in two different modes: stand-alone or slave. In the former mode, a flash memory can
 585 be connected on the SPI interface allowing the SoC to draw data from it, or from the L2 storage
 586 otherwise. In the second mode, instead, PULP is seen as an accelerator that must be coupled with an
 587 external processor that is on charge of loading data in the L2 by means of the SPI and synchronize the
 588 elaboration by means of dedicated signals directly mapped in the memory. As highlighted in figure 11,
 589 in order to improve the power management, SoC domain and cluster domain are exposed to different
 590 clock and voltage signals. Each cluster and the SoC itself are equipped with clock dividers, so it is
 591 possible to fine tuning the frequency of each different part. Moreover, each core can be clock-gated
 592 to further reduce the power consumption. This allows to better allocate hardware resources to different
 593 workloads.
 594 Since different workloads can require different computational power, a multiplexer denominated
 595 BBMUX allows to choose the back bias for each cluster, thus implementing a dynamic body bias.
 596 Transitioning from a back bias value to another is possible to pass from a normal mode to a boost
 597 one, improving the speed elaboration capability. In order to manage those transitions and make them
 598 transparent to the final user, a Power Management Unit (PMU) is introduced. This is in charge of
 599 producing all the control signals needed to control the clock gating and the BBMUX selection.
 600

601 PULP is composed of cores based on OpenRISC and RISC-V ISA with a parallelism that can vary
602 from 32 to 64 bits. Being synthesized exploiting a 28 nm UTBB FD-SOI technology provided by
603 STMicroelectronics, PULP can reach 211 GOPS/W that makes him one of the most interesting open
604 source ultra-low power core suitable for several tasks, included convolutional networks and image
605 processing.

606 5. Conclusion and future perspective

607 This paper proposes an overview of the main techniques to design hardware platforms able to
608 cope with IoT requirements, by exploiting the Edge Computing paradigm. As it can be observed, many
609 approaches have a clear focus on lowering the power consumption. This is, by now, a basic requirement,
610 since the increasing computational effort affects battery lifetime of many mobile devices. From section
611 to section, all the fundamental components of an IoT electronic system have been analyzed. Despite
612 the general architecture is common to different devices, the input/output part, namely peripherals, is
613 the most variable part. In fact, sensors depend on the type of applications and sometimes the design
614 can be very complex.

615 The main point addressed through the different sections is that power consumption comes from three
616 main sources: sensor sampling, read and store from and to the memory, transmitting and receiving
617 information. The former is strictly related to the frequency at which data is acquired, and since the
618 power consumption directly depends on the frequency, the faster the system, the larger the energy
619 required. As for memory usage, this represents a huge bottleneck from the power consumption point
620 of view, especially in systems where many accesses are required. Finally, the transmission or the
621 reception of data is a power-hungry task as well, since the intervention of the antenna comes at a
622 non-negligible cost.

623 The Internet of Things era is rapidly evolving toward a scenario where everything is connected
624 and this requires the adoption of new design rules to adapt available technologies to the new
625 challenges. The actual trend suggests that in the next future IoT will become more and more pervasive,
626 incorporating all the devices around us. However, since it will be impossible to upgrade so many
627 devices in a second moment of their lifetime, one of IoT paradigm requests is to develop now solutions
628 able to withstand these issues. This is a difficult task also from an economic standpoint, as the processes
629 involved to produce such systems could require to sell millions of units in order to cut down the cost
630 for the final user and to make the IoT truly pervasive and accessible. Moreover, Big Data experts must
631 be ready to handle an enormous amount of information never seen before, making Data Mining a
632 fundamental tool.

633 From a hardware point of view, new devices must be able to accommodate the 5G technology, that will
634 invest the entire technological community, bringing the communication to a totally different level from
635 today, thus further widening the range of IoT applications.

636 Hardware must also be aware of another important and very pervasive tool, strictly related to the
637 above mentioned Data Mining, that is Machine Learning. Models such as computer vision, speech
638 recognition and many others are becoming more and more popular, and it is just a matter of time
639 before they will change our everyday life. However, these kind of algorithms require quite high
640 computing power and, consequently, a substantial amount of energy. This aspect is strictly related
641 to two interesting research fields, on one hand algorithms must be optimized to meet the hardware
642 and energy constraints dictated by the IoT platforms. On the other hand, researchers need to find new
643 solutions to cope with power requirements of modern devices by working on new battery technologies
644 and more effective energy harvesters.

645 To conclude, the IoT market is very dynamic and constantly evolving, resulting in an extremely
646 appealing field both for manufacturers and inventors. Since it gathers very open-minded people,
647 it represents a good opportunity for emerging technologies to show off their qualities, especially
648 regarding the hardware world.

649 Even though this paper presents many of the main aspects characterizing the IoT world, many
650 other points are untouched. In fact, if on one hand all the key points related to the hardware design are
651 analyzed in detail, on the other hand, the software-based literature is still to be investigated in depth.
652 This represents an interesting opportunity for future work.

653

- 654 1. Gartner website. <http://www.gartner.com>.
- 655 2. Routh, K.; Pal, T. A survey on technological, business and societal aspects of Internet of Things by Q3,
656 2017. 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU),
657 2018, pp. 1–4. doi:10.1109/IoT-SIU.2018.8519898.
- 658 3. Ture, K.; Devos, A.; Maloberti, F.; Dehollain, C. Area and Power Efficient Ultra-Wideband Transmitter
659 Based on Active Inductor. *IEEE Transactions on Circuits and Systems II: Express Briefs* **2018**, *65*, 1325–1329.
660 doi:10.1109/TCSII.2018.2853190.
- 661 4. Saputra, N.; Long, J.R. A Fully-Integrated, Short-Range, Low Data Rate FM-UWB Transmitter in 90 nm
662 CMOS. *IEEE Journal of Solid-State Circuits* **2011**, *46*, 1627–1635. doi:10.1109/JSSC.2011.2144050.
- 663 5. Mercier, P.P.; Daly, D.C.; Chandrakasan, A.P. An Energy-Efficient All-Digital UWB Transmitter Employing
664 Dual Capacitively-Coupled Pulse-Shaping Drivers. *IEEE Journal of Solid-State Circuits* **2009**, *44*, 1679–1688.
665 doi:10.1109/JSSC.2009.2020466.
- 666 6. Zhang, Z.; Li, Y.; Wang, G.; Lian, Y. The Design of an Energy-Efficient IR-UWB Transmitter With
667 Wide-Output Swing and Sub-Microwatt Leakage Current. *IEEE Transactions on Circuits and Systems
668 II: Express Briefs* **2018**, *65*, 1485–1489. doi:10.1109/TCSII.2018.2861833.
- 669 7. Zhao, M.J.; Li, B.; Wu, Z.H. 20-pJ/Pulse 250 Mbps Low-Complexity CMOS UWB Transmitter
670 for 3–5 GHz Applications. *IEEE Microwave and Wireless Components Letters* **2013**, *23*, 158–160.
671 doi:10.1109/LMWC.2013.2245412.
- 672 8. Xu, R.; Jin, Y.; Nguyen, C. Power-efficient switching-based CMOS UWB transmitters for UWB
673 communications and Radar systems. *IEEE Transactions on Microwave Theory and Techniques* **2006**,
674 *54*, 3271–3277. doi:10.1109/TMTT.2006.877830.
- 675 9. Chen, L.; Thombre, S.; Järvinen, K.; Lohan, E.S.; Alén-Savikko, A.; Leppäkoski, H.; Bhuiyan, M.Z.H.;
676 Bu-Pasha, S.; Ferrara, G.N.; Honkala, S.; Lindqvist, J.; Ruotsalainen, L.; Korpisaari, P.; Kuusniemi, H.
677 Robustness, Security and Privacy in Location-Based Services for Future IoT: A Survey. *IEEE Access* **2017**,
678 *5*, 8956–8977. doi:10.1109/ACCESS.2017.2695525.
- 679 10. Wang, S.; Zhao, Y.; Xu, J.; Yuan, J.; Hsu, C.H. Edge server placement in mobile edge computing. *Journal of
680 Parallel and Distributed Computing* **2019**, *127*, 160 – 168. doi:https://doi.org/10.1016/j.jpdc.2018.06.008.
- 681 11. Anawar, M.R.; Wang, S.; Zia, M.A.; Jadoon, A.K.; Akram, U.; Raza, S. Fog Computing: An
682 Overview of Big IoT Data Analytics. *Wireless Communications and Mobile Computing* **2018**, *2018*, 1–22.
683 doi:10.1155/2018/7157192.
- 684 12. Liu, J.; Wang, S.; Zhou, A.; Yang, F.; Buyya, R. Availability-aware Virtual Cluster Allocation
685 in Bandwidth-Constrained Datacenters. *IEEE Transactions on Services Computing* **2018**, pp. 1–1.
686 doi:10.1109/TSC.2017.2694838.
- 687 13. Lau, F.; Kuziemsky, C. Handbook of eHealth Evaluation: An Evidence-based Approach; Published by
688 University of Victoria: Victoria, British Columbia Canada v8p 5c2, 2017; chapter 9.
- 689 14. Finfgeld-Connett, D.; Johnson, E.D. Literature search strategies for conducting knowledge-building
690 and theory-generating qualitative systematic reviews. *Journal of Advanced Nursing*,
691 *69*, 194–204, [<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2648.2012.06037.x>].
692 doi:10.1111/j.1365-2648.2012.06037.x.
- 693 15. Levy, Y.; Ellis, T.J. A Systems Approach to Conduct an Effective Literature Review in Support of Information
694 Systems Research. *Informing Science: The International Journal of an Emerging Transdiscipline* **2006**, *9*, 181–212.
695 doi:10.28945/479.
- 696 16. Keutzer, K.; Malik, S.; Newton, A.R. From ASIC to ASIP: the next design discontinuity. Proceedings.
697 IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2002, pp. 84–90.
698 doi:10.1109/ICCD.2002.1106752.
- 699 17. Bluetooth communication system. <https://www.bluetooth.com>.

- 700 18. Wi-Fi Alliance. <https://www.wi-fi.org>.
- 701 19. Zigbee Alliance. <https://www.zigbee.org>.
- 702 20. Z-Wave communication system. <https://www.z-wave.com>.
- 703 21. LoRa Alliance. <https://lora-alliance.org>.
- 704 22. Fernandes, J.R.; Wentzloff, D. Recent advances in IR-UWB transceivers: An overview. Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010, pp. 3284–3287. doi:10.1109/ISCAS.2010.5537916.
- 705
- 706
- 707 23. Roch, M.R.; Martina, M. Integrated Light Sensing and Communication for LED Lighting **2018**, 2, 49. Exported from <https://app.dimensions.ai> on 2019/01/17, doi:10.3390/designs2040049.
- 708
- 709 24. Haas, H. LiFi is a paradigm-shifting 5G technology. *Reviews in Physics* **2018**, 3, 26 – 31. doi:<https://doi.org/10.1016/j.revip.2017.10.001>.
- 710
- 711 25. Nanda, U.; Pattnaik, S.K. Universal Asynchronous Receiver and Transmitter (UART). 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), 2016, Vol. 01, pp. 1–5. doi:10.1109/ICACCS.2016.7586376.
- 712
- 713
- 714 26. Serial Peripheral Interface specifications. https://www.nxp.com/files-static/microcontrollers/doc/ref_manual/S12SPIV3.pdf.
- 715
- 716 27. Inter-Integrated Circuit bus specifications. <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>.
- 717 28. MIPI Alliance specifications overview. <https://www.mipi.org/specifications>.
- 718 29. Hsieh, S.; Hsieh, C. A 0.44fJ/conversion-step 11b 600KS/s SAR ADC with semi-resting DAC. 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits), 2016, pp. 1–2. doi:10.1109/VLSIC.2016.7573519.
- 719
- 720 30. Wang, A.; Wang, Y.; Dhawan, V.; Shi, R. Design of a 10 bit 50MS/s SAR ADC in 14nm SOI FinFET CMOS. 2015 International SoC Design Conference (ISOCC), 2015, pp. 85–86. doi:10.1109/ISOCC.2015.7401645.
- 721
- 722 31. Hsieh, S.; Hsieh, C. A 0.3V 0.705fJ/conversion-step 10-bit SAR ADC with shifted monotonic switching scheme in 90nm CMOS. 2016 IEEE International Symposium on Circuits and Systems (ISCAS), 2016, pp. 2899–2899. doi:10.1109/ISCAS.2016.7539202.
- 723
- 724
- 725 32. Kung, C.; Huang, C.; Li, C.; Chang, S. A Low Energy Consumption 10-Bit 100kS/s SAR ADC with Timing Control Adaptive Window. 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1–4. doi:10.1109/ISCAS.2018.8350989.
- 726
- 727
- 728 33. Lin, J.; Hsieh, C. A 0.3 V 10-bit SAR ADC With First 2-bit Guess in 90-nm CMOS. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2017**, 64, 562–572. doi:10.1109/TCSI.2016.2613505.
- 729
- 730 34. Hsieh, S.; Hsieh, C. A 0.44-fJ/Conversion-Step 11-Bit 600-kS/s SAR ADC With Semi-Resting DAC. *IEEE Journal of Solid-State Circuits* **2018**, 53, 2595–2603. doi:10.1109/JSSC.2018.2847306.
- 731
- 732 35. Alioto, M., IoT: Bird’s Eye View, Megatrends and Perspectives. In *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*; Alioto, M., Ed.; Springer International Publishing: Cham, 2017; pp. 1–45. doi:10.1007/978-3-319-51482-6_1.
- 733
- 734
- 735 36. Xu, J.; Ma, J.; Zhang, D.; Zhang, Y.; Lin, S. Compressive video sensing based on user attention model. 28th Picture Coding Symposium, 2010, pp. 90–93. doi:10.1109/PCS.2010.5702586.
- 736
- 737 37. Wang, Z.; Zhu, J. Single-pixel compressive imaging based on motion compensation. *IET Image Processing* **2018**, 12, 2283–2291. doi:10.1049/iet-ipr.2018.5741.
- 738
- 739 38. Dias, U.V.; Patil, S.A. Compressive sensing based microarray image acquisition. International Conference for Convergence for Technology-2014, 2014, pp. 1–5. doi:10.1109/I2CT.2014.7092273.
- 740
- 741 39. Zhang, M.; Bermak, A. Compressive Acquisition CMOS Image Sensor: From the Algorithm to Hardware Implementation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **2010**, 18, 490–500. doi:10.1109/TVLSI.2008.2011489.
- 742
- 743
- 744 40. Lima, J.A.; Miosso, C.J.; Farias, M.C.Q. Per-Pixel Mirror-Based Acquisition Method for video compressive sensing. 2014 22nd European Signal Processing Conference (EUSIPCO), 2014, pp. 1058–1062.
- 745
- 746 41. Zhu, H.; Zhang, M.; Suo, Y.; Tran, T.D.; der Spiegel, J.V. Design of a Digital Address-Event Triggered Compressive Acquisition Image Sensor. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2016**, 63, 191–199. doi:10.1109/TCSI.2015.2512719.
- 747
- 748
- 749 42. Ejaz, W.; Anpalagan, A.; Imran, M.; Jo, M.; Muhammad, N.; Qaisar, S.; Wang, W. Internet of Things (IoT) in 5G Wireless Communications. *IEEE Access* **2016**, 4, 10310–10314. doi:10.1109/ACCESS.2016.2646120.
- 750
- 751 43. Al Ridhawi, I.; Aloqaily, M.; Kotb, Y.; Al Ridhawi, Y.; Jararweh, Y. A collaborative mobile edge computing and user solution for service composition in 5G systems. *Transactions on Emerging Telecommunications*
- 752

- 753 *Technologies* **2018**, *29*, e3446, [<https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.3446>]. e3446
754 ETT-18-0080.R1, doi:10.1002/ett.3446.
- 755 44. Hezam, A.; Konstantas, D.; Mahyoub, M. A Comprehensive IoT Attacks Survey based on a
756 Building-blocked Reference Mode. *International Journal of Advanced Computer Science and Applications*
757 **2018**, *Vol. 9*. doi:10.14569/IJACSA.2018.090349.
- 758 45. Yaseen, Q.; AlBalas, F.; Jararweh, Y.; Al-Ayyoub, M. A Fog Computing Based System for Selective
759 Forwarding Detection in Mobile Wireless Sensor Networks. 2016 IEEE 1st International Workshops on
760 Foundations and Applications of Self* Systems (FAS*W), 2016, pp. 256–262. doi:10.1109/FAS-W.2016.60.
- 761 46. Otoum, S.; Kantarci, B.; Mouftah, H. Adaptively Supervised and Intrusion-Aware Data Aggregation for
762 Wireless Sensor Clusters in Critical Infrastructures. 2018 IEEE International Conference on Communications
763 (ICC), 2018, pp. 1–6. doi:10.1109/ICC.2018.8422401.
- 764 47. Otoum, S.; Kantarci, B.; Mouftah, H.T. Detection of Known and Unknown Intrusive Sensor Behavior in
765 Critical Applications. *IEEE Sensors Letters* **2017**, *1*, 1–4. doi:10.1109/LESENS.2017.2752719.
- 766 48. Ghafir, I.; Saleem, J.; Hammoudeh, M.; Faour, H.; Prenosil, V.; Jaf, S.; Jabbar, S.; Baker, T. Security
767 threats to critical infrastructure: the human factor. *The Journal of Supercomputing* **2018**, *74*, 4986–5002.
768 doi:10.1007/s11227-018-2337-2.
- 769 49. Otoum, S.; Kantarci, B.; Mouftah, H.T. Mitigating False Negative intruder decisions in WSN-based Smart
770 Grid monitoring. 2017 13th International Wireless Communications and Mobile Computing Conference
771 (IWCMC), 2017, pp. 153–158. doi:10.1109/IWCMC.2017.7986278.
- 772 50. Yaseen, Q.; Jararweh, Y.; Al-Ayyoub, M.; AlDwairi, M. Collusion attacks in Internet of Things: Detection
773 and mitigation using a fog based model. 2017 IEEE Sensors Applications Symposium (SAS), 2017, pp. 1–5.
774 doi:10.1109/SAS.2017.7894031.
- 775 51. Aloqaily, M.; Kantarci, B.; Mouftah, H.T. Trusted Third Party for service management in vehicular clouds.
776 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017, pp.
777 928–933. doi:10.1109/IWCMC.2017.7986410.
- 778 52. Aloqaily, M.; Kantarci, B.; Mouftah, H.T. Fairness-Aware Game Theoretic Approach for Service
779 Management in Vehicular Clouds. 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), 2017, pp.
780 1–5. doi:10.1109/VTCFall.2017.8288282.
- 781 53. Alkheir, A.A.; Aloqaily, M.; Mouftah, H.T. Connected and Autonomous Electric Vehicles (CAEVs). *IT*
782 *Professional* **2018**, *20*, 54–61. doi:10.1109/MITP.2018.2876977.
- 783 54. Ridhawi, I.A.; Aloqaily, M.; Kantarci, B.; Jararweh, Y.; Mouftah, H.T. A continuous diversified
784 vehicular cloud service availability framework for smart cities. *Computer Networks* **2018**, *145*, 207 –
785 218. doi:https://doi.org/10.1016/j.comnet.2018.08.023.
- 786 55. Otoum, S.; Kantarci, B.; Mouftah, H.T. Hierarchical trust-based black-hole detection in WSN-based
787 smart grid monitoring. 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–6.
788 doi:10.1109/ICC.2017.7997099.
- 789 56. Khan, S.; Paul, D.; Momtahan, P.; Aloqaily, M. Artificial intelligence framework for smart city microgrids:
790 State of the art, challenges, and opportunities. 2018 Third International Conference on Fog and Mobile
791 Edge Computing (FMED), 2018, pp. 283–288. doi:10.1109/FMED.2018.8364080.
- 792 57. Vianello, E.; Thomas, O.; Molas, G.; Turkyilmaz, O.; Jovanović, N.; Garbin, D.; Palma, G.; Alayan, M.;
793 Nguyen, C.; Coignus, J.; Giraud, B.; Benoist, T.; Reyboz, M.; Toffoli, A.; Charpin, C.; Clermidy, F.; Perniola,
794 L. Resistive Memories for Ultra-Low-Power embedded computing design. 2014 IEEE International
795 Electron Devices Meeting, 2014, pp. 6.3.1–6.3.4. doi:10.1109/IEDM.2014.7046995.
- 796 58. DeSalvo, B.; Vianello, E.; Thomas, O.; Clermidy, F.; Bichler, O.; Gamrat, C.; Perniola, L. Emerging resistive
797 memories for low power embedded applications and neuromorphic systems. 2015 IEEE International
798 Symposium on Circuits and Systems (ISCAS), 2015, pp. 3088–3091. doi:10.1109/ISCAS.2015.7169340.
- 799 59. Hemavathy, B.; Meenakshi, V. A novel design for low power Re-RAM based non-volatile flip flop using
800 content addressable memory. 2017 Third International Conference on Science Technology Engineering
801 Management (ICONSTEM), 2017, pp. 879–883. doi:10.1109/ICONSTEM.2017.8261328.
- 802 60. Kazi, I.; Meinerzhagen, P.; Gaillardon, P.; Sacchetto, D.; Leblebici, Y.; Burg, A.; Micheli, G.D.
803 Energy/Reliability Trade-Offs in Low-Voltage ReRAM-Based Non-Volatile Flip-Flop Design. *IEEE*
804 *Transactions on Circuits and Systems I: Regular Papers* **2014**, *61*, 3155–3164. doi:10.1109/TCSI.2014.2334891.

- 805 61. Biglari, M.; Lieske, T.; Fey, D. High-Endurance Bipolar ReRAM-Based Non-Volatile Flip-Flops with
806 Run-Time Tunable Resistive States. 2018 IEEE/ACM International Symposium on Nanoscale Architectures
807 (NANOARCH), 2018, pp. 1–6.
- 808 62. Lee, A.; Lo, C.; Lin, C.; Chen, W.; Hsu, K.; Wang, Z.; Su, F.; Yuan, Z.; Wei, Q.; King, Y.; Lin, C.; Lee, H.;
809 Amiri, P.K.; Wang, K.; Wang, Y.; Yang, H.; Liu, Y.; Chang, M. A ReRAM-Based Nonvolatile Flip-Flop With
810 Self-Write-Termination Scheme for Frequent-OFF Fast-Wake-Up Nonvolatile Processors. *IEEE Journal of*
811 *Solid-State Circuits* **2017**, *52*, 2194–2207. doi:10.1109/JSSC.2017.2700788.
- 812 63. Chien, T.; Chiou, L.; Chuang, Y.; Sheu, S.; Li, H.; Wang, P.; Ku, T.; Tsai, M.; Wu, C. A low store energy and
813 robust ReRAM-based flip-flop for normally off microprocessors. 2016 IEEE International Symposium on
814 Circuits and Systems (ISCAS), 2016, pp. 2803–2806. doi:10.1109/ISCAS.2016.7539175.
- 815 64. Koga, M.; Iida, M.; Amagasaki, M.; Ichida, Y.; Saji, M.; Iida, J.; Sueyoshi, T. A power-gatable reconfigurable
816 logic chip with FeRAM cells. TENCON 2010 - 2010 IEEE Region 10 Conference, 2010, pp. 317–322.
817 doi:10.1109/TENCON.2010.5686019.
- 818 65. Qazi, M.; Amerasekera, A.; Chandrakasan, A.P. A 3.4-pJ FeRAM-Enabled D Flip-Flop in 0.13- μ mCMOS
819 for Nonvolatile Processing in Digital Systems. *IEEE Journal of Solid-State Circuits* **2014**, *49*, 202–211.
820 doi:10.1109/JSSC.2013.2282112.
- 821 66. Masui, S.; Yokozeki, W.; Oura, M.; Ninomiya, T.; Mukaida, K.; Takayama, Y.; Teramoto, T. Design and
822 applications of ferroelectric nonvolatile SRAM and flip-flop with unlimited read/program cycles and
823 stable recall. Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, 2003., 2003, pp. 403–406.
824 doi:10.1109/CICC.2003.1249428.
- 825 67. Lee, H.; Jung, S.; Song, Y.H. PCRAM-assisted ECC management for enhanced data reliability in flash storage
826 systems. *IEEE Transactions on Consumer Electronics* **2012**, *58*, 849–856. doi:10.1109/TCE.2012.6311327.
- 827 68. Raoux, S.; Burr, G.W.; Breitwisch, M.J.; Rettner, C.T.; Chen, Y.; Shelby, R.M.; Salinga, M.; Krebs, D.; Chen, S.;
828 Lung, H.; Lam, C.H. Phase-change random access memory: A scalable technology. *IBM Journal of Research*
829 *and Development* **2008**, *52*, 465–479. doi:10.1147/rd.524.0465.
- 830 69. Lung, H.L.; Ho, Y.H.; Zhu, Y.; Chien, W.C.; Kim, S.; Kim, W.; Cheng, H.Y.; Ray, A.; Brightsky, M.; Bruce, R.;
831 Yeh, C.W.; Lam, C. A novel low power phase change memory using inter-granular switching. 2016 IEEE
832 Symposium on VLSI Technology, 2016, pp. 1–2. doi:10.1109/VLSIT.2016.7573405.
- 833 70. Cheng, H.Y.; Chien, W.C.; BrightSky, M.; Ho, Y.H.; Zhu, Y.; Ray, A.; Bruce, R.; Kim, W.; Yeh, C.W.;
834 Lung, H.L.; Lam, C. Novel fast-switching and high-data retention phase-change memory based on new
835 Ga-Sb-Ge material. 2015 IEEE International Electron Devices Meeting (IEDM), 2015, pp. 3.5.1–3.5.4.
836 doi:10.1109/IEDM.2015.7409620.
- 837 71. Cheon, J.; Lee, I.; Ahn, C.; Stanisavljevic, M.; Athmanathan, A.; Papandreou, N.; Pozidis, H.; Eleftheriou,
838 E.; Shin, M.; Kim, T.; Kang, J.H.; Chun, J.H. Non-resistance metric based read scheme for multi-level
839 PCRAM in 25 nm technology. 2015 IEEE Custom Integrated Circuits Conference (CICC), 2015, pp. 1–4.
840 doi:10.1109/CICC.2015.7338358.
- 841 72. Nirschl, T.; Philipp, J.B.; Happ, T.D.; Burr, G.W.; Rajendran, B.; Lee, M.; Schrott, A.; Yang, M.; Breitwisch,
842 M.; Chen, C.; Joseph, E.; Lamorey, M.; Cheek, R.; Chen, S.; Zaidi, S.; Raoux, S.; Chen, Y.C.; Zhu, Y.;
843 Bergmann, R.; Lung, H.; Lam, C. Write Strategies for 2 and 4-bit Multi-Level Phase-Change Memory. 2007
844 IEEE International Electron Devices Meeting, 2007, pp. 461–464. doi:10.1109/IEDM.2007.4418973.
- 845 73. Lakys, Y.; Zhao, W.S.; Klein, J.; Chappert, C. Hardening techniques for MRAM-based non-volatile storage
846 cells and logic. 2011 12th European Conference on Radiation and Its Effects on Components and Systems,
847 2011, pp. 669–674. doi:10.1109/RADECS.2011.6131445.
- 848 74. Münch, C.; Bishnoi, R.; Tahoori, M.B. Multi-bit non-volatile spintronic flip-flop. 2018 Design, Automation
849 Test in Europe Conference Exhibition (DATE), 2018, pp. 1229–1234. doi:10.23919/DATE.2018.8342203.
- 850 75. Huang, K.; Lian, Y. A Low-Power Low-VDD Nonvolatile Latch Using Spin Transfer Torque MRAM. *IEEE*
851 *Transactions on Nanotechnology* **2013**, *12*, 1094–1103. doi:10.1109/TNANO.2013.2280338.
- 852 76. Sakimura, N.; Sugibayashi, T.; Nebashi, R.; Kasai, N. Nonvolatile Magnetic Flip-Flop
853 for Standby-Power-Free SoCs. *IEEE Journal of Solid-State Circuits* **2009**, *44*, 2244–2250.
854 doi:10.1109/JSSC.2009.2023192.
- 855 77. Pyle, S.D.; Li, H.; DeMara, R.F. Compact low-power instant store and restore D flip-flop using a
856 self-complementing spintronic device. *Electronics Letters* **2016**, *52*, 1238–1240. doi:10.1049/el.2015.4114.

- 857 78. Chabi, D.; Zhao, W.; Zhang, Y.; Klein, J.; Chappert, C. Low power magnetic flip-flop based on checkpointing
858 and self-enable mechanism. 2013 IEEE 11th International New Circuits and Systems Conference
859 (NEWCAS), 2013, pp. 1–4. doi:10.1109/NEWCAS.2013.6573616.
- 860 79. Sakimura, N.; Sugibayashi, T.; Nebashi, R.; Kasai, N. Nonvolatile Magnetic Flip-Flop for
861 standby-power-free SoCs. 2008 IEEE Custom Integrated Circuits Conference, 2008, pp. 355–358.
862 doi:10.1109/CICC.2008.4672095.
- 863 80. Gangalakshmi, S.; Banu, A.K.; Kumar, R.H. Implementing the design of magnetic flip-flop based on
864 swapped MOS design. 2015 2nd International Conference on Electronics and Communication Systems
865 (ICECS), 2015, pp. 987–989. doi:10.1109/ECS.2015.7125062.
- 866 81. Zhao, W.; Belhaire, E.; Chappert, C. Spin-MTJ based Non-volatile Flip-Flop. 2007 7th IEEE Conference on
867 Nanotechnology (IEEE NANO), 2007, pp. 399–402. doi:10.1109/NANO.2007.4601218.
- 868 82. Kimura, H.; Zhong, Z.; Mizuochi, Y.; Kinouchi, N.; Ichida, Y.; Fujimori, Y. Highly Reliable Non-Volatile
869 Logic Circuit Technology and Its Application. 2013 IEEE 43rd International Symposium on Multiple-Valued
870 Logic, 2013, pp. 212–218. doi:10.1109/ISMVL.2013.32.
- 871 83. Cai, H.; Wang, Y.; Naviner, L.; Zhao, W. Novel Pulsed-Latch Replacement in Non-Volatile Flip-Flop
872 Core. 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2017, pp. 57–61.
873 doi:10.1109/ISVLSI.2017.19.
- 874 84. Jovanović, N.; Thomas, O.; Vianello, E.; Portal, J.; Nikolić, B.; Naviner, L. OxRAM-based non volatile
875 flip-flop in 28nm FDSOI. 2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS),
876 2014, pp. 141–144. doi:10.1109/NEWCAS.2014.6934003.
- 877 85. Li, M.; Huang, P.; Shen, L.; Zhou, Z.; Kang, J.; Liu, X. Simulation of the RRAM-based flip-flops
878 with data retention. 2016 IEEE International Nanoelectronics Conference (INEC), 2016, pp. 1–2.
879 doi:10.1109/INEC.2016.7589321.
- 880 86. Kazi, I.; Meinerzhagen, P.; Gaillardon, P.; Sacchetto, D.; Burg, A.; Micheli, G.D. A ReRAM-based
881 non-volatile flip-flop with sub-VT read and CMOS voltage-compatible write. 2013 IEEE 11th International
882 New Circuits and Systems Conference (NEWCAS), 2013, pp. 1–4. doi:10.1109/NEWCAS.2013.6573586.
- 883 87. Ali, K.; Li, F.; Lua, S.Y.H.; Heng, C. Compact spin transfer torque non-volatile flip flop design for
884 power-gating architecture. 2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2016,
885 pp. 119–122. doi:10.1109/APCCAS.2016.7803911.
- 886 88. Kang, W.; Ran, Y.; Lv, W.; Zhang, Y.; Zhao, W. High-Speed, Low-Power, Magnetic Non-Volatile
887 Flip-Flop With Voltage-Controlled, Magnetic Anisotropy Assistance. *IEEE Magnetics Letters* **2016**, *7*, 1–5.
888 doi:10.1109/LMAG.2016.2604205.
- 889 89. Ueda, M.; Otsuka, T.; Toyoda, K.; Morimoto, K.; Morita, K. A novel non-volatile flip-flop using a
890 ferroelectric capacitor. Proceedings of the 13th IEEE International Symposium on Applications of
891 Ferroelectrics, 2002. ISAF 2002., 2002, pp. 155–158. doi:10.1109/ISAF.2002.1195893.
- 892 90. Park, J. Area-efficient STT/CMOS non-volatile flip-flop. 2017 IEEE International Symposium on Circuits
893 and Systems (ISCAS), 2017, pp. 1–4. doi:10.1109/ISCAS.2017.8050697.
- 894 91. Zhao, W.; Belhaire, E.; Javerliac, V.; Chappert, C.; Dieny, B. A non-volatile flip-flop in magnetic FPGA chip.
895 International Conference on Design and Test of Integrated Systems in Nanoscale Technology, 2006. DTIS
896 2006., 2006, pp. 323–326. doi:10.1109/DTIS.2006.1708702.
- 897 92. Na, T.; Ryu, K.; Kim, J.; Jung, S.; Kim, J.P.; Kang, S.H. High-performance low-power magnetic tunnel
898 junction based non-volatile flip-flop. 2014 IEEE International Symposium on Circuits and Systems (ISCAS),
899 2014, pp. 1953–1956. doi:10.1109/ISCAS.2014.6865544.
- 900 93. Izumi, S.; Kawaguchi, H.; Yoshimoto, M.; Kimura, H.; Fuchikami, T.; Marumoto, K.; Fujimori, Y. A
901 ferroelectric-based non-volatile flip-flop for wearable healthcare systems. 2015 15th Non-Volatile Memory
902 Technology Symposium (NVMTS), 2015, pp. 1–4. doi:10.1109/NVMTS.2015.7457489.
- 903 94. Jung, Y.; Kim, J.; Ryu, K.; Jung, S.; Kim, J.P.; Kang, S.H. MTJ based non-volatile flip-flop
904 in deep submicron technology. 2011 International SoC Design Conference, 2011, pp. 424–427.
905 doi:10.1109/ISOCC.2011.6138622.
- 906 95. Kimura, H.; Fuchikami, T.; Maramoto, K.; Fujimori, Y.; Izumi, S.; Kawaguchi, H.; Yoshimoto, M. A 2.4 pJ
907 ferroelectric-based non-volatile flip-flop with 10-year data retention capability. 2014 IEEE Asian Solid-State
908 Circuits Conference (A-SSCC), 2014, pp. 21–24. doi:10.1109/ASSCC.2014.7008850.

- 909 96. Portal, J.M.; Bocquet, M.; Moreau, M.; Aziza, H.; Deleruyelle, D.; Zhang, Y.; Kang, W.; Klein, J.O.; Zhang, Y.;
910 Chappert, C.; Zhao, W. An Overview of Non-Volatile Flip-Flops Based on Emerging Memory Technologies.
911 *Journal of Electronic Science and Technology* **2014**, *12*, 173 – 181. doi:10.3969/j.issn.1674-862X.2014.02.007.
- 912 97. Aiassa, S.; Ros, P.M.; Masera, G.; Martina, M. A low power architecture for AER event-processing
913 microcontroller. 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS), 2017, pp. 1–4.
914 doi:10.1109/BIOCAS.2017.8325170.
- 915 98. Georgiou, J.; Andreou, A.G. Address-data event representation for communication in multichip
916 neuromorphic system architectures. *Electronics Letters* **2007**, *43*. doi:10.1049/el:20070897.
- 917 99. Georgiou, J.; Andreou, A.G. Address Data Event Representation (ADER) for Efficient Neuromorphic
918 Communication. 2007 41st Annual Conference on Information Sciences and Systems, 2007, pp. 755–758.
919 doi:10.1109/CISS.2007.4298409.
- 920 100. Kapoor, C.; Singh, H.; Laxmi, V. A Survey on Energy Efficient Routing for Delay Minimization in IoT
921 Networks. 2018 International Conference on Intelligent Circuits and Systems (ICICS), 2018, pp. 320–323.
922 doi:10.1109/ICICS.2018.00072.
- 923 101. Rzepecki, W.; Iwanecki, L.; Ryba, P. IEEE 802.15.4 Thread Mesh Network – Data Transmission in Harsh
924 Environment. 2018 6th International Conference on Future Internet of Things and Cloud Workshops
925 (FiCloudW), 2018, pp. 42–47. doi:10.1109/W-FiCloud.2018.00013.
- 926 102. Joo, S.; Kim, H. Low power wireless mesh network over the TDMA link for connecting things. 2015
927 17th International Conference on Advanced Communication Technology (ICACT), 2015, pp. 143–146.
928 doi:10.1109/ICACT.2015.7224774.
- 929 103. Kim, J.; Lee, J. Cluster-Based Mobility Supporting WMN for IoT Networks. 2012
930 IEEE International Conference on Green Computing and Communications, 2012, pp. 700–703.
931 doi:10.1109/GreenCom.2012.114.
- 932 104. Matthys, N.; Yang, F.; Daniels, W.; Michiels, S.; Joosen, W.; Hughes, D.; Watteyne, T. uPnP-Mesh: The
933 plug-and-play mesh network for the Internet of Things. 2015 IEEE 2nd World Forum on Internet of Things
934 (WF-IoT), 2015, pp. 311–315. doi:10.1109/WF-IoT.2015.7389072.
- 935 105. Vittecoq, S. A Radio Mesh Platform for the IOT. 2013 Seventh International Conference on Innovative
936 Mobile and Internet Services in Ubiquitous Computing, 2013, pp. 530–534. doi:10.1109/IMIS.2013.94.
- 937 106. Liu, Y.; Tong, K.F.; Qiu, X.; Liu, Y.; Ding, X. Wireless Mesh Networks in IoT networks. 2017 International
938 Workshop on Electromagnetics: Applications and Student Innovation Competition, 2017, pp. 183–185.
939 doi:10.1109/iWEM.2017.7968828.
- 940 107. Paek, J. Fast and Adaptive Mesh Access Control in Low-Power and Lossy Networks. *IEEE Internet of
941 Things Journal* **2015**, *2*, 435–444. doi:10.1109/JIOT.2015.2457940.
- 942 108. Darroudi, S.M.; Gomez, C. Modeling the Connectivity of Data-Channel-Based Bluetooth Low Energy
943 Mesh Networks. *IEEE Communications Letters* **2018**, *22*, 2124–2127. doi:10.1109/LCOMM.2018.2864994.
- 944 109. Daly, E.L.; Bernhard, J.T. The rapidly tuned analog-to-information converter. 2013 Asilomar Conference
945 on Signals, Systems and Computers, 2013, pp. 495–499. doi:10.1109/ACSSC.2013.6810327.
- 946 110. Cambareri, V.; Mangia, M.; Pareschi, F.; Rovatti, R.; Setti, G. A rakeness-based design flow for
947 Analog-to-Information conversion by Compressive Sensing. 2013 IEEE International Symposium on
948 Circuits and Systems (ISCAS2013), 2013, pp. 1360–1363. doi:10.1109/ISCAS.2013.6572107.
- 949 111. Pareschi, F.; Albertini, P.; Frattini, G.; Mangia, M.; Rovatti, R.; Setti, G. Hardware-Algorithms Co-Design
950 and Implementation of an Analog-to-Information Converter for Biosignals Based on Compressed Sensing.
951 *IEEE Transactions on Biomedical Circuits and Systems* **2016**, *10*, 149–162. doi:10.1109/TBCAS.2015.2444276.
- 952 112. Liu, S.; Zhang, M.; Jiang, W.; Wang, J.; Qi, P. Theory and hardware implementation of an
953 analog-to-Information Converter based on Compressive Sensing. 2013 IEEE 10th International Conference
954 on ASIC, 2013, pp. 1–4. doi:10.1109/ASICON.2013.6812033.
- 955 113. Cambareri, V.; Mangia, M.; Rovatti, R.; Setti, G. Joint analog-to-information conversion of heterogeneous
956 biosignals. 2013 IEEE Biomedical Circuits and Systems Conference (BioCAS), 2013, pp. 158–161.
957 doi:10.1109/BioCAS.2013.6679663.
- 958 114. Mangia, M.; Rovatti, R.; Setti, G. Analog-to-information conversion of sparse and non-white signals:
959 Statistical design of sensing waveforms. 2011 IEEE International Symposium of Circuits and Systems
960 (ISCAS), 2011, pp. 2129–2132. doi:10.1109/ISCAS.2011.5938019.

- 961 115. Singh, W.; Deb, S. Energy Efficient Analog-to-Information Converter for Biopotential Acquisition Systems. 2015 IEEE International Symposium on Nanoelectronic and Information Systems, 2015, pp. 141–145. doi:10.1109/iNIS.2015.48.
- 962
963
- 964 116. Abari, O.; Chen, F.; Lim, F.; Stojanovic, V. Performance trade-offs and design limitations of analog-to-information converter front-ends. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 5309–5312. doi:10.1109/ICASSP.2012.6289119.
- 965
966
- 967 117. Bellasi, D.E.; Bettini, L.; Benkeser, C.; Burger, T.; Huang, Q.; Studer, C. VLSI Design of a Monolithic Compressive-Sensing Wideband Analog-to-Information Converter. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2013**, *3*, 552–565. doi:10.1109/JETCAS.2013.2284618.
- 968
969
- 970 118. Amarlingam, M.; Mishra, P.K.; Rajalakshmi, P.; Giluka, M.K.; Tamma, B.R. Energy efficient wireless sensor networks utilizing adaptive dictionary in compressed sensing. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018, pp. 383–388. doi:10.1109/WF-IoT.2018.8355140.
- 971
972
- 973 119. Ros, P.M.; Crepaldi, M.; Damilano, A.; Demarchi, D. Integrated bio-inspired systems: An event-driven design framework. 2014 10th International Conference on Innovations in Information Technology (IIT), 2014, pp. 48–53. doi:10.1109/INNOVATIONS.2014.6987560.
- 974
975
- 976 120. Crepaldi, M.; Dapra, D.; Bonanno, A.; Aulika, I.; Demarchi, D.; Civera, P. A Very Low-Complexity 0.3–4.4 GHz 0.004 mm²All-Digital Ultra-Wide-Band Pulsed Transmitter for Energy Detection Receivers. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2012**, *59*, 2443–2455. doi:10.1109/TCSI.2012.2188954.
- 977
978
- 979 121. Crepaldi, M.; Sanginario, A.; Ros, P.M.; Demarchi, D. Low-latency asynchronous networking for the IoT: Routing analog pulse delays using IR-UWB. 2016 14th IEEE International New Circuits and Systems Conference (NEWCAS), 2016, pp. 1–4. doi:10.1109/NEWCAS.2016.7604842.
- 980
981
- 982 122. Crepaldi, M.; Angotzi, G.N.; Maviglia, A.; Berdondini, L. A 1 Gpps asynchronous logic OOK IR-UWB transmitter based on master-slave PLL synthesis. 2017 IEEE International Symposium on Circuits and Systems (ISCAS), 2017, pp. 1–4. doi:10.1109/ISCAS.2017.8050693.
- 983
984
- 985 123. Crepaldi, M.; Angotzi, G.N.; Maviglia, A.; Diotalevi, F.; Berdondini, L. A 5 pJ/pulse at 1-Gpps Pulsed Transmitter Based on Asynchronous Logic Master–Slave PLL Synthesis. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2018**, *65*, 1096–1109. doi:10.1109/TCSI.2017.2762159.
- 986
987
- 988 124. Rani, M.; Dhok, S.B.; Deshmukh, R.B. A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications. *IEEE Access* **2018**, *6*, 4875–4894. doi:10.1109/ACCESS.2018.2793851.
- 989
990
- 991 125. Loi, I.; Benini, L. A multi banked - Multi ported - Non blocking shared L2 cache for MPSoC platforms. 2014 Design, Automation Test in Europe Conference Exhibition (DATE), 2014, pp. 1–6. doi:10.7873/DATE.2014.093.
- 992
993
- 994 126. Poursafaei, F.R.; Bazzaz, M.; Ejlali, A. NPAM: NVM-Aware Page Allocation for Multi-Core Embedded Systems. *IEEE Transactions on Computers* **2017**, *66*, 1703–1716. doi:10.1109/TC.2017.2703824.
- 995
996
- 997 127. Ax, J.; Sievers, G.; Daberkow, J.; Flaskamp, M.; Vohrmann, M.; Jungeblut, T.; Kelly, W.; Porrmann, M.; Rückert, U. CoreVA-MPSoC: A Many-Core Architecture with Tightly Coupled Shared and Local Data Memories. *IEEE Transactions on Parallel and Distributed Systems* **2018**, *29*, 1030–1043. doi:10.1109/TPDS.2017.2785799.
- 998
999
- 1000 128. Conti, F.; Rossi, D.; Pullini, A.; Loi, I.; Benini, L. PULP: A Ultra-Low Power Parallel Accelerator for Energy-Efficient and Flexible Embedded Vision. *Journal of Signal Processing Systems* **2016**, *84*, 339–354. doi:10.1007/s11265-015-1070-9.
- 1001