Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348478

REVIEW OF LOW POWER TECHNIQUES FOR INTERNET OF THINGS IMPLEMENTATION

SONAL ARVIND BARGE
School of Electronics Engineering, Vellore Institute of Technology, Tamilnadu, India.
Email: sonaldigvijaysingh.2018@vitstudent.ac.in

GERARDINE IMMACULATE MARY
School of Electronics Engineering, Vellore Institute of Technology, Tamilnadu, India.
Email: gerardine@vit.ac.in

Abstract

A low-power system is created with a long battery life in mind. One can carefully use low-power design strategies to reduce power consumption while maintaining correct system operation in order to lengthen battery life. The Internet of Things (IoT) is one such system where a lot of gadgets are often battery-operated and connected to the Internet. Devices connected to the Internet of Things are frequently used to collect data, drive a series of operations, and communicate the data for automation, connectivity, and data analysis. IoT communication and computing in IoT devices are the subject of so much study, which increases power consumption in those devices. This essay investigates low-power design strategies used at the architectural level. Highlights IoT elements that allow for the application of low-power design methodologies to achieve low-power consumption in IoT.

Keywords: Low Power, Internet of Things, Clock gating, Power gating, DVFS, Reconfigurable architecture

1. INTRODUCTION

Batteries are typically used to power IoT devices. Finite battery capacity is a major restriction on battery-powered devices because when communication in Internet of Things devices happens, a significant amount of energy is consumed, which results in the device operating for a short time until the battery runs out. Changing batteries can be a good option for IoT systems that are tiny, but it can be challenging to maintain and replace many batteries in systems that are large. For large IoT systems, extending battery life can be a good solution. Utilizing low-power design techniques is a viable answer to this issue.

Existing low-power design methodologies are used to build embedded system models at the RTL level or low level. Research is required to develop more application strategies depending on the power needs of IoT applications.

When creating low power IoT nodes, hardware architecture, operating systems, applications, and wireless technologies are crucial, much as semiconductor technology is. Leakage current can be used in VLSI chips, for instance, to minimize power consumption by shrinking transistor size. Reduced power supply prevents overheating of the equipment and the impact of high electric field on small electronics. The main focus of chip manufacturers is high performance processors; as a result, processor architecture optimization is of utmost importance.

Electronic device power consumption can be divided into two categories: dynamic and static [1]. The shifts of logic states caused by active calculations determine the dynamic power usage. When elements are turned on to hold or maintain the logic states in-between switching events—even when no computation is being done on them—static power consumption results.

The total power consumption in the sensor node is the sum of static and dynamic power consumption as follows:

$$P_{total} = P_{static} + P_{dynamic}$$

$$P_{total} = \left(P_{leakage} + P_{bias}\right) + (P_{sc} + P_{sw}) \qquad (1)$$

The static powerconsumptionP_{static}has two parts: leakage power and bias power, while the dynamic power consumption $P_{dynamic}$consists of short-circuit power and switching power. Among these four sources of power consumption, leakage power $P_{leakage}$ and switching power $P_{sw}$are currently the dominant ones which causes approximately more than 95% the total power consumption[2].

This paper discusses decrease in the power consumption P_{sw} using several architecture level low power design techniques. The computation of dynamic powerconsumption for performing the task ta is given as follows:

$$P_{dynamic} = 0.5CV_{dd}^2\alpha f \qquad (2)$$

where C is the capacitance at the output of the node, V_{dd} is the supply voltage, α is referred as the switching activity and is the average number of transitions per cycle time and f is the operating frequency. Switched capacitance can be reduced through logic optimization or by turning off clock signals to parts of circuits that are idle to avoid unnecessary switching, thereby reducing power consumption.

From eq. (2), reducing operating frequency causes reduction in power consumption, but does not reduce energy consumption. Hence, reducing supply voltage $V_{dd}$of circuit is the only way to reduce energy consumption. Reduction in supply voltage needs reduction in operating frequency to ensure correct operation of the circuit [3].

Table 1 illustrates low power design approaches for both static and dynamic power usage. The strategies that have been employed thus far for low power design at the system level are presented in the following portion of this study. The main focus of this study is on low power architecture level techniques because circuit/transistor level design approaches will require a different survey.

Table 1: Overview of Commonly used Low Power Design Technique[1]

Techniques	Description
Clock gating	Disables clock when digital logic is not in use.
Power gating	Reduces leakage power by turning of supply voltage to blocks not in use. Output of these block are required to be isolated before connecting to active block.
Multiple threshold voltages	Different threshold voltage levels are used in circuit keeping timing constraints in check, resulting leakage reduction.
Gate sizing	Upsizing reduces dynamic power and downsizing reduces leakage power.

Logic restructuring	Gate level dynamic power reduction is achieved by moving high switching logic to front and low switching logic to back.
Multiple supply voltage	Fixed but different supply voltage levels are given to different block domains. Signals those travel from one domain to another domain are need to be level-shifted.
Operand isolation	Forbids switching of inactive Datapath element.
Voltage scaling	Different blocks are catered dynamically based on performance requirements by variable supply voltage.
Frequency scaling	Frequency scaling is done along with voltage scaling dynamically.
Substrate biasing	Transistor threshold voltage is raised dynamically when it is in inactive mode by biasing the substrate which reduces leakage.
Memory partitioning	Memory is divided into quite a few blocks and is power down when not required.
Bus segmentation	System bus is divided into several segments. These segments are charged upon an access.
Hardware acceleration	Dedicated hardware is used to accelerate tasks. This hardware is power down when not in use.
Reconfigurable hardware modules	Particular hardware is reconfigured when required. This hardware design is stored in the form on Net list.

2. POWER MANAGEMENT TECHNIQUES

Through actuators and sensors, the IoT sensor node interacts with its surroundings and communicates wirelessly with other nodes. Any sensor node's standard configuration includes a radio transceiver module, battery, DC-DC converter, memory, actuator and sensor interfaces, processor, and memory. Power consumption is being prioritised over other factors like area, performance, and cost in the growing market for high-performance SoCs (System on Chips) in communication and computing [4] [5]. By lowering dynamic power consumption, wireless sensor nodes' autonomy can be increased. There are a tonne of low-power design strategies available, and a lot of research is being done on how to employ them automatically by the node itself [6]. When creating an RTL model of a power-controlled system based on an abstract specification, the proposed automation method in [6] uses the electronic system level (ESL) design flow. After RTL synthesis, a high-level synthesis for power management creates the system's components, which are then further integrated into the functional model. Although less power was used in this activity, physical labour was still necessary, and there was only a limited amount of support from power reduction techniques.

For a very long time, it has been difficult to design hardware with low power consumption in mind. Some strategies, such operand isolation or clock gating, are simple to use, while others are highly challenging. UPF- Unified Power Format, the industry standard for the design and verification of low-power integrated circuits, can be used to overcome this challenge [7]. This standard limits the use of low power design techniques during the design phase. The most crucial idea behind UPF is that it offers methods for dividing a system into different power domains. A power domain consists of a number of units that operate at the same supply voltage. Some of the power domains are using the same or various voltage levels to operate. In the early phases of VHDL modelling, UPF offers constructs for power management components like level shifters that allow designers to organise blocks into power domains and set voltage levels for each one.

The next subsections of this paper cover each strategy in turn.

A. Clock Gating: Clock transitions cause significant power usage. Clock gating is one of the most effective and extensively utilized low-power strategies. It is based on the notion that many transitions are superfluous. The functionality is unaffected by suppressing such transitions. Any functional component's utilization is dependent on the application. As a result, by blocking the clock from moving through the circuits, it is possible to turn off inactive circuits.

The amount of power that can be saved is significantly influenced by the circuit block's graininess where clock gating is implemented. Larger blocks that are clock gated save more energy but allow for fewer off cycles. It has three levels of granininess-

1. Module level clock gating
2. Register level clock gating
3. Cell level clock gating

Power Management (PM) module, implements operand isolation and clock gating to lower power consumption in baseband processor modules [5] The Wake-up Identification Receiver and Non-volatile memory are utilized as additional hardware in [5] so that the main transmitter and receiver can be powered down. All modules other than the RF wake-up module are forced into a power-down condition by the PM module. In general, operand isolation is used in arithmetic modules where there is a chance to lower internal and dynamic power consumption [8][9][10][11].

A proposed clock gating cell with power gating mechanism is illustrated in [12] as an illustration of cell level clock gating. There are three operating modes for the proposed cell: working mode, disable mode, and sleep mode. In this case, power gating is utilised to lower power usage while in sleep mode. Future study can concentrate on rise/fall time balance and power reduction in sleep mode taking into account cell level clock gating.

This approach ensures a glitch-free enable signal during clock transitions, making it a highly relevant example of cell level integrated clock gating for dual edge triggered flip-flops.

In [14] and [15], module-based clock gating is demonstrated. High-level optimisations, according to Nidhi Khanna and D. K. Mishra [14], are accomplished at the RTL level where all logical operations are carried out in register blocks. It is demonstrated by a comparison of the various methods of clock-gating they have employed when coupled with a 16-bit ALU. According to the output of the frame identification module, a brand-new clock-gating module called BUFGCE is proposed in [15] to govern the clock of the demodulation module.

In their design of a 65-nm 32-bit MIPS and a 28-nm industrial network processor, Doron Gluzer and Shmuel Wimer [16] utilised data-driven clock gating (DDCG) and multibit flip-flops (MBFFs) approach to achieve low power. In order to reduce hardware overhead, data driven clock gating suppresses pointless clock pulses by using a single clock enabling signal for a collection of flip-flops. The clock is the main factor affecting dynamic power consumption in sequential circuits. Clock gating was employed in a smart grid circuit by Nandita Srinivasan et al. & all [17], and the results show that it reduces power while increasing area.

In order to achieve quicker closure, several academics are switching to design flows based on HLS (High Level Synthesis). A unique system-level design methodology was put out in [18], where a relative power reduction model' is used to simulate the design at a cycle-accurate transaction level and then forecast the effect of clock-gating on resister/bank. This facilitates the automatic application of clock-gating to appropriate registers.

Clock gating can be implemented into the synthesis stage of a high-level dataflow design flow in the case of streaming application domains such as signal processing, digital media coding, video analytics, etc. [19]. Clock enabler controller finite state machines were employed in the design of video decoders by E. Bezati et al.

Hai Li et al. [20] presented a deterministic clock-gating (DCG) technique in microprocessor design because hardware is crucial in IoT applications. The circuit block utilisation is forecast a few cycles in advance in this study based on pipelined stage operation. Intel DCG chooses to clock-gate unoccupied blocks in light of this. The area overhead is DCG's only flaw.

Sequential circuits can use clock-gating, as seen in [21] [22]. The clock's behaviour when activating the flip-flops is seen in [21].

B. Power Gating: Another technique used to prevent leaky power consumption when specific blocks are not used in particular modes of operation is power gating (PG). Power gating involves using a power switch to completely turn off the power supply to power gated blocks, virtually eliminating leakage. For efficient low power designing, [23] explains many forms of power gating.

By calculating at top speed and then power-gating those modules when not needed for a while, Energy Proportional calculating (EPC) can be accomplished. In their dynamic work, Mohammad Hosseinabady and Jose Luis Nunez-Yanez [24] have used power-gating. There is a control overhead for programmable voltage regulators, and then there is a reconfiguration overhead.

The IoT market is expanding, necessitating security in data transfer, necessitating data encryption design. Transmission power consumption increases as a result of several repeat cycles in encryption. As a result, [22] can employ the power management strategy to lower power usage. Clock gating, power gating, and dynamic voltage frequency scaling are all utilised in this work's data encryption architecture.

Power gating and parallelism can work together to achieve low power and high performance [25]. By dividing the application into various power gating areas, Yufeng Tong and colleagues sought to reduce energy consumption without compromising execution speed. Parallelism is modified for power gating zones once the needs for resources and energy efficiency have been examined. After rescheduling each zone with its own parallelism, additional power gating instructions are given to manage application hardware.

Because the radio component of the Internet of Things uses a lot of power, Jean-Francois Pons et al. [26] proposed RF power gating. During symbol time, every component of the RF front-end is either turned ON or OFF. Digital blocks that use minimum-shift keying (MSK) modulation are used in this study to gate RF power. As an alternative to voltage and frequency scaling, a unique cyclic power gating (CPG) technique has been put out in [27]. With CPG, the power supply of the core can be turned on or off while operating at high speed with hardly any impact on programmes already in use. By adjusting the on-off ratio of the duty cycle within a single power-gating interval, the effective frequency of the CPU and power consumption can be managed. Despite the fact that this work is a good substitute for voltage frequency scaling, state-retentive architecture is required to preserve the core's state during power gating [28].

C. Frequency scaling: Since frequency is one of the variables that affect dynamic power consumption, it is possible to lower a device's frequency when it is not required to run at a high frequency while inactive. In [29] Q. Si et al. proposes a technique for dynamically creating microarchitecture at runtime that changes its operating frequency to cut down on power or energy usage. A reconfigurable FPGA's dynamic frequency controller chooses a microarchitecture that is appropriate for the operating frequency.

Frequency scaling is used in a number of works to increase throughput and decrease energy or power consumption, including mobile edge computation (MEC) [30][31], asynchronous circuits [32], and RFID computing devices [33].

D. Voltage Scaling: Supply voltage has a significant impact on power consumption. In order to reduce power usage, supply voltage scaling might be used. Based on implementation, there are many forms of voltage scaling.

Static voltage Scaling (SVS) In this strategy, several fixed voltages are applied to various blocks or subsystems within a huge system. Supply voltage is decreased during design time based on the performance need, and that voltage is maintained during operation. This method is referred to as static voltage scaling since the voltage is not altered while it is in use. Different blocks with various voltages are possible. Device feature size scaling and architecture level methods, such as parallelism and pipelining, can both be used to achieve SVS.

Although the primary goal of parallel processing is to boost performance, parallelism can also result in low power usage. The primary strategy is to retain the same throughput while trading space for power. Simply put, when supply voltage is halved, power is reduced by a fourth and performance is decreased by half. For instance, multicore for low power.

A sensor node in [34] has two microcontrollers or microprocessors, and they both share components such communication modules, power supplies, sensors, and actuators. Each of them is responsible for overseeing the proper workload time. A microcontroller controls the higher workload phase, and a counter controls the lower workload phase. Since no processing power is required for the relative workload period, a counter is used instead of a real microcontroller.

A method of implementation called "piping" involves performing several activities concurrently [35]. It can be broken down into two or more smaller jobs that can be carried out separately. The hardware units that conduct these various duties are referred to as stages. Each step's output is briefly buffered in latches before being forwarded to the following stage. Using voltage scaling in conjunction with pipelining can assist reduce power consumption. Voltage scaling can be used with parallel processing and pipelining to further reduce power consumption.

The Luffa hash function is implemented in hardware for low power consumption in [36]. A pipeline technique using just positive edge flip-flops is used for Luffa positive. Similar to how the pipeline technique with clock gating is implemented, Luffa gating is used to represent it, as is Luffa negative for the pipeline technique with positive and negative edge flip-flops.

Multilevel Voltage Scaling (MVS): The supply voltage is switched between two or a few set voltages in this expansion of SVS. This approach employs two or a few fixed voltage domains in various circuit elements. At different coarse levels, such as the macro level or the standard cell level, voltage islands are produced. The overall performance of the circuit can be improved without reducing total power consumption. It is possible to use more than two supply voltages but the benefit of multiple V_{dd}saturates quickly. The major gain is obtained by moving from single a singleV_{dd} to dualV_{dd}.

Globally Asynchronous and Locally Synchronous (GALS), which reduces the number of global interconnects while achieving low power consumption and modularity of a system, is proposed in [37]. Multiple clock domain architectures can have a range of frequency/voltage values for each domain depending on the demands of the task. The possibility of having various clock frequencies for each domain also makes it possible to create designs that are power-conscious. Frequency and voltage scaling are made possible by voltage-frequency islands (VFIs).

Dynamic Voltage and Frequency Scaling (DVFS): A wide range of voltage levels are dynamically applied in this MVS extender for various workloads. This technique allows for the usage of several voltage and frequency pairs. Voltage and frequency can be dynamically adjusted while running depending on the workload. While designing, these voltage and frequency pairs are chosen. The workload predictor, variable frequency generator, variable voltage processor, and variable voltage generator are the basic parts of DVFS. In [38] a DVFS controller is designed that achieves low cost overhead in terms of complexity and power consumption. The look-up table (LUT) for speed setting, workload, and deadline tracking are all features of the controller. Voltage-controlled oscillator (VCO) frequency is monitored by a frequency counter. The workload and deadline tracker keeps track of workloads and calculates scaling factors. In order to achieve good system performance, scheduling techniques are integrated with dynamic voltage and frequency selection techniques [39] [40] [41] [42]. Decisions about task scheduling and DVFS are based on [39]'s short-term projection of the energy harvesting rate.

A hardware-based control method that dynamically chooses the operating frequency and voltages for individual VFIs is presented in [37] for a VFI-based system. The advantage of the hardware-based method is that it equips the system with the required building pieces for fine-grained application workload monitoring. Decisions about the choice of new frequency and voltage values for different VFIs are made locally as well as globally using the information collected by such blocks at a fine-grain level. [37] presents a mixed-clock/mixed-voltage first-in, first-out (FIFO) architecture that enables dynamic scaling of frequency and voltage of various VFIs.

Adaptive Voltage Scaling (AVS): A control loop is utilised in this expansion of DVFS to modify voltage and frequency in response to shifting workload. It resembles a closed loop approach in many ways [43]. During operation, the circuit's behaviour is observed, and the voltage and frequency are adjusted as necessary [3]. At execution time, a closed-loop feedback system is set up between the delay detecting performance monitor and the voltage scaling power supply.

In [44], a multi-hop routing-based task-driven feedback dynamic voltage scaling technique is described. It includes a Proportional Integral Derivative (PID) feedback control model, an Execution time model, an Energy consumption model, and a Sensor Node Task model. Earliest Deadline First (EDF) scheduling, which assigns the highest priority task with the earliest deadline, is used in the execution time model. The difference between the actual execution time and the prescribed execution time is reduced using the PID feedback control approach. To fix the voltage and frequency level for the real-time task, the feedback interval is scaled in accordance with the heterogeneity of the sensor node's workload, and the assigned execution time is updated with the aid of the feedback control method.

In [45] C. T. Chow et. al. proposes a logic delay measuring circuit (LDMC) that, at runtime, estimates the speed of an inverter chain under a variety of operating conditions. A dummy circuit's on-chip delay is calculated by LDMC. This intended LDMC value should match the working circuit's critical path and a safety margin. The closed loop control system automatically modifies the voltage delivered to the FPGA to maintain the desired value of LDMC as the chip temperature changes.

Voltage-frequency scaling is frequently used in conjunction with other low-power strategies, including per-core power gating, to balance system performance and power consumption in large-scale systems [46]. According to [47], bit error rate and system operational conditions affect voltage scaling to a SoC. Timing errors that occur in circuits are removed using the forward error correction approach. To provide logic scalability, in-situ detectors identify valid voltage-frequency pairings during run-time and partial dynamic reconfiguration [48].

E. Memory Partitioning: Memory can be divided into different banks so that each one can be accessed separately, reducing the amount of dynamic power used [49]. One bank is always active per access, while the other banks can be power-gated to cut down on static electricity. The processor in MPSoCs makes advantage of a variety of resources, such as processing elements, to increase speed and performance, which reduces execution time and energy consumption. Using off-chip memory helps speed up data retrieval. With the integration of Scratch Pad Memory on-chip memory, partitioning and job allocation can be done independently [50].

A significant amount of data can be reused in loop pipelining in some applications that perform image and video processing. In [51] study, it is suggested to cache reusable data using on-chip registers, and to segregate non-reusable data, memory partitioning method is utilised to establish multiple memory banks. To divide memory into smaller pieces, several works suggest linear transformation-based memory partitioning [52] [53]. Memory duplication and partitioning work in tandem to lessen memory usage and interference in order to improve performance [54]

F. Bus Segmentation: By reducing switched capacitance on the bus, bus segmentation can lower the power that the bus uses for data communication [55]. Bus-segmentation divides the bus into numerous segments using pass transistors, reducing the amount of switching and critical path required for data communication [56]. According to another study [57], bus supply power gating is a type of bus segmentation.

G. Hardware acceleration: An accelerator is a distinct architectural substrate that is designed with different goals in mind than the base processor [58], goals that are derived from the needs of a certain class of applications.

Based on the requirements of specific application sectors, accelerators can be developed for fixed function, special-purpose chips or highly programmable engines. As a lot of calculation must be done to generate a key and given the computationally expensive nature of cryptographic methods, hardware implementation is a preferable option to software in IoT applications [59]. In various works, custom instruction set extensions is suggested to speed up programmes like floating point coprocessors, cryptography, and security software. The well-known machine to machine communication (MMC) technique known as narrow band IoT (NB-IOT) uses a power-hungry component called the Viterbi decoder. A repeated pattern-based, completely parallel Viterbi decoder for NB-IOT was proposed as hardware acceleration by Mamdouh H. Ellamei1 and Mohamed A. Abd El Ghany [64].

H. Reconfigurable Hardware Modules: Due to its ability to instantly change the micro-architecture, RISC32 has recently attracted a lot of attention as an IoT processor for the creation of FPGA-based sensor nodes. Modifiable hardware modules are saved as bitstreams so that they can be modified as needed. The FPGA-based IoT sensor node with an energy-saving programme analyzer is proposed in [3], where dynamic voltage-frequency scaling, clock gating, and partial reconfiguration are employed to cut down on dynamic power usage. In order to switch between the two micro-architectures (pipeline and multicycle), partial reconfiguration is used. Toggled Microarchitecture (TMA) is dependent upon the characteristics of the given tasks [65]. A whole FPGA reconfiguration is suggested in [66], where many bitstreams are produced off-line. Each bitstream is associated with a certain circuit configuration. An external controller downloads these bitstreams and then configures the FPGA to meet the circuit's demands for low power consumption. The main drawback of a complete FPGA reconfiguration is that it prevents circuit operation while the bitstream is changed.

Tamimi et al. [67] proposed the creation of a soft-core CPU using reconfigurable architecture. Infrequently used functional units are included into reconfigurable units (RUs) based on look-up tables (LUTs). These less frequently utilised functional units are only set up when essential to complete particular tasks. L. Sterpone et al. have created partially reconfigurable clock gating of clock routing resources by directly influencing the configuration memory's contents [68].

3. IOT CHARACHTERISTICS

One of the disruptive technologies that have changed life, business, and the global economy is the Internet of Things. Efficiency in the gathering and transmission of complex data is required to meet the rising demand for high-performance IoT applications. Examples include Internet of Things (IoT) applications for medical diagnosis, high-resolution images of several gigabytes obtained from affordable portable MRI devices and portable ultra-sound machines, which will be communicated to medical professionals for remote data processing for diagnosis. In real-time medical applications, data transmission is difficult because latency must meet a rigorous deadline.

Data transmission energy use exceeds energy utilised for data computation [69] by a significant margin. [70]. When the transmission range and particular calculation are taken into account, the energy needed by Rockwell Automation's sensor nodes to send a single bit of data is greater than the energy needed to execute a single instruction [71].Edge computing, where edge nodes contain sufficient compute capability resulting in minimised data transmission, is presented as a solution to the problems of reducing bandwidth, latency, and energy consumption [72].

The following properties of the IoT set it apart from other connected systems, according to prior research [73]: intelligence, heterogeneity, complexity, size, real-time limitations, and spatial constraints.

1) Intelligence: The IoT minimises human intervention in data collecting and processing to provide information that can be used for decision-making. To reduce power usage, IoT must dynamically react to changes in system requirements [3].

2) Heterogeneity: The Internet of Things (IoT) includes a variety of various types of devices, applications, and environments. Different IoT application criteria must be met by IoT microprocessors. For instance, an IoT microprocessor with heterogeneous cores may be a single chip or may have many cores with various device features [74].

3) *Complexity:* IoT should be adaptable in order to run a variety of applications, in addition to handling multiple architectures.

4) *Scale:* Continuous growth of IoT nodes is anticipated, along with an increase in data interchange among them. IoT microprocessors must therefore have low overhead

and be cost, power, and area efficient.

5) *Real-Time restrictions:* Real-time applications, such as those that monitor patients or aircraft, are subject to real-time restrictions. As a result, fulfilling the execution deadline is crucial in these applications.

IoT nodes are exposed to a variety of environmental conditions, the majority of which are not optimal. For instance, tracking devices may have a shorter lifespan if they are exposed to heat, cold, and rain at different times of the year and in different locations.The environment's electromagnetic radiation also has an impact on the throughput of nodes because it might lead to data transmission errors. IoT nodes must therefore be fault tolerant and able to adapt to changing operational circumstances.

4. POWER HUNGRY IOT APPLICATION FUNCTIONS AND REMEDIES

IoT applications are divided into three categories based on their use cases and domains: industry, environment, and society [75][76]. These applications will determine how each node performs. We looked at application functions accountable for excessive power consumption while analysing IoT applications. These functions are classified as follows.

1. Computation.
2. Communication.
3. Security.
4. Fault tolerance.

1) Computation: In IoT applications, computation is essential. IoT applications today are anticipated to have all the necessary capabilities to run computations and algorithms, allowing the device to collect data and analyse it with less energy or space usage.[77].

In IoT applications, reconfigurable designs can offer a compromise between performance, area, and power. Reconfiguration refers to an IoT node's capacity to dynamically configure its microarchitecture or other functional building components. Issue queue [78], register files [79], customizable caches [80], configurable micro-architecture [3,] and configurable floating-point unit [82] are a few examples. Configurable caches are crucial in this situation since memory in the IoT node has a significant impact on size, energy use, and performance. IoT nodes must have more advanced memory hierarchies due to the computational intensity of newly growing IoT applications. Memory hierarchy has an impact on system performance and energy use; hence IoT node CPU caching solutions must be a priority.

Configuring reconfigurable modules to work with the rest of the hardware is a major task. The full potential of configurable modules can be realised with less overhead to enable configuration. Similar to configurable caches, bit-width configuration registers are utilised to facilitate configurability by allowing cache banks to shut down in order to modify cache size [80].

Cache closes the performance gap between the processor and memory, but processing in memory closes this gap even further by carrying out computation on the memory chip without processor to memory connection. As frequent data movement results in power consumption, previous work demonstrates that in-memory processing (or processing in memory) has been examined in large data and distributed computing systems [83]. Real-time in-situ data processing is required to maximise the power of the IoT node with minimal hardware overhead given the recent expansion of IoT, the enormous volume of data created, and the resource limitations of IoT devices [84] [85].The processor-memory interface is completely eliminated thanks to the compute-in-memory design, which integrates memory and processing into one architecture [86] [87].

It is possible to outfit the microprocessor with several core configurations or core types thanks to heterogeneous architectures [88]. While carrying out the identical command, different cores have varied capabilities and performance levels. System software selects the core that is appropriate for the current execution based on resource requirements during execution to achieve low power [3], [89]. The number and kind of cores, as well as properly scheduling the programme, present the most design issues for IoT applications with heterogeneous cores [90]. To add cores into the microprocessor, it is vital to have a thorough understanding of application needs. By having an understanding of the programme beforehand, one can select the best core on which to statically or dynamically schedule the application [91].Dynamic scheduling assesses application characteristics while the application is running and selects the best core, as opposed to static scheduling, which is done when application characteristics are known.

Using a distributed network of IoT nodes with heterogeneous architectures is a substitute for heterogeneous architecture built into microprocessors. Each node has a variety of computational resources that can be employed as needed at various times. Consider a scenario where a deadline-driven application arrives on one of the nodes, each of which has a different type of microprocessor. An alternate network node that is appropriately supplied may be employed if the relevant node is not adequately equipped for execution [72]. Waiting for the right-provision node to become free if the node is busy is a major challenge when enabling distributed heterogenous architectures.

To store processor state during power interruption, non-volatile processors use non-volatile memory. Therefore, despite a power interruption, non-volatile processors can continue to operate. The majority of embedded systems use a lot of power while they are not in use, yet non-volatile processors are useful for decreasing idle power by turning off the processor. On waking, the processor state can be recovered [92].

2. Communication: The IoT apps' greatest power-hungry feature is communication. Numerous communication technologies (such as Bluetooth and Wi-Fi) and communication protocols (such as TCP and 6lowpan) are available. Software defined Radio (SDR) has been quite popular recently because to its versatility in terms of a greater variety of frequencies, coding, and modulation schemes [93]. An antenna, an analogue to digital converter for receiving analogue signals, and a digital to analogue converter for sending signals are often used with SDR. Operations for digital signal processing change input signals into any format needed by applications [94].

Previous research suggests that by optimising crucial kernels like synchronisation and finite impulse response with an eye towards SDR computation and power conservation, SDR may require less overhead in the IoT sector [95]. DSPs, FPGA, or general-purpose microprocessors are all capable of effectively running SDR algorithms. Heterogeneous architectures can also be used since they can carry out various activities depending on the needs of the execution, resulting in minimal overhead and energy expenditure.

The concept of the perfect SDR is approximated by the all-digital transmitter (ADT). It offers a good deal of carrier flexibility and shows promise in a wide range of applications [96]. It creates the signal to be broadcast in the digital domain and converts it to the analogue domain using single- or low-bit count DACs. via a single-bit DAC, a signal is transmitted via delta-sigma modulation (DSM). A multi-bit signal is reduced to a single-bit signal via DSM. Prior to DSM conversion, the digital signal processing blocks upsamples the signal. Due of high clock rates that lead to significant digital resource demand and high power consumption, DSP blocks are frequently implemented utilising polyphase techniques [97]. When necessary, higher carrier frequencies and wider bandwidths increase power consumption [98], [99].

To modulate baseband signals, [100] proposes a behavioural model of a bandpass DSM. Without installing its complicated hardware, the LUT hardware stores the transmitter's behavioural model. By encapsulating the digital upconversion and upsampling, the LUT minimises the amount of real-time computing machinery that ADT has to be implemented in FPGA. The open-source RISC-V ISA with an instruction set expansion for ultra-low power SDR was proposed by Hela Belhadj Amor et al. The instructions are specifically designed to meet the requirements of wireless DSP, which provides significant cycle count reductions with "near zero" power overhead [101].

By adopting effective data compression techniques, transmission delay and bandwidth costs in communication can be reduced while using less power. Data compression can lower connectivity needs for emerging IoT applications, assuring rapid data retrieval, transmission, and analysis [102]. When data needs to be stored on an edge node, compression also minimises the amount of storage needed.

Data information is compressed to use fewer bits for encoding than the original data representation [103]. Source encoding refers to data that is encoded at the source end prior to storage, whereas channel encoding refers to data that is encoded while being sent.

3. Security: Device and data security is required because IoT devices are susceptible to hostile assaults. For instance, security applications are required to guard against unauthorised access to critical medical diagnostic data and functionalities. Data secrecy is often achieved through the use of data encryption. Data that is encrypted by an encryption algorithm can only be used again after being decoded. Since the speed of encryption is influenced by memory access latency for data storage and retrieval, data encryption applications are typically memory and compute intensive. Jae Seong Lee and colleagues [104] have developed an advanced encryption standard (AES) accelerator that uses the SRAM to store both the input key and enlarged keys. Without a bus, data can be transported directly between the SRAM and AES accelerator. Less energy is consumed since data transport requires fewer resources and takes less time.

A low-power SBox, power gating methodology, and power management method are used in the low-power consumption AES data encryption architecture (LPADA) that is proposed in [105]. An AES-128 co-processor is built into the RISC-V host processor, the RISC32-E, according to et al.'s [106] introduction. If employed to optimise a particular functionality in a system, function-specific hardware modules can lower energy consumption. The unique system that is suggested in this work improves data processing speed while using 16% less energy.

IoT devices are vulnerable to network, software, physical, side channel, and cryptoanalysis assaults. As a result, security in microprocessor design is essential, however difficult given the strict resource limitations of these devices. Most Internet of Things (IoT) devices lack improved security features like trusted execution [107]. In IoT applications, the processing power of the microprocessor may degrade as it meets the resource needs of security processes and algorithms. These devices also produce large amounts of data, some of which may be quite sensitive [108], necessitating trade-offs between energy use, performance, and cost [109].

IoT devices must be versatile and adaptable to various device interconnections and protocols in order to support hardware-based security [110]. As a result, IoT devices must include hardware security policies that may be configured at runtime to accommodate changing security requirements [111]. These security standards can be modified to produce secure microarchitectures that can change to meet application requirements. It is suggested [111] that customizable hardware security in microprocessor design, which may achieve numerous optimisation goals including energy consumption, be used in particular for IoT devices. Configurable cache was suggested by R. Zhuang, S. A. DeLoach, and X. Ou [112] as a moving target defence against side-channel attacks in caches.

4. Fault Tolerance: This refers to a system's capacity to continue functioning flawlessly even if some of its components malfunction [113]. The harsh and unsupervised environmental conditions, that IoT devices are subjected to include high temperatures, shock, vibration, electromagnetic radiation, etc. In order to maintain service quality, fault-tolerant applications are crucial. As a result, defects are produced. For instance, fault tolerance must be built into IoT equipment utilised in the automobile and health industries.

Fault tolerance can be achieved in a variety of ways. Redundancy is typically a component of hardware-based fault tolerant solutions. IoT devices may feature redundancy at the expense of additional space and power requirements [114] [74] [115]. Tanya Mendez and Subramanya G. Nayak offer three ideas for fault-tolerant adders with decreased switching activity and gate count [116]. A pair of high-performance out-of-order cores, with a group of small low-power cores was proposed by Sam Ainsworth and Timothy M. Jones [115]. The execution of each high-performance out-of-order core is checked, allowing both hard and soft mistakes to be found.

In large-scale IoT systems with heterogeneous device connectivity, malfunctions can spread along the link and have an impact on the entire system [117]. The IoT system must therefore be watched over. For extensive IoT-driven applications, a RISC-V-based optimised low-power reconfigurable fault-safe CPU platform is suggested [117]. Software-based fault tolerance can be used to lower the costs associated with hardware-based fault tolerance [113] [118] [119].

5. CONCLUSION

The Internet of Things is anticipated to expand quickly and generate enormous amounts of data, which will cause bottlenecks in latency, power consumption, and connection capacity. By carefully designing IoT devices, it can be decreased. An overview of low power design strategies is provided in this paper in order to support IoT, scale up, and have power optimisation.

Researchers can use the survey results as a starting point to create safe, scalable, and effective IoT applications. Solutions for power or energy optimisation are presented along with IoT features that need more power.

References

[1] D. Macko,“Contribution to System-Level Design and Verification of Low-Power Digital Systems”Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 7, No. 3-4 (2015) 10-17.

[2] V. T. Hoang, N. Julien, and P. Berruet, “Increasing the autonomy of wireless sensor node by effective use of both DPM and DVFS methods,” *2013 IEEE Faibl. Tens. Faibl. Consomm. FTFC 2013*, 2013, doi: 10.1109/FTFC.2013.6577766.

[3] B. L. Tan, K. M. Mok, J. J. Chang, W. K. Lee, and S. O. Hwang, “RISC32-LP: Low-Power FPGA-Based IoT Sensor Nodes with Energy Reduction Program Analyzer,” *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4214–4228, 2022, doi: 10.1109/JIOT.2021.3103035.

[4] S. E. Schulz, “Low Power Design And Verification,” White Paper, www.mentor.com, 2008.

[5] Y. Shuangming, F. Peng, and W. Nanjian, “A low power non-volatile LR-WPAN baseband processor with wake-up identification receiver,” *China Commun.*, vol. 13, no. 1, pp. 33–46, 2016, doi: 10.1109/CC.2016.7405702.

[6] M. Škuta, D. Macko, and K. Jelemenská, “Automation of dynamic power management in FPGA-based energy-constrained systems,” *IEEE Access*, vol. 8, pp. 165894–165903, 2020, doi: 10.1109/ACCESS.2020.3022955.

[7] ISO 7267, IEEE Standard for Design and Verification ofLow-Power Integrated Circuits, vol. 2003. 2003.

[8] L. Siddhu, A. Mishra, and V. Singh, “Operand Isolation Circuits with Reduced Overhead for Low Power Data-Path Design,” 2014, 2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems, doi: 10.1109/VLSID.2014.90.

[9] A. Chattopadhyay *et al.*, “Automatic ADL-based Operand Isolation for Embedded Processors”, Proceedings of Design Automation and Test in Europe Conference, 6-10 March 2006.

[10] N. Banerjee, A. Raychowdhury, S. Bhunia, H. Mahmoodi, and K. Roy, “Novel Low-Overhead Operand Isolation Techniques for Low-Power Datapath Synthesis,” Ieee Transactions On Very Large-Scale Integration (Vlsi) Systems, Vol. 14, NO. 9, SEPTEMBER 2006.

[11] J. Brandt and K. Schneider, “The Model Checking View to Clock Gating and Operand Isolation”, 10th International Conference on Application of Concurrency to System Design, doi: 10.1109/ACSD.2010.22.

[12] W. Healthcare, “A 0 . 7-V Clock-gating Cell with Power Gating Technology and 1 . 56-pA Sleep,” IEEE International Conference on Integrated Circuits, Technologies and Applications, pp.2020–2021, 2020.

[13] T. Noor and E. Salman, “A novel glitch-free integrated clock gating cell for high reliability,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2019-May, 2019, doi: 10.1109/ISCAS.2019.8702507.

[14] N. Khanna and D. K. Mishra, “Clock Gated 16-Bits ALU Design & Implementation on FPGA,” 4th International Conference for Convergence in Technology (I2CT) SDMIT Ujire, Mangalore, India. Oct 27-28, 2018 pp. 1–5, 2018.

[15] J. Zhang *et al.*, “A Clock-Gating-Based Energy-Efficient Scheme for ONUs in Real-Time IMDD OFDM-PONs,” Journal Of Lightwave Technology, Vol. 38, NO. 14, JULY 15, 2020.

[16] D. Gluzer and S. Wimer, “Probability-Driven Multibit Flip-Flop Integration with Clock Gating,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 3, pp. 1173–1177, 2017, doi: 10.1109/TVLSI.2016.2614004.

[17] N. Srinivasan, N. S. Prakash, Shalakha D., Sivaranjani D., S. Sri Lakshmi G., and B. B. T. Sundari, “Power Reduction by Clock Gating Technique,” *Procedia Technol.*, vol. 21, pp. 631–635, 2015, doi: 10.1016/j.protcy.2015.10.075.

[18] S. Ahuja, W. Zhang, and S. K. Shukla, “System Level Simulation Guided Approach to Improve the Efficacy of Clock-gating,” pp. 9–16, 2010.

[19] E. Bezati, S. Casale-Brunet, M. Mattavelli, and J. W. Janneck, “Clock-gating of streaming applications for energy efficient implementations on FPGAS,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 36, no. 4, pp. 699–703, 2017, doi: 10.1109/TCAD.2016.2597215.

[20] H. Li, S. Bhunia, Y. Chen, K. Roy, and T. N. Vijaykumar, “DCG : Deterministic Clock-Gating for Low-Power Microprocessor Design,” vol. 12, no. 3, pp. 245–254, 2004.

[21] Q. Wu, M. Pedram, and X. Wu, “Clock-gating and its application to low power design of sequential circuits,” vol. 47, no. 103, pp. 1196–1199, 2000.

[22] K. Tsai, F. Leu, I. You, S. Member, and S. Chang, “Special Section On Security And Privacy In Emerging Decentralized Low-Power AES Data Encryption Architecture for a LoRaWAN,” *IEEE Access*, vol. 7, pp. 146348–146357, 2019, doi: 10.1109/ACCESS.2019.2941972.

[23] B. Kapoor, S. Hemmady, S. Verma, K. Roy, and M. A. D’Abreu, “Impact of SoC power management techniques on verification and testing,” *Proc. 10th Int. Symp. Qual. Electron. Des. ISQED 2009*, pp. 692–695, 2009, doi: 10.1109/ISQED.2009.4810377.

[24] M. Hosseinabady and J. L. Nunez-yanez, “Run-Time Power Gating in Hybrid ARM-FPGA Devices”,2014,24th International Conference on Field Programmable Logic and Applications(FPL) 2-4 Sept 2014.

[25] Y. Tong *et al.*, “Compiler-Guided Parallelism Adaption Based on Application Partition for Power-Gated ILP Processor,” IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 25, NO. 4, pp. 1329–1341, April 2017.

[26] J. Pons, N. Dehaese, S. Bourdel, J. Gaubert, and B. Paille, “RF Power Gating : A Low-Power Technique for Adaptive Radios,” IEEE Transactions On Very Larg Scale Integration (Vlsi) Systems, Vol. 24, NO. 4, pp. 1377–1390, April 2016.

[27] W. Toms, J. Navaridas, and M. Luj, “Cyclic Power-Gating as an Alternative to Voltage and Frequency Scaling n,” IEEE Computer Architecture Letters, vol. 15, no. 2, pp. 77–80, 2017, doi:

10.1109/LCA.2015.2478784.

[28] T. Kim, H. Park, and T. Kim, “Allocation of Always-On State Retention Storage for Power Gated Circuits - Steady-State- Driven Approach,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 29, no. 3, pp. 499–511, 2021, doi: 10.1109/TVLSI.2020.3047056.

[29] Q. Si, I. Rashid, and B. Carrion Schafer, “Micro-architecture Tuning for Dynamic Frequency Scaling in Coarse-Grain Runtime Reconfigurable Arrays with Adaptive Clock Domain Support,” *Proc. IEEE Comput. Soc. Annu. Symp. VLSI, ISVLSI*, vol. 2021-July, pp. 212–217, 2021, doi: 10.1109/ISVLSI51109.2021.00047.

[30] Y. Chen *et al.*, “TOFFEE : Task Offloading and Frequency Scaling for Energy Efficiency of Mobile Devices in Mobile Edge Computing,” IEEE Transactions On Cloud Computing, Vol. 9, NO. 4, October-December 2021vol. 9, no. 4, pp. 1634–1644, 2021.

[31] Y. Luo, L. Pu, and C. H. Liu, “CPU Frequency Scaling Optimization in Sustainable Edge Computing,” *IEEE Trans. Sustain. Comput.*, pp. 1–14, 2022, doi: 10.1109/TSUSC.2022.3217970.

[32] N. Srivastava, S. Member, R. Manohar, and S. Member, “Operation-Dependent Frequency Scaling Using Desynchronization,” IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 27, No. 4, ApriL 2019 vol. 27, no. 4, pp. 799–809, 2019.

[33] S. Li, S. Li, M. Chen, C. Song, and L. Lu, “Frequency Scaling Meets Intermittency: Optimizing Task Rate for RFID-Scale Computing Devices,” *IEEE Trans. Mob. Comput.*, pp. 1–12, 2023, doi: 10.1109/tmc.2023.3239515.

[34] M. Cerchecci, F. Luti, A. Mecocci, S. Parrino, and G. Peruzzi, “A Low Power IoT Sensor Node Architecture for Waste Management Within Smart Cities Context,” Sensors 2018, 1282; doi:10.3390/s18041282.

[35] W. Zhang, Y. Zhang, and K. Zhao, “Design and Verification of Three-stage Pipeline CPU Based on RISC-V Architecture,” *Proc. 2021 5th Asian Conf. Artif. Intell. Technol. ACAIT 2021*, pp. 697–703, 2021, doi: 10.1109/ACAIT53529.2021.9731161.

[36] P. Kitsos, “Low Power FPGA Implementations of 256-bit Luffa Hash Function,” *2010 13th Euromicro Conf. Digit. Syst. Des. Archit. Methods Tools*, pp. 416–419, 2010, doi: 10.1109/DSD.2010.19.

[37] H. Methods, P. Choudhary, S. Member, and D. Marculescu, “Power Management of Voltage / Frequency Island-Based Systems Using,” IEEE Transactions On Very Large Scale Integration (Vlsi) Systems,vol. 17, no. 3, pp. 427–438, 2009.

[38] R. A. Antonio *et al.*, “Implementation of Dynamic Voltage Frequency Scaling on a Processor for Wireless Sensing Applications,” Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, no. 1, pp. 2955–2960, November 5-8, 2017.

[39] S. Liu, J. Lu, Q. Wu, and Q. Qiu, “Harvesting-Aware Power Management for Real-Time,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 20, no. 8, pp. 1473–1486, 2012, doi: 10.1109/TVLSI.2011.2159820.

[40] W. Y. Lee, “Energy-Efficient Scheduling of Periodic Real-Time Tasks on Lightly Loaded Multicore Processors,” IEEE Transactions On Parallel And Distributed Systems, vol. 23, no. 3, pp. 530–537, March 2012.

[41] C. M. Krishna, S. Member, and Y. Lee, “Voltage-Clock-Scaling Adaptive Scheduling Techniques for Low Power in Hard Real-Time Systems,” IEEE Transactions On Computers, vol. 52, no. 12, pp. 1586–1593, December 2003.

[42] N. Gao, C. Xu, X. Peng, H. Luo, W. Wu, and G. Xie, “Energy-Efficient Scheduling Optimization for Parallel Applications on Heterogeneous Distributed Systems,” *J. Circuits, Syst. Comput.*, vol. 29, no. 13, pp. 3426–3442, 2020, doi: 10.1142/S0218126620502035.

[43] Nunez-Yanez , J . (2017). "Adaptive voltage scaling in a heterogeneous FPGA device with memory and logic in-situ detectors . University of Bristol - Explore Bristol Research Adaptive Voltage Scaling in a Heterogeneous FPGA Device with Memory and Logic In," pp. 227–238, 2017.

[44] W. Tuming, Y. Sijia, and W. Hailong, “A dynamic voltage scaling algorithm for wireless sensor networks,” *2010 3rd Int. Conf. Adv. Comput. Theory Eng.*, vol. 1, pp. V1-554-V1-557, 2010, doi: 10.1109/ICACTE.2010.5578956.

[45] C. T. Chow, L. S. M. Tsui, P. H. W. Leong, W. Luk, and S. J. E. Wilton, “Dynamic Voltage Scaling for Commercial FPGAs,” *Proceedings. 2005 IEEE Int. Conf. Field-Programmable Technol. 2005.*, pp. 173–180, 2005, doi: 10.1109/FPT.2005.1568543.

[46] S. Hou *et al.*, “Fine-Grained Online Energy Management of Edge Data Centers Using Per-Core Power Gating and Dynamic Voltage and Frequency Scaling,”IEEE Transactions on Sustainable Computing. pp. 1–14, 2023, doi: 10.1109/TSUSC.2023.3250487.

[47] Y. Wu, S. Thomson, H. Sun, D. Krause, S. Yu, and G. Kurio, “Free Razor: A Novel Voltage Scaling Low-Power Technique for Large SoC Designs,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 11, pp. 2431–2437, 2015, doi: 10.1109/TVLSI.2014.2377573.

[48] J. Luis Nunez-Yanez, M. Hosseinabady, and A. Beldachi, “Energy Optimization in Commercial FPGAs with Voltage, Frequency and Logic Scaling,” *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1484–1493, 2016, doi: 10.1109/TC.2015.2435771.

[49] S. Mai, C. Zhang, Y. Zhao, J. Chao, and Z. Wang, “An Application Specific Memory Partitioning Method for Low Power,”2007 7th International Conference on ASIC, DOI: 10.1109/ASIC12738.2007, 22-25 Oct. 2007, pp. 221–224.

[50] A. Poorani, B. Anuradha, and C. Vivekanadhan, “An Effectual Elucidation of Task Scheduling and Memory Partitioning for MPSoC,”, 2014 IEEE 8th Proceedings International Conference on Intelligent Systems and Control (ISCO), pp. 213–217, 2014.

[51] J. Su, F. Yang, X. Zeng, D. Zhou, S. Member, and J. Chen, “Efficient Memory Partitioning for Parallel Data Access in FPGA via Data Reuse,” IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, vol. 36, no. 10, pp. 1674–1687, OCTOBER 2017.

[52] Y. Wang, P. Li, P. Zhang, C. Zhang, and J. Cong, “Memory Partitioning for Multidimensional Arrays in High-level Synthesis,” 50th ACM/EDAC/IEEE Design Automation Conference (DAC)2013.

[53] M. Strobel, M. Eggenberger, and M. Radetzki, “Low power memory allocation and mapping for area-constrained systems-on-chips,” *EURASIP J. Embed. Syst.*, 2017, doi: 10.1186/s13639-016-0039-5.

[54] G. Jia and G. Han, “Coordinate Memory Deduplication and Partition for Improving Performance in Cloud Computing,” IEEE Transactions On Cloud Computing, vol. 7, no. 2, pp. 357–368, APRIL-JUNE 2019.

[55] H. Wang, A. Papanikolaou, M. Miranda, F. Catthoort, and U. Leuven, “A global bus power optimization methodology for physical design of memory dominated systems by coupling bus segmentation and activity driven block placement,” ASP-DAC 2004: Asia and South Pacific Design Automation Conference 2004, IEEE Cat. No.04EX753, pp. 759–761.

[56] J. Y. Chen, W. B. Jone, J. S. Wang, H. Lu, and T. F. Chen, “Segmented Bus Design for Low-Power Systems,” IEEE Transactions On Very Large Scale Integration(VLSI) SYSTEMS, vol. 7, no. 1, pp. 25–29, 1999.

[57] K. Heyrman, S. Member, A. Papanikolaou, F. Catthoor, and P. Veelaert, “Control for Power Gating of Wires,” IEEE Transactions On Very Large Scale Integration (VLSI) Systems, vol. 18, no. 9, pp. 1287–1300, September 2010.

[58] S. Patel and W. M. W. Hwu, “Accelerator architectures,” *IEEE Micro*, vol. 28, no. 4, pp. 4–12, 2008, doi: 10.1109/MM.2008.50.

[59] P. He and Y. Tu, “Hardware-Implemented Lightweight Accelerator for Large Integer Polynomial Multiplication,” IEEE Computer Architecture Letters, no. 1, 2023, doi: 10.1109/LCA.2023.3274931.

[60] V. Patil, A. Raveendran, P. M. Sobha, A. D. Selvakumar, and D. Vivian, “Out of order floating point coprocessor for RISC v ISA,” *19th Int. Symp. VLSI Des. Test, VDAT 2015 - Proc.*, pp. 1–7, 2015, doi: 10.1109/ISVDAT.2015.7208116.

[61] S. Majzoub and H. Diab, “Instruction-set extension for cryptographic applications on reconfigurable platform,” *Proc. - 6th IEEE Int. Work. Syst. Chip Real Time Appl. IWSOC 2006*, pp. 173–178, 2006, doi: 10.1109/IWSOC.2006.348231.

[62] R. Paludo and L. Sousa, “NTT Architecture for a Linux-Ready RISC-V Fully-Homomorphic Encryption Accelerator,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 69, no. 7, pp. 2669–2682, 2022, doi: 10.1109/TCSI.2022.3166550.

[63] F. Regazzoni and P. Ienne, “Instruction Set Extensions for secure applications,” *Proc. 2016 Des. Autom. Test Eur. Conf. Exhib. DATE 2016*, pp. 1529–1534, 2016, doi: 10.3850/9783981537079_1009.

[64] M. H. Ellamei and M. A. Abd El Ghany, “Hardware Acceleration of a Fully Parallel Viterbi Decoder Architecture for Narrow Band IOT,” *ICECS 2022 - 29th IEEE Int. Conf. Electron. Circuits Syst. Proc.*, pp. 1–4, 2022, doi: 10.1109/ICECS202256217.2022.9970985.

[65] W. P. Kiat, K. M. Mok, W. K. Lee, H. G. Goh, and R. Achar, “An energy efficient FPGA partial reconfiguration based micro-architectural technique for IoT applications,” *Microprocess. Microsyst.*, vol. 73, p. 102966, 2020, doi: 10.1016/j.micpro.2019.102966.

[66] J. Becker, M. Hübner, G. Hettich, R. Constapel, J. Eisenmann, and J. Luka, “Dynamic and partial FPGA exploitation,” *Proc. IEEE*, vol. 95, no. 2, pp. 438–452, 2007, doi: 10.1109/JPROC.2006.888404.

[67] S. Tamimi, Z. Ebrahimi, B. Khaleghi, H. Asadi, and S. Member, “An Efficient SRAM-Based Reconfigurable Architecture for Embedded Processors,” IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, vol. 38, no. 3, pp. 466–479, MARCH 2019.

[68] L. Sterpone, “A New Reconfigurable Clock-Gating Technique for Low Power SRAM-based FPGAs,” 2011 Design, Automation & Test in Europe DOI: 10.1109/DATE18608.2011, 14-18 March 20112011.

[69] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, “Green cloud computing: Balancing energy in processing, storage, and transport,” *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, 2011, doi: 10.1109/JPROC.2010.2060451.

[70] T. T. O. Kwok and Y. K. Kwok, “Computation and energy efficient image processing in wireless sensor networks based on reconfigurable computing,” *Proc. Int. Conf. Parallel Process. Work.*, pp. 43–50, 2006, doi: 10.1109/ICPPW.2006.30.

[71] V. Raghunathan, S. Ganeriwal, M. Srivastava, and C. Schurgers, “Energy Efficient Wireless Packet Scheduling and Fair Queuing,” *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 1, pp. 3–23, 2004, doi: 10.1145/972627.972629.

[72] A. Rogacs, “Enabling Right-Provisioned Microprocessor Architectures For The Internet Of Things,” Proceedings of the ASME 2015 International Mechanical Engineering Congress and Exposition IMECE2015 November 13-19, 2015, Houston, Texas pp. 1–10.

[73] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, 2014, doi: 10.1109/SURV.2013.042313.00197.

[74] S. Shukla and K. C. Ray, “A Low-Overhead Reconfigurable RISC-V Quad-Core Processor Architecture for Fault-Tolerant Applications,” *IEEE Access*, vol. 10, pp. 44136–44146, 2022, doi: 10.1109/ACCESS.2022.3169495.

[75] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[76] Harald Sundmaeker Patrick Guillemin Peter Friess Sylvie Woelfflé, "Vision and Challenges for Realising the Internet of Things", CERP-IoT - Cluster of European Research Projects on Internet of Things, March 2010. doi: 10.2759/26127.

[77] M. Urbina, T. Acosta, J. Lazaro, A. Astarloa, and U. Bidarte, “Smart Sensor: SoC Architecture for the Industrial Internet of Things,” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6567–6577, 2019, doi: 10.1109/JIOT.2019.2908264.

[78] D. Folegnani and A. González, “Energy-effective issue logic,” *Conf. Proc. - Annu. Int. Symp. Comput. Archit. ISCA*, vol. 00, no. C, pp. 230–239, 2001, doi: 10.1145/379240.379266.

[79] Y. Wu, Y. J. Chen, T. S. Chen, Q. Guo, and L. Zhang, “An elastic architecture adaptable to various application scenarios,” *J. Comput. Sci. Technol.*, vol. 29, no. 2, pp. 227–238, 2014, doi: 10.1007/s11390-014-1425-x.

[80] C. Zhang, F. Vahid, and W. Najjar, “A highly configurable cache architecture for embedded systems,” *Conf. Proc. - Annu. Int. Symp. Comput. Archit. ISCA*, pp. 136–146, 2003, doi: 10.1145/859634.859635.

[81] H. Hajimiri and P. Mishra, “Intra-task dynamic cache reconfiguration,” *Proc. IEEE Int. Conf. VLSI Des.*, pp. 430–435, 2012, doi: 10.1109/VLSID.2012.109.

[82] S. Mach, F. Schuiki, F. Zaruba, and L. Benini, “FPnew: An Open-Source Multiformat Floating-Point Unit Architecture for Energy-Proportional Transprecision Computing,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 29, no. 4, pp. 774–787, 2021, doi: 10.1109/TVLSI.2020.3044752.

[83] T. Jiang *et al.*, “Understanding the Behavior of In-Memory Computing Workloads,” *2014 IEEE Int. Symp. Workload Charact.*, pp. 22–30, doi: 10.1109/IISWC.2014.6983036.

[84] K. Monga, S. Behera, N. Chaturvedi, and S. Gurunarayanan, “Design of In-Memory Computing Enabled SRAM Macro,” *INDICON 2022 - 2022 IEEE 19th India Counc. Int. Conf.*, pp. 1–4, 2022, doi: 10.1109/INDICON56171.2022.10039958.

[85] A. Appukuttan, E. Thomas, H. R. Nair, S. Hemanth, K. J. Dhanaraj, and M. A. Azeez, “In-Memory Computing Based Hardware Accelerator Module for Deep Neural Networks,” *INDICON 2022 - 2022 IEEE 19th India Counc. Int. Conf.*, pp. 1–6, 2022, doi: 10.1109/INDICON56171.2022.10040126.

[86] M. Kang, S. K. Gonugondla, M. S. Keel, and N. R. Shanbhag, “An energy-efficient memory-based high-throughput VLSI architecture for convolutional networks,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2015-Augus, pp. 1037–1041, 2015, doi: 10.1109/ICASSP.2015.7178127.

[87] S. Yu *et al.*, “A Heterogeneous Microprocessor Based on All-Digital Compute-in-Memory for End-to-End AIoT Inference,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. PP, no. X, p. 1, 2023, doi: 10.1109/TCSII.2023.3249245.

[88] R. Kumar, K. Farkas, N. P. Jouppi, and P. Ranganathan, “Processor power reduction via single-isa heterogeneous multi-core architectures,” *IEEE Comput. Archit. Lett.*, vol. 2, no. 1, p. 2, 2003, doi: 10.1109/L-CA.2003.6.

[89] M. Brandalero, L. Carro, A. Carlos, S. Beck, M. Shafique, and S. Member, “Multi-Target Adaptive Reconfigurable Acceleration for Low-Power IoT Processing,” IEEE Transactions On Computers, vol. 70, no. 1, pp. 83–98, 2021.

[90] Z. Wang, Y. Liu, and D. Zhang, “An Energy-Efficient Heterogeneous Dual-Core Processor for Internet of Things,” *2015 IEEE Int. Symp. Circuits Syst.*, pp. 2301–2304, 2015, doi: 10.1109/ISCAS.2015.7169143.

[91] M. Soc, I. Devices, Y. Yang, and W. Diao, “An Energy-efficient Frame-based Task Scheduling Algorithm for Heterogeneous,”2020 International Wireless Communications and Mobile Computing (IWCMC) DOI: 10.1109/IWCMC48107.2020, pp. 1404–1409, 15-19 June 2020.

[92] Y. Liu *et al.*, “A 65nm ReRAM-enabled nonvolatile processor with 6× reduction in restore time and 4× higher clock frequency using adaptive data retention and self-write-termination nonvolatile logic,” *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*,vol.59, pp.84–86, 2016,doi: 10.1109/ISSCC.2016.7417918.

[93] F. M. Ghannouchi, “Power amplifier and transmitter architectures for software defined radio systems,” *IEEE Circuits Syst. Mag.*, vol. 10, no. 4, pp. 56–63, 2010, doi: 10.1109/MCAS.2010.938639.

[94] R. Akeela and B. Dezfouli, “Software-defined Radios: Architecture, state-of-the-art, and challenges,” *Comput. Commun.*, vol. 128, no. June, pp. 106–125, 2018, doi: 10.1016/j.comcom.2018.07.012.

[95] Y. Chen, S. Lu, H. S. Kim, D. Blaauw, R. G. Dreslinski, and T. Mudge, “A low power software-defined-radio baseband processor for the Internet of Things,” *Proc. - Int. Symp. High-Performance Comput. Archit.*, vol. 2016-April, pp. 40–51, 2016, doi: 10.1109/HPCA.2016.7446052.

[96] D. C. Dinis *et al.*, “A Real-Time Architecture for Agile and FPGA-Based Concurrent Triple-Band All-Digital RF Transmission,” *IEEE Trans. Microw. Theory Tech.*, vol. 66, no. 11, pp. 4955–4966, 2018, doi: 10.1109/TMTT.2018.2860972.

[97] S. Balasubramanian *et al.*, “Systematic analysis of interleaved digital-to-analog converters,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 58, no. 12, pp. 882–886, 2011, doi: 10.1109/TCSII.2011.2172526.

[98] S. Y. Yang, J. Yang, L. Y. Huang, J. L. Bai, and X. Y. Zhang, “A Dual-Band RF All-Digital Transmitter Based on MPWM Encoding,” *IEEE Trans. Microw. Theory Tech.*, vol. 70, no. 3, pp. 1745–1756, 2022, doi: 10.1109/TMTT.2021.3135858.

[99] D. C. Dinis, R. F. Cordeiro, A. S. R. Oliveira, J. Vieira, and T. O. Silva, “A Fully Parallel Architecture for Designing Frequency-Agile and Real-Time Reconfigurable FPGA-Based RF Digital Transmitters,” *IEEE Trans. Microw. Theory Tech.*, vol. 66, no. 3, pp. 1489–1499, 2018, doi: 10.1109/TMTT.2017.2764451.

[100] A. Transmitter, S. S. Pereira, and G. S. Member, “Scalable Resource Optimized LUT-Based All-Digital Transmitter,” IEEE Transactions On Circuits And Systems-I: Regular Papers, pp.1-9,2023,doi:10.1109/TCSI.2023.3274432.

[101] H. B. Amor, C. Bernier, and Z. Prikryl, “A RISC-V ISA Extension for Ultra-Low Power IoT Wireless Signal Processing,” *IEEE Trans. Comput.*, vol. 71, no. 4, pp. 766–778, 2022, doi: 10.1109/TC.2021.3063027.

[102] M. Jankowski, D. Gunduz, and K. Mikolajczyk, “Deep Joint Source-Channel Coding for Wireless Image Retrieval,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2020-May, pp. 5070–5074, 2020, doi: 10.1109/ICASSP40776.2020.9054078.

[103] X. Zhao, V. Sadhu, and D. Pompili, “Analog Signal Compression and Multiplexing Techniques for Healthcare Internet of Things,” *Proc. - 14th IEEE Int. Conf. Mob. Ad Hoc Sens. Syst. MASS 2017*, pp. 398–406, 2017, doi: 10.1109/MASS.2017.62.

[104] J. S. Lee, P. Choi, and D. K. Kim, “Lightweight and Low-Latency AES Accelerator Using Shared SRAM,” *IEEE Access*, vol. 10, pp. 30457–30464, 2022, doi: 10.1109/ACCESS.2022.3156291.

[105] K. L. Tsai, F. Y. Leu, I. You, S. W. Chang, S. J. Hu, and H. Park, “Low-Power AES Data Encryption Architecture for a LoRaWAN,” *IEEE Access*, vol. 7, pp. 146348–146357, 2019, doi: 10.1109/ACCESS.2019.2941972.

[106] J. C. See, K. M. Mok, W. K. Lee, and H. G. Goh, “RISC32-E: Field programmable gate array based sensor node with queue system to support fast encryption in Industrial Internet of Things

applications," *Int. J. Circuit Theory Appl.*, vol. 48, no. 8, pp. 1209–1226, 2020, doi: 10.1002/cta.2797.

[107] T. Gebremichael *et al.*, "Security and Privacy in the Industrial Internet of Things: Current Standards and Future Challenges," *IEEE Access*, vol. 8, pp. 152351–152366, 2020, doi: 10.1109/ACCESS.2020.3016937.

[108] G. A. Fink, Di. V. Zarzhitsky, T. E. Carroll, and E. D. Farquhar, "Security and privacy grand challenges for the Internet of Things," *2015 Int. Conf. Collab. Technol. Syst. CTS 2015*, pp. 27–34, 2015, doi: 10.1109/CTS.2015.7210391.

[109] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Ravi, "Security as a new dimension in embedded system design," *Proc. - Des. Autom. Conf.*, pp. 753–760, 2004, doi: 10.1145/996566.996771.

[110] J. Gopika Rajan and R. S. Ganesh, "Hardware Based Data Security Techniques in IOT: A Review," *3rd Int. Conf. Smart Electron. Commun. ICOSEC 2022 - Proc.*, no. Icosec, pp. 408–413, 2022, doi: 10.1109/ICOSEC54921.2022.9952021.

[111] J. Crenne, R. Vaslin, G. Gogniat, J. P. Diguet, R. Tessier, and D. Unnikrishnan, "Configurable memory security in embedded systems," *Trans. Embed. Comput. Syst.*, vol. 12, no. 3, 2013, doi: 10.1145/2442116.2442121.

[112] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 2014-Novem, no. November, pp. 31–40, 2014, doi: 10.1145/2663474.2663479.

[113] V. Izosimov, P. Pop, P. Eles, and Z. Peng, "Design optimization of time- and cost-constrained fault-tolerant distributed embedded systems," *Proc. -Design, Autom. Test Eur. DATE '05*, vol. II, pp. 864–869, 2005, doi: 10.1109/DATE.2005.116.

[114] U. Afzaal and J. Lee, "Low-cost Hardware Redundancy for Fault-mitigation in Power-constrained IoT Systems," 2020 International Conference on Information and Communication Technology Convergence (ICTC), 21-23 Oct. 2020 pp. 60–62, 2020, doi: 10.1109/ICTC49870.2020.9289420.

[115] S. Ainsworth and T. M. Jones, "Parallel Error Detection Using Heterogeneous Cores," *2018 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks*, pp. 338–349, 2018, doi: 10.1109/DSN.2018.00044.

[116] T. Mendez, "Performance Evaluation of Fault-Tolerant Approximate Adder," *2022 6th Int. Conf. Devices, Circuits Syst.*, no. April, pp. 1–5, 2022, doi: 10.1109/ICDCS54290.2022.9780792.

[117] H. Moon, J. Cho, and D. Park, "Reconfigurable Fault-Safe Processor Platform Based on RISC-V for Large-Scaled IoT-Driven Applications," pp. 627–632, 2019, doi: 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00119.

[118] Y. Ko, "Survey of Software-Implemented Soft Error Protection," *Electron.*, vol. 11, no. 3, 2022, doi: 10.3390/electronics11030456.

[119] O. S. Unsal, I. Koren, and C. M. Krishna, "Towards Energy-Aware Software-Based Fault Tolerance in Real-Time Systems", Proceedings of the International Symposium on Low Power Electronics and Design, DOI: 10.1109/LPE.2002, 14-14 Aug. 2002, vol. 1, pp. 124–129,2002.