Day–Night architecture: Development of an ultra-low power RISC-V processor for wearable anomaly detection ☆,☆☆

Eunjin Choi [a],[1], Jina Park [a],[1], Kyeongwon Lee [a],[1], Jae-Jin Lee [b], Kyuseung Han [b], Woojoo Lee [a],*

[a] *Chung-Ang University, Seoul, 06974, Republic of Korea*
[b] *Electronics and Telecommunications Research Institute, Daejeon, 34129, Republic of Korea*

ARTICLE INFO

ABSTRACT

In healthcare, anomaly detection has emerged as a central application. This study presents an ultra-low power processor tailored for wearable devices dedicated to anomaly detection. Introducing a unique *Day–Night* architecture, the processor is bifurcated into two distinct segments: The *Day* segment and the *Night* segment, both of which function autonomously. The Day segment, catering to generic wearable applications, is designed to remain largely inactive, awakening only for specific tasks. This approach leads to considerable power savings by incorporating the Main-CPU and system interconnect, both major power consumers. Conversely, the Night segment is dedicated to real-time anomaly detection using sensor data analytics. It comprises a Sub-CPU and a minimal set of IPs, operating continuously but with minimized power consumption. To further enhance this architecture, the paper presents an ultra-lightweight RISC-V core, *All-Night* core, specialized for anomaly detection applications, replacing the traditional Sub-CPU. To validate the Day–Night architecture, we developed a prototype processor and implemented it on an FPGA board. An anomaly detection application, optimized for this prototype, was also developed to showcase its functional prowess. Finally, when we synthesized the processor prototype using 45 nm process technology, it affirmed our assertion of achieving an energy reduction of up to 57%.

1. Introduction

As technological advancements continue, pervasive integration of systems harness mobile, wearable, and implant devices to collect diverse data. This data is subsequently analyzed using big data techniques, culminating in actionable insights derived through AI algorithms. Healthcare stands as a testament to this trend, illustrating how deeply such systems have penetrated our daily lives. A plethora of contemporary wearable devices are now embedded with sensors that capture metrics like heart rate, blood pressure, body temperature, and oxygen saturation. As a result, there is an upsurge in services offering remote monitoring, fitness management, chronic disease detection, and support for the elderly [1–4].

In the realm of healthcare, the predominant application of data collection is continuous user monitoring, aiming to identify anomalies and promptly intervene upon detection. Such proactive approaches significantly enhance patient care efficiency [5–12]. Consequently, the evolution of wearable devices now emphasizes increased capabilities for anomaly detection. This development trajectory prioritizes the gathering of a broader range of physiological data, the enhancement of detection accuracy, and the user convenience of the devices. Crucially, achieving these objectives necessitates low power consumption. Wearable devices must sustain always-on data collection from various sensors, ensure real-time abnormality detection, and minimize the frequency of recharging to provide an uninterrupted user experience.

Low-power technologies such as dynamic voltage and frequency scaling [13,14], dynamic power management (DPM) [15–21], energy efficient multi-core architectures [22–25], and application specific hardware accelerators [26–29] have been actively introduced into processors for wearable devices. Nevertheless, as the types of applications

(apps) running on wearable devices are explosively diversifying, and at the same time as the number of sensors installed and the amount of data to be processed increases even for anomaly detection, wearable devices still long for a low-power processor.

In this study, our primary objective is the development of ultra-low power (ULP) processors tailored for anomaly detection in wearable devices. We initiated our approach by designing a baseline processor that integrates contemporary wearable processor technology. This architecture encompasses two heterogeneous CPU units: the high-performance, power-intensive Main-CPU, and the low-power, low-performance Sub-CPU dedicated to data collection from wearable sensors and anomaly detection. The Main-CPU transitions to a standby mode during periods of inactivity. For this standby mode, dynamic power management (DPM) technologies such as power gating (PG) and clock gating (CG) are available. While PG is recognized as the most effective technique for DPM, it might not provide significant benefits for smaller Systems on Chips (SoCs) [15–17]. Conversely, CG, though it only cuts off the dynamic power while the static power remains, enables fast activation and deactivation within just 1 clock cycle and requires no separate power switches, resulting in very low control overhead [18–21]. As CG has been widely implemented in small SoCs like our target processor, we also apply CG for the standby mode in this paper. Meanwhile, the Sub-CPU constantly runs the anomaly detection application, and upon detecting anomalies, triggers the Main-CPU to alert the user and liaise with external medical entities. The Main-CPU utilizes the RISC-V Rocket core [30], whereas the Sub-CPU integrates the RISC-V ORCA core [31].

Subsequent power consumption analysis of the baseline processor in standby mode highlighted a significant oversight in prevailing wearable processor designs. Power consumption of the system interconnect constituted a substantial 49.3% of the total standby mode power draw. Given that contemporary architectures necessitate the continual operation of the system interconnect in conjunction with the Sub-CPU, relegating it to standby mode remains unfeasible.

Motivated by this challenge, we came up with a new processor architecture that enables the system interconnect to transition to standby mode, allowing the Sub-CPU to operate autonomously. Moreover, despite employing the least power-intensive core available in the public domain RISC-V processors for the Sub-CPU, our evaluations indicated it was still over-specified relative to anomaly detection apps' demands. Consequently, we engineered a bespoke ultra-lightweight RISC-V processor tailored for these applications. Validation was carried out by prototyping with both the Digilent FPGA and 45 nm process technology. Additionally, a dedicated anomaly detection application was developed for the prototype processor. Key contributions of this study are:

- Proposal and elaboration of a processor architecture tailored for wearable devices, termed the *Day–Night* architecture. This dual-segment design comprises the *Day* part, which primarily encompasses the Main-CPU and system interconnect, and the *Night* part that includes the Sub-CPU and peripherals, each operating autonomously. The Day part can transition to standby mode, while the Night part remains active.
- Introduction of the *All-Night* core, an ultra-lightweight RISC-V core for the Sub-CPU. Despite its compact design, executing a minimal instruction set from RV32I, it features a 3-stage pipeline to ensure minimal performance degradation.
- Development of a prototype processor based on the Day–Night architecture, coupled with the development of an anomaly detection application for it. This application continually monitors user behavior, body temperature, and heart rate (photoplethysmography, PPG) on the All-Night core. Upon anomaly detection, it activates the Main-CPU for external alert generation and rescue initiation.
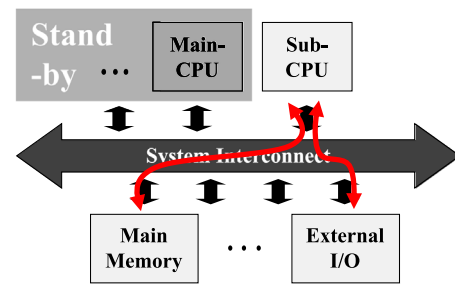

Fig. 1. Problem of the conventional processor architecture when accessing sensors.

The structure of this paper is outlined as follows: Section 2 presents the Day–Night architecture. Specifically, Section 2.1 introduces the structure of this processor architecture, while Section 2.2 delves into the detailed development of the All-Night core. In Section 3, we detail the development of a processor prototype employing the proposed Day–Night architecture and the anomaly detection app running on the prototype. Specifically, Section 3.1 offers insights into the anomaly detection app developed for verifying and evaluating the processor. Section 3.2 presents a thorough discussion on the design of a processor utilizing the proposed techniques, juxtaposed with a baseline processor for comparative assessment. Section 4 validates the efficacy of our technology, including performance evaluations of the FPGA prototype in Section 4.1 and its synthesis with process technology in Section 4.2. Finally, Section 5 concludes the paper.

2. Day–Night architecture

2.1. Structure of the proposed processor

A wearable anomaly detection processor should offer sufficient computational capabilities for general wearable apps. Simultaneously, it must ensure prolonged operation in anomaly detection, utilizing sensor data with minimal power drain. Traditionally, this balance is achieved through a heterogeneous architecture, complemented by the DPM technique. Specifically, the high-performance, high-power Main-CPU manages general apps. Given that these apps run sporadically based on user interaction, the Main-CPU transitions to standby mode during idle periods. In contrast, the Sub-CPU, tailored for low power and performance, is dedicated exclusively to anomaly detection, running persistently without entering standby.

Fig. 1 depicts a processor architecture based on this traditional approach. Within this structure, the system interconnect facilitates communication among the CPUs, memory, and other IPs. Consequently, even when the Sub-CPU executes the anomaly detection app, it still relies on the system interconnect. This means that whether reading sensing data from external I/O or accessing memory, the Sub-CPU's interactions always pass through the system interconnect. This architecture demands the continuous operation of the system interconnect, leading to unavoidable power costs. Alarmingly, in scenarios where only the anomaly detection app is active and all general app-related IPs are in standby mode, power consumption of the system interconnect dominates, accounting for a considerable fraction (e.g., 55% in the baseline processor). Given that wearable devices predominantly run the anomaly detection app, optimizing this aspect could yield significant power savings.

To address this challenge, we introduce the Day–Night processor architecture. This design divides the processor into two distinct segments: the Day segment, dedicated to general apps, and the Night segment, focused solely on anomaly detection. Each segment functions autonomously, ensuring efficient task segregation. As illustrated in Fig. 2, the system interconnect, paired with the Main-CPU, is encompassed within the Day segment. This arrangement allows the system
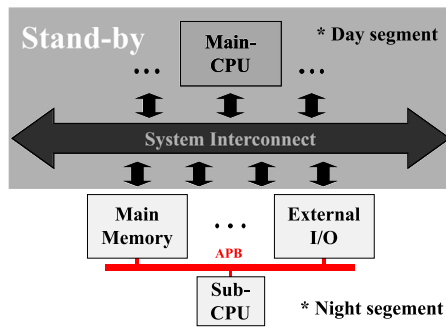
Fig. 2. Operation in standby mode of the proposed processor architecture.

Available (implemented)	Unavailable
LUI,	AUIPC, BNE,
JAL,	BLT, BGE,
JALR,	BLTU, BGEU,
BEQ,	LB, LH,
LW,	LBU, LHU,
SW,	SB, SH,
ADD	SLTI, SLTIU,
ADDI	XORI, ORI,
SUB,	ANDI, SLLI,
SLL,	SRAI, SLLI,
SLT,	SLTU, SRI,
SRA,	FENCE, FENCE.I,
XOR,	ECALL, EBREAK,
OR,	CSRRW, CSRRS,
AND,	CSRRC, CSRRWI,
(MUL in RV32M)	CSFFSI, CSRRCI

Table 2
Power consumption percentage by ALU in RISC-V cores.

	ALU power (µW)	ALU proportion (%)	Multiplier power (µW)	Multiplier proportion (%)
picoRV32	333.60	18.85	169.46	50.80
e203	985.14	21.14	257.42	26.12
mriscvcore	550.78	28.10	245.21	44.52
tinyriscv	203.03	9.81	84.08	41.42
All-Night	113.5	11.50	–	–

interconnect to transition to an inactive state when only the Night segment is operational. For the Night segment to execute the anomaly detection app, the Sub-CPU requires access to the memory housing the CPU code and data, as well as to peripheral devices capable of reading sensor values.

While the most intuitive solution might seem to be the addition of dedicated memory and peripheral modules for the Night segment, such an approach would inevitably result in increased design complexities, a larger footprint, and heightened power consumption. Given these drawbacks, we opted against this method. Instead, our strategy allows both the Day and Night segments to leverage the existing main memory and peripheral modules, while ensuring the Night segment has unhindered access without navigating through the system interconnect.

To achieve this, we incorporated a dual-port memory controller within the existing main memory. Concurrently, we affixed a distinct port to the peripheral device, ensuring its connectivity with the Sub-CPU. Recognizing that the Sub-CPU does not necessitate high-performance interactions with the main memory or external peripherals, we have configured all these entities with the Advanced Peripheral Bus (APB) interface, utilizing the APB bus as the primary communication framework. Additionally, we integrated arbiters that sequentially allocate priority, effectively averting potential data collisions between the Day and Night segments. A comprehensive breakdown of the processor designed around this architecture, complemented by an intricate implementation schematic, is delineated in Section 3.2 (see Fig. 9 for reference).

2.2. All-Night core

Anomaly detection applications continuously monitor sensor data. Upon detecting any deviations from the norm, these apps trigger an alarm, alerting both the user and external systems. While the Sub-CPU shoulders the responsibility of sensor data reading and anomaly identification, the Main-CPU takes charge of relaying external signals. The Sub-CPU's task is notably lightweight, negating the need for high-performance computing. Instead, a constant operational state is vital for the Sub-CPU, necessitating energy-efficient performance. Thus, we have engineered the All-Night core, an ultra-lightweight core, encompassing only the essential computational capabilities required by anomaly detection apps.

In designing the All-Night core, complexities like interrupts and pipelines, typically associated with CPU design, are avoided. This is attributable to the core's lack of direct interaction with interrupts, resulting in simplified control logic and negating the necessity for intricate control and state register (CSR) components. Given that the All-Night core is not performance-intensive, a pipeline consisting of many stages is unnecessary.

Interestingly, when compared to general CPU designs, the All-Night core offers significant instruction reduction opportunities. A designer, for instance, might opt to only incorporate instructions relevant to

the target application, or might decide to downsize the area at the expense of increased latency for components like multipliers or shifters. This study focuses on establishing a versatile instruction set to aid in application maintenance. We derived essential instructions from RISC-V32I [32], with the outcomes detailed in Table 1. Furthermore, the core's arithmetic capability has been augmented with the MUL instruction from the RV32M standard extension. Consequently, despite the minimized instructions which might necessitate some assembly coding, the core is streamlined and can execute a mere 16 instructions.

Fig. 3 delineates our developed All-Night core architecture. This core sports a rudimentary three-stage pipeline structure: FETCH, DECODE, and EXECUTE. Notably, the core's ALU handles only six operations: ADD (SUBSTRACT), SHIFT, AND, OR, XOR, and MULT (multiplication). To effectively run the anomaly detection app and maintain a compact core size, AND, OR, and XOR are straightforwardly executed as bitwise operators. In contrast, ADD, SHIFT and MULT are manifested as distinct 32-bit adders, 1-bit shifters, and 32-bit multipliers, respectively.

In our pursuit to optimize the ALU within the All-Night core, we recognized that typically, ALUs are significant power consumers in cores, with multipliers being especially power-intensive. By analyzing four renowned, open-source embedded RISC-V cores - picoRV32 [33], e203 [34], mriscvcore [35], and Tinyriscv [36], we discerned the power consumption metrics of ALUs and the proportion attributed to multipliers. Table 2 encapsulates our findings. Our approach to mitigate power consumption of the All-Night core was to utilize lightweight implementations of a 32-bit adder and a 1-bit shifter, named *adder_32bit* and *shifter_1bit*, in lieu of implementing a 32-bit multiplier. To this end, we employed the algorithm detailed in Fig. 4. This algorithm computes the product of two inputs, *rs1_data* and *rs2_data*. The algorithm first involves performing an AND operation with *rs1_data* and each of *rs2_data*'s bits, starting from the least significant bit (LSB) to the most significant bit (MSB), for 32 iterations. The calculation then considers the weight of the current i value and shifts *rs1_data* to the left by i, employing the *shifter_1bit* module. After 32 repetitions, the aggregated AND operation results yield the final output. As a result, as reported
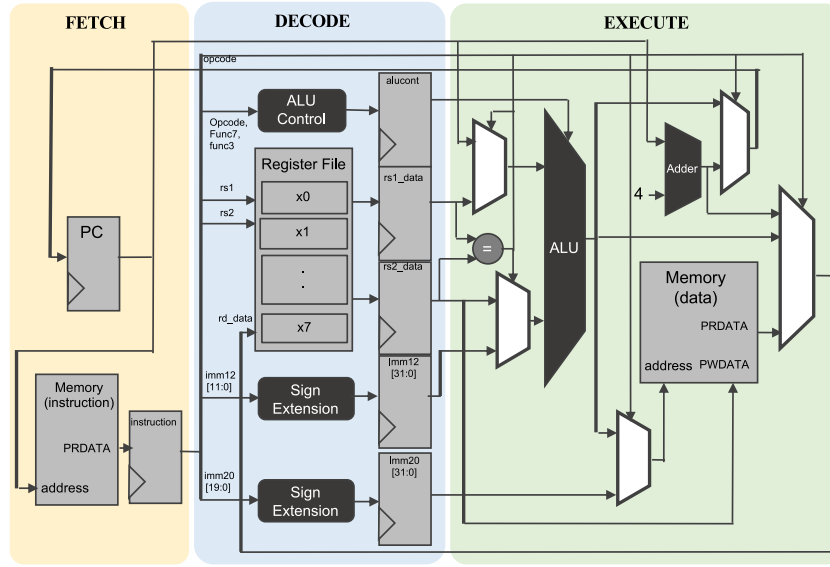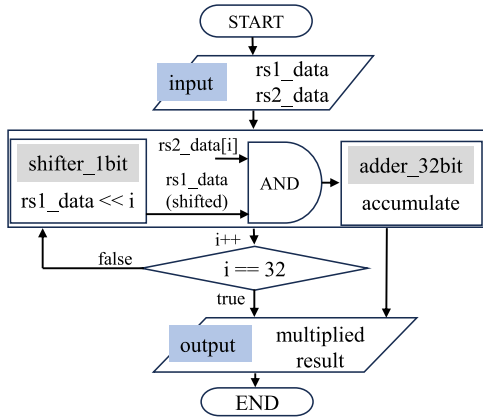
Fig. 3. Architecture of the All-Night core.

Fig. 4. Algorithm to replace multiplication operations using *adder_32bit* and *shifter_1bit*.

Fig. 5. Comparison of FPGA resource consumption between the four compact RISC-V cores and the All-Night core.
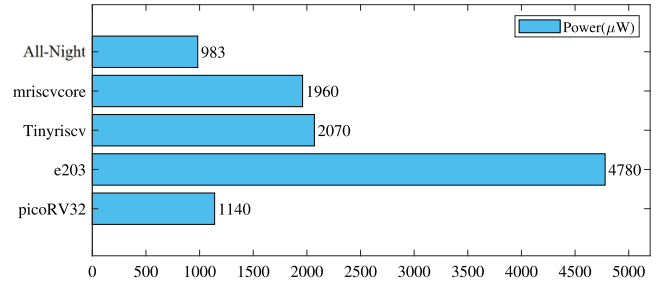

Fig. 6. Comparison of power consumption based on 45 nm process technology synthesis between the four compact RISC-V cores and the All-Night core.
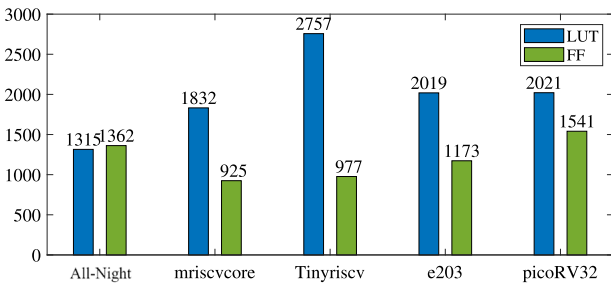
in Table 2, the power consumption of the ALU in the All-Night core is significantly reduced compared to other cores.

The All-Night core has been designed with the primary goal of being ultra-lightweight. Its efficacy in achieving this can be ascertained through a comparison with the four other RISC-V cores. Firstly, Fig. 5 depicts the resource consumption of the All-Night core versus the four compact RISC-V cores when synthesized on an FPGA. The LUT (Look-Up Table) utilization of the All-Night core was found to be 71.78% relative to mriscvcore, 47.7% relative to Tinyriscv, 65.18%

relative to e203, and 65.01% relative to picoRV32. Next, Fig. 6 presents power consumption measurements for each core synthesized using the Nangate 45 nm technology library [37]. As indicated in the figure, the All-Night core's power consumption stands at approximately 50.15% of that of mriscvcore, 47.48% of Tinyriscv, 20.56% of e203, and 86.22% of picoRV32. From these comparisons, it is evident that the All-Night core demonstrates significantly lower power consumption compared to existing compact cores.

Meanwhile, a salient feature of the All-Night core architecture is its ability to share memory with the Main-CPU, obviating the need for a distinct system memory map or compilation process. As a result, developers can seamlessly integrate operations for the All-Night core by appending specialized functions to the conventional wearable app code intended for the Main-CPU. This integrated approach is embodied in a function referred to as *Night_func*.

However, this design presents a challenge: the standard CPU boot code is incompatible with booting the All-Night core. To address this, we developed a dedicated booting mechanism, as depicted in Fig. 7. In this schema, *Night_addr* represents the memory location where *Night_func* is compiled. After the Main-CPU concludes its boot sequence, it registers both the *Night_addr* for the All-Night core and an enabling signal, *enable_Night*. Upon entering its main operation, the Main-CPU continually verifies the state of *enable_Night*. When set to 1, control is transferred to *Night_addr* to execute *Night_func*. This mechanism necessitated two supplementary registers for *Night_addr* and *enable_Night*. We have pragmatically addressed this by situating them on the external I/O interface. This streamlined approach facilitates memory and peripheral
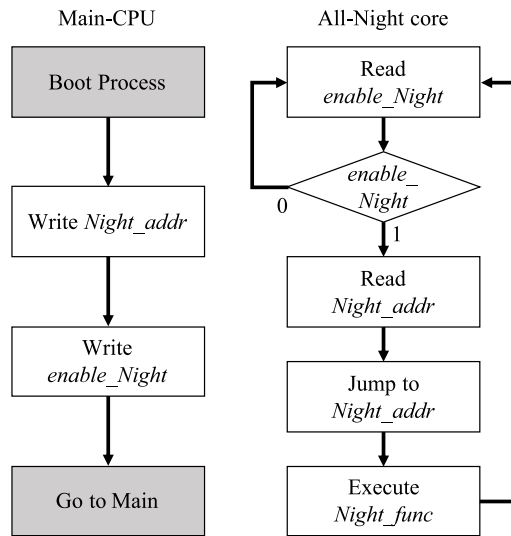
Fig. 7. Booting mechanism of the All-Night core.

circuit sharing, allowing for dual-mode operation: *Day-mode* for routine tasks and *Night-mode* exclusively for anomaly detection.

3. Implementation

3.1. Wearable anomaly detection application

As anomaly detection becomes increasingly important, a variety of algorithms have been developed for this purpose. Traditional threshold-based anomaly detection algorithms have existed [5,7,8], which detect abnormal states based on whether sensor signals exceed or fall below predefined thresholds. Recently, research and development have been active in learning-based algorithms that combine and learn from signal features for more accurate or personalized anomaly detection [9–12].

The ultra-low power RISC-V processor we propose can be utilized for both traditional and modern algorithms. More specifically, for traditional methods, low-complexity models that detect anomalies from sensor values can be directly executed on the Sub-CPU, and if an anomaly is detected, the Main-CPU can be activated to handle the abnormal situation. For modern machine learning or neural network-based learning models, simple pre-processing can be conducted on the Sub-CPU before the Main-CPU performs complex algorithms. If the complexity of the algorithm exceeds the capabilities of the Main-CPU, it might be necessary to adopt an edge computing approach, where the main CPU sends data to a high-performance server cluster for processing.

The processor architecture proposed in this paper aims for power savings during phases where only the Sub-CPU is activated—either performing simple anomaly detection algorithms of traditional approaches or pre-processing stages of complex models. Therefore, we plan to evaluate the performance of the proposed processor architecture by running simple anomaly detection algorithms of traditional methods and the pre-processing part of learning-based anomaly detection algorithms on the Sub-CPU of the proposed processor. Furthermore, to demonstrate seamless operation throughout the entire process from the Sub-CPU to the Main-CPU, we have developed an application that performs anomaly detection on the Sub-CPU and wakes the Main-CPU for post-processing if an anomaly is detected. This self-developed anomaly detection application distinguishes itself from existing applications used for performance validation by utilizing a more extensive array of sensors. Additionally, the post-processing tasks conducted by the Main-CPU are developed to be simple enough to operate independently without the need for connecting to an external server. This

development approach ensures that the application serves its primary purpose of validating the processor's functionality and demonstrating its low-power superiority.

The detailed operational mechanism of the developed anomaly detection test application is depicted in Fig. 8. As illustrated, the application gathers data at 100 ms intervals from the temperature sensor, PPG sensor, and accelerometer to obtain the user's body temperature D_{temp}, heart rate D_{PPG}, and behavior D_{Acc}, respectively. Standard values for the user's body temperature, heart rate, and behavior are denoted as α, β, and γ. Acceptable boundaries for these parameters are stored in the main memory as α_{min}, α_{max}, β_{min}, β_{max}, γ_{min}, and γ_{max}.

Initially, α_{min} and α_{max} are set based on the average human body temperature range, 34 °C to 38 °C, which can be adjusted as required. A deviation from this range persisting for 30 s or more, equivalent to 300 intervals, is interpreted as hypothermia or high fever, prompting an interruption to the Main-CPU.

Subsequently, β_{min} and β_{max} are determined to monitor any heart rate abnormalities. The PPG sensor conveys heart rate values via UART. A signal initiated by 254 (8'b11111110) and terminated by 255 (8'b11111111) is recognized as a synchronization signal, with the interim value considered as the current heart rate. By storing the current heart rate in a list, its average over predefined intervals (1 min, 8 min, 20 min, and a day) is computed. If the current heart rate deviates from these four averages by more than 15 for 20 s or 200 intervals, the user is assessed to be in an anomalous state, leading to an interrupt of the Main-CPU.

The third sensor is an accelerometer, which calculates both acceleration and inclination to determine if the user has experienced a fall. The analysis employs three-dimensional raw data, i_X, i_Y, and i_Z, from the accelerometer. A zero reading for i_Z suggests a potential anomaly in inclination, indicating that the user is not upright. This triggers a subsequent evaluation of acceleration (acc), which is computed through the following formula.

$$acc = i_X \times (i_X \gg 7) + i_Y \times (i_Y \gg 7) + i_Z \times (i_Z \gg 7)$$

The lower threshold, γ_{min}, is defined as 0.5 times the 20-s average, while the upper threshold, γ_{max}, is set at 1.5 times the 20-s average. If the acceleration deviates from these boundaries several times within a short period, the user's current motion is identified as anomalous. Should both inclination and acceleration register anomalies, the system then shifts its focus to detecting irregularities in heart rate. This is predicated on the rationale that a user is only deemed in a critical situation from a fall, warranting immediate attention, when both acceleration and inclination abnormalities are coupled with heart rate deviations from the standard range. If the heart rate too exhibits abnormalities, the system recognizes an emergency, subsequently generating a signal to rouse the Main-CPU.

Upon receiving an interrupt from the All-Night core, the Main-CPU transitions from standby mode to active mode if it was in the former state. If the Main-CPU was engaged in another operation, it halts the current task and takes appropriate measures in response to the detected anomaly. For instance, the Main-CPU can execute a more sophisticated anomaly detection program to further analyze the suspected abnormal conditions identified by the All-Night core. Actions can range from notifying the user directly to sending detailed status reports to guardians or medical institutions, potentially accompanied by an emergency signal. However, the scope of this paper does not encompass the development of intricate anomaly detection programs. As such, many details regarding the operations executed by the Main-CPU are omitted. For demonstration purposes, we have implemented functionalities to display alarms on the OLED display connected to the processor and to send an emergency rescue alert to a server via Bluetooth.

When the Main-CPU completes the emergency alert, as the main core reverts to its prior state—be it standby or its previous operation—the All-Night core is informed through the *Night_enable* variable depicted in Fig. 7. This is facilitated by the All-Night core consistently
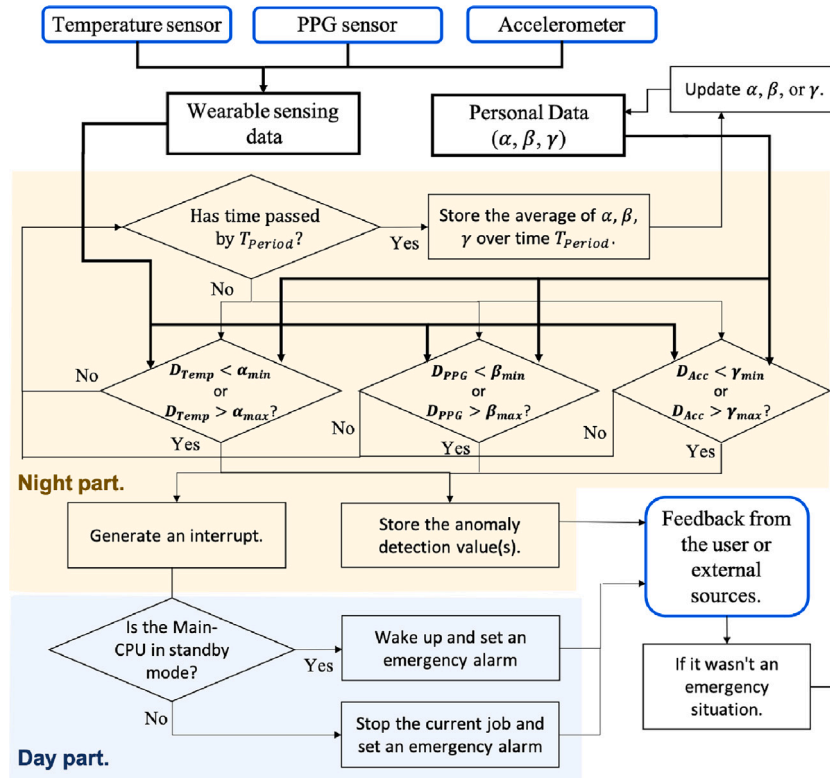
Fig. 8. Operating flow chart of the developed healthcare app.

monitoring the *Night_enable* signal during the FETCH stage. Subsequent to recognizing this signal, it resumes operation from the address previously stored in the shared main memory by the main core prior to activating the *Night_enable*, or the anomaly detection mode. This process corresponds to the *Night part* shown in Fig. 8.

In addition, the app we developed is equipped with a feature allowing customized steady-state updates. This enables adjustment of the user's standard range values based on feedback—whether it is from the user themselves or from external medical agencies, particularly when the flagged emergency falls within an acceptable range. As illustrated in Fig. 8, the stored values of α, β, and γ can be fine-tuned based on this feedback or modified through data accumulated over a designated time span, T_{period}. Any adjustments to the user's standard data by the main core can be immediately utilized by the All-Night core without additional procedures, as these values are housed in the global variable segment of the shared memory.

3.2. Prototype processor

For the evaluation of our proposed processor, we meticulously developed prototypes for both the baseline and proposed processors using the RISC-V eXpress (RVX) tool [38]. As illustrated in Fig. 9(a), we present a structural overview of the baseline processor prototype. As aforementioned, we used Rocket core as the Main-CPU, while a quad-stage pipelined ORCA core is employed as the Sub-CPU. The system interconnect relies on the micro-NoC [39], tailored for power-efficient SoCs. The power manager stands as a pivotal element, transitioning the Day part to a standby state and orchestrating power gating. As with most processor designs [40–42], the CPUs interface with the Advanced Extensible Interface (AXI) to fulfill the imperatives of high throughput and minimal latency. Similarly, the SRAM aligns with this configuration. Conversely, the SPI, UART, I2C, IROM, JTAG, and FLASH are synchronized via APB interfaces, given their diminished performance requisites.
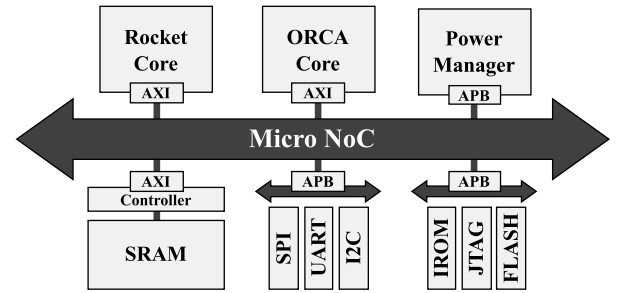
(a) The baseline processor prototype
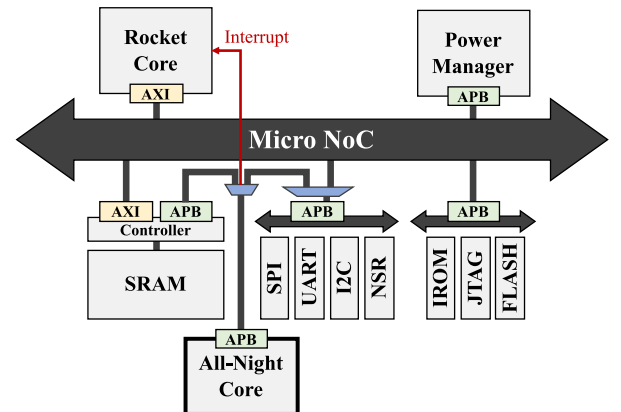
(b) The proposed processor prototype

Fig. 9. Implemented processor prototypes for verification and evaluation.
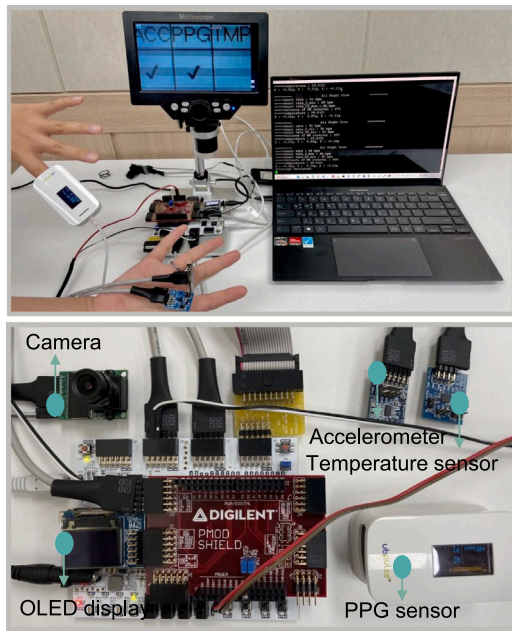
Fig. 10. Complete demonstration of running the anomaly detection app on the FPGA prototype processor.

Fig. 11. Comparison of power saving strategies of three different processors.

Transitioning to Fig. 9(b), we delineate the architectural representation of our proposed processor prototype. Adhering to the architectural backbone of the baseline, the Rocket remains our Main-CPU of choice. However, the Sub-CPU is innovatively replaced by our custom-developed All-Night core. To ensure unbiased juxtaposition, all IPs, excluding the NSR depicted in the figure, parallel those of the baseline. NSR, an acronym for Night Support Register, encompasses dual registers, specifically for *Night_addr* and *enable_Night* (refer to Fig. 7 for details). Accentuating the autonomy of the Night segment, we integrated a dual-port memory controller (consisting of AXI for the Main-CPU and APB for the All-Night core) with the extant main memory. It is imperative to note that these ports uphold a mutually exclusive operational paradigm. We further enhanced the design by embedding a multiplexer (mux), facilitating external I/O interfacing with the All-Night core bypassing the system interconnect. Another consequential inclusion was a subsequent mux, strategically positioned between the All-Night core, SRAM, and external I/O, thereby empowering the All-Night core to dispatch interrupts directly to the Main-CPU. Given the limited performance demands of the All-Night core, we opted for the APB protocol for all ensuing communications.

Additionally, for a comprehensive performance assessment of the Day–Night processor architecture, we conceived a distinct processor prototype, devoid of the All-Night core. This configuration mirrors the architecture delineated in Fig. 9(b), with the exception of utilizing the ORCA core as a surrogate for the All-Night core. Owing to its inherent resemblance, we abstain from an exhaustive elaboration of this structure.

To facilitate the execution of our curated anomaly detection application on the processor prototypes, we interfaced the FPGA board with a suite of sensors, encompassing PPG, accelerometer, and a temperature sensor, as vividly portrayed in Fig. 10. Parallelly, to emulate an application orchestrated by the Main-CPU—a prerequisite that necessitates suspension upon anomaly detection by the Sub-CPU—we integrated a camera module with the FPGA board and devised a rudimentary video application. Complementing this setup, an OLED display was affixed to the board, earmarked for disseminating emergency alert notifications. Moreover, to ensure seamless transmission of these alerts to a centralized server, a Bluetooth module was incorporated. For optimizing
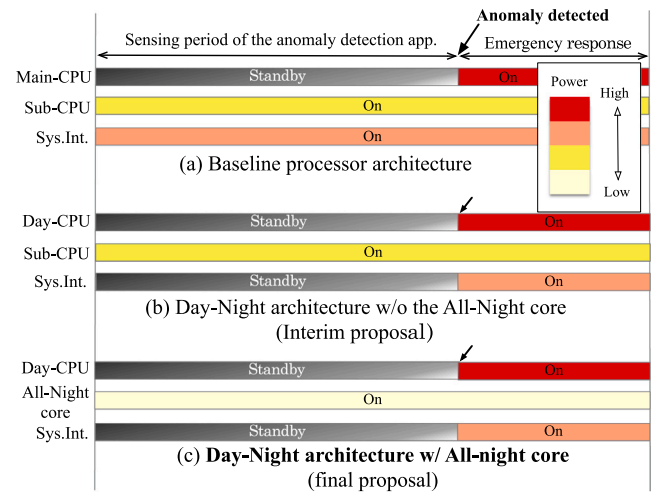
communication with the processor, we established UART interfacing for the PPG and monitor, I2C connectivity for the accelerometer and temperature sensor, and SPI compatibility for both the camera and OLED display.

4. Evaluation

4.1. Results from FPGA prototyping

Fig. 11 depicts the potential power-saving effects of three different processors. Specifically, (a), (b), and (c) represent the baseline processor, a processor following the Day–Night structure with the ORCA as its Sub-CPU, and a processor based on the Day–Night architecture that incorporates the All-Night core, respectively. The period of interest in this figure is when only the anomaly detection sensor value processing is active, with the Main-CPU being inactive. Within this context, the respective scenarios depicted in (a), (b), and (c) are referred to as Case-I, Case-II, and Case-III. In detail: (i) Case-I has both the Sub-CPU and micro-NoC (i.e., system interconnect) active, (ii) In Case-II, only the Sub-CPU is active while the micro-NoC is deactivated, and (iii) Case-III operates most efficiently, with only the All-Night core active.

Leveraging the Digilent Arty A7 FPGA board [43], we prototyped both the baseline and proposed processors, each operating at a clock frequency of 50 MHz. Table 3 details the FPGA prototyping results for Cases-I, -II, and -III. The table lists the resources (LUTs and FFs) and the estimated power consumption for the components used in the three processor prototypes. It is worth noting that among these components, the Rocket core interface and ORCA core interface serve as network interface modules required for connecting each core to the Micro-NoC. The external peripheral encompasses modules that manage external sensor communication protocols such as I²C, SPI, and GPIO, and includes the entire module that connects these to the micro-NoC with APB. The table indicates whether each component is included in each processor prototype and also details the *on* or *standby* states of each component.

In the table, the resource (LUTs, and FFs) and the estimated power consumption of the components used in the three processor prototypes are indicated, and whether they are included in each processor is also marked. In addition, the *on* or *standby* states of each component are also included.

Upon examining the resource and power metrics, it becomes evident that the Rocket, serving as the Main-CPU across all configurations, dominates in terms of resource and power usage, considerably exceeding the metrics of other components. The ORCA consumes approximately 7.8 times fewer LUTs and 3.9 times fewer FFs than Rocket,

Table 3
FPGA prototyping results for the three cases.

Components	LUTs, FFs	$P_{dynamic}$ (mW)	P_{static} (mW)	Case-I	Case-II	Case-III
Rocket core	17 723, 8450	46.8	17.2	*incl.* *(standby)*	*incl.* *(standby)*	*incl.* *(standby)*
ORCA core	2263, 2149	8.1	2.9	*incl.* *(on)*	*incl.* *(on)*	*excl.*
All-Night core	1381, 1397	2.3	0.7	*excl.*	*excl.*	*incl.* *(on)*
Micro -NoC	4571, 5967	8.9	2.0	*incl.* *(on)*	*incl.* *(standby)*	*incl.* *(standby)*
External peripherals	1196, 1350	1.5	0.5	*incl.* *(on)*	*incl.* *(on)*	*incl.* *(on)*
Rocket core interface	1884, 5285	4.8	1.2	*incl.* *(standby)*	*incl.* *(standby)*	*incl.* *(standby)*
ORCA core interface	1884, 5285	4.8	1.2	*incl.* *(on)*	*incl.* *(on)*	*excl.*
Normal peripherals	195, 235	2.3	0.7	*incl.* *(on)*	*excl.*	*excl.*
Day-Night peripherals	254, 329	3.1	0.9	*excl.*	*incl.* *(on)*	*incl.* *(on)*

Table 4
Estimated energy savings for the baseline processor (using Rocket as the main CPU and ORCA as the Sub-CPU) and the Day–Night processor, both prototyped on FPGA.

	Baseline processor	Day-Night processor
Resource consumption (LUTs, FFs)	29 716, 28 719	27 009, 22 778
Power (mW)	51.4	29.5
Energy (μJ)	1185.2	743.7
Energy saving (%)	–	37.3

resulting in nearly 5.8 times less estimated power consumption. The All-Night core further trims this consumption, using roughly 1.6 times fewer LUTs and FFs than ORCA and reducing estimated power consumption by nearly 3.7 times in comparison to ORCA. This strongly suggests that the All-Night core is optimized for both area and power consumption.

The micro-NoC, despite being significantly smaller than the Main-CPU, consumes about 2.4 times more resources than the Sub-CPU ORCA, and its estimated power consumption aligns closely with that of ORCA. This reiterates the fact that system interconnect power usage can be significant, particularly when only the anonymous detection app is operated. The results of the peripherals are also reported in the table, revealing that their resource and power consumption figures are on par with the All-Night core, further emphasizing the All-Night core's minimalist design.

Moreover, components newly incorporated or modified within our proposed architectural design are labeled as Day–Night peripherals in the table. These encompass the AXI-APB dual-port SRAM controllers, two multiplexers, and NSR, as depicted in Fig. 9(b). Conversely, the peripherals corresponding to the baseline processor are designated as normal peripherals, including the AXI single port SRAM controller. These normal peripherals are replaced with Day–Night peripherals in the Day–Night architecture. It is evident that while Day–Night peripherals consume slightly more resources and power than their normal counterparts, the overhead remains minimal when juxtaposed with other components.

From the component states of each processor, we can calculate their respective estimated power consumptions. Case-I registers a power consumption of 51.4 mW, whereas Case-II consumes 42.4 mW, marking an approximate power savings of 17.3% owing to the Day–Night structure. Impressively, Case-III operates at only 29.5 mW, slashing the power consumption by 42.6% when compared to the baseline processor.

The energy efficiency of the proposed architecture was further validated by measuring the execution time of the anomaly detection application across each prototype. The execution time consisted of the Sub-CPU's time to read and process sensor values (T_{active}), plus the standby time ($T_{standby}$) when it was clock-gated after completing its tasks. T_{active} was 11.55 ms for Case-I, with Case-II being slightly faster at 10.98 ms. The minimal time difference between Case-I and II could be attributed to unchanged factors in core performance, compilation strategy, and memory transaction volume. The slight speed advantage of Case-II is likely due to altered traffic patterns within the system interconnect. Meanwhile, for Case-III, T_{active} was observed to be 25.21 ms, indicating that the All-Night core was approximately 2.2 times slower compared to the ORCA core. However, considering that the All-Night core consumes only about 27.8% of the power compared to the ORCA core, it is evident that our approach of balancing minimal instructions with a 3-stage design to mitigate performance degradation in the All-Night core was effective. Lastly, with the application's sample rate set at 1/25.21 ms (Hz), Table 4 reporting the energy consumption based on execution times and energy savings between the baseline and proposed designs clearly demonstrates that the proposed Day–Night processor achieves an energy saving of approximately 37.3%.

4.2. Results from synthesis with 45 nm technology

To precisely assess the power reduction efficacy of the proposed architecture, we synthesized both the baseline and proposed processors with the Nangate 45 nm technology library [37] using Synopsys Design Compiler [44]. Consistent with FPGA prototyping, both processors were synthesized with a clock frequency of 50 MHz. The results, illustrated in Fig. 12, retain the definitions of Case-I, Case-II, and Case-III from the previous section. More specifically, Fig. 12(a) delineates the dynamic and static power consumption, as well as the cell usage breakdown for each module. The figure prominently displays that the micro-NoC has a substantial power consumption and resource footprint compared to other modules. This distinction is even more pronounced in Fig. 12(b) and Fig. 12(c), revealing respective cell usage and power consumption for each case (The dark gray-colored parts in (c) denote the clock-gated modules where the dynamic power is zero, while the light gray-colored parts represent the static power of the clock-gated modules). Notably, due to the ability to clock-gate the micro-NoC in the Day–Night structure, there is a substantial reduction in power consumption: Case-I consumed 20621.3 μW, whereas Case-II only consumed 12413.1 μW, marking a 39.8% power saving.

Observing Fig. 12(b) and (c) elucidates the advantages of the All-Night core. While the cell count of the Case-II processor slightly exceeds that of Case-I due to the addition of the AXI-APB dual-port SRAM controllers, two multiplexers, and NSR for the Day–Night structure, the Case-III processor, thanks to the All-Night core, uses fewer cells. The All-Night core requires only 2.05 times fewer cells than ORCA, making the Case-III configuration the most efficient. In terms of power, the consumption of the Case-III processor stands at 8545.8 μW, achieving a 31.2% reduction from Case-II, and a comprehensive 58.6% reduction when compared to the baseline.

Next, to assess the energy efficiency of our proposed processor architecture, we derived the energy consumption when executing four different anomaly detection applications on the processor prototype. These applications include our self-developed test application and others that have been directly programmed in assembly to operate on the All-Night core, based on algorithms proposed in previous works [5, 6,9]. Briefly, the algorithm from [5] calculates the magnitude of acceleration collected from an accelerometer mounted on the upper body, applies a low-pass filter, and then determines falls based on a predefined threshold, achieving a specificity (true positive rate) and sensitivity (true negative rate) of 91.3% and 100%, respectively. We developed this application to operate entirely on the Sub-CPU. The algorithm from [6] detects collisions based on a set threshold and
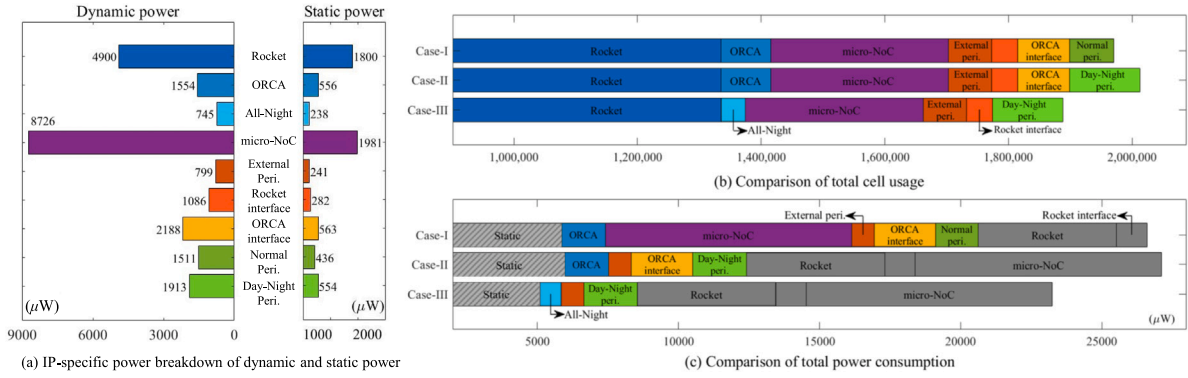
Fig. 12. Power consumption results and analysis of power savings for three processor prototypes synthesized with 45 nm process technology during exclusive execution of the anomaly detection test application.

Table 5
Comparison of Sub-CPU energy consumption for different applications when using Rocket, ORCA, and All-Night as the Sub-CPU configurations. E_{Rocket}, E_{ORCA}, and $E_{All-Night}$ stand for energy consumption of Rocket, ORCA, and All-Night, respectively.

App.	Sensors	Sample rate (Hz)	Rocket		ORCA		All-Night		Energy saving (%)	
			T_{active} (µs)	E_{Rocket} (µJ)	T_{active} (µs)	E_{ORCA} (µJ)	T_{active} (µs)	$E_{All-Night}$ (µJ)	Compared to Rocket	Compared to ORCA
[5]	Accelerometer	105	5575	37.35	6270	13.2	9510	9.4	79.0	37.8
Test app. in Fig. 8	Accelerometer, PPG sensor, temperature sensor	40	8983	60.2	11 554	24.4	25 210	24.8	72.3	22.5
[6]	Accelerometer	30	6869	46.0	11 862	25.1	33 045	32.5	65.1	11.8
[9]	Accelerometer, PPG sensor	21	7491	50.2	14 317	30.2	47 985	47.2	61.7	3.6

extracts eight features from the acceleration data collected within a window around the collision event for a classifier model to determine the presence of an anomaly, with specificity and sensitivity of 95.6% and 83.3%(in the case of the SVM classifier), respectively. Assuming the complex classifier part operates on the Main-CPU, we excluded it from implementation and developed an app that performs up to the feature extraction pre-processing on the Sub-CPU. Similarly, [9] describes an algorithm that uses a Gaussian Mixture Model on feature vectors extracted from heart rate and acceleration sensor data to detect anomalies via SVM, with specificity and sensitivity of 98.9% and 97.1%, respectively. Like the app for [6], we developed an app that excludes the classifier and operates the feature extraction on the Sub-CPU. Our self-developed application, designed for testing both the low-power performance and functional verification of the developed processor, utilizes three sensors and tests the operation up to the Main-CPU. However, we did not evaluate the specificity and sensitivity of this particular application, leaving the in-depth study of the anomaly detection algorithm itself for future research.

Meanwhile, determining the execution times of the developed anomaly detection applications presents another challenge. Executing these applications on the synthesized processor is highly challenging and time-consuming. To overcome this, we relied on previous research that validated the temporal congruence between an FPGA prototype developed by RVX and an actual SoC [38]. Consequently, we borrowed the execution times obtained from running these applications on the FPGA prototype. The results are presented in Table 5, which lists the T_{active} values, representing the duration for which the Sub-CPU remains active to process each application, organized by each core used as the Sub-CPU.

Before evaluating the overall energy-saving impact of our proposed processor architecture, we analyzed the energy savings of the All-Night core specifically, with the results reported in Table 5. In addition to comparing the energy consumption of the All-Night core with the ORCA core, used as the Sub-CPU in the baseline processor, we also conducted comparisons with the energy consumption of the Rocket

Table 6
Energy saving results of the proposed processor.

App.	$E_{baseline}$ (µJ)	$E_{Day-Night}$ (µJ)	Energy saving (%)
[5]	191.1	81.3	57.5
Test app. in Fig. 8	498.6	215.4	56.8
[6]	648.5	282.4	56.4
[9]	937.2	410.0	56.2

core. For each application, the sample rate was set based on the time it took for the All-Night core to execute the application (i.e., the All-Night core was not clock-gated), while the higher-performance Rocket and ORCA cores were set to transition to a clock-gated standby state during the remaining time until the next sample was processed. For instance, in our self-developed test application, the All-Night core processed the given task in 25.21 ms, whereas the ORCA and Rocket cores completed the same task more quickly, in 11.55 ms and 8.98 ms, respectively, thus spending 13.65 ms and 16.23 ms in a standby state consuming only static energy. As demonstrated in Table 5, the All-Night core proved to be a more energy-efficient choice in all cases compared to the ORCA or Rocket cores. More detailed analysis reveals that applications with longer execution times on the All-Night core tend to show reduced energy savings. This is due to the significant difference between the cores' dynamic and static power; as the ORCA and Rocket cores quickly complete tasks and spend longer durations in standby consuming static power, the gap narrows with the continuously active All-Night core. However, this trend does not scale linearly, as the operational performance of the cores (and thus their power consumption) does not directly correlate with their T_{active}. This is because real-time sensor systems based on external sensors utilize slower external serial communications to receive data, limiting the impact of the cores' computational speed on the overall task execution time.

In essence, for applications where the time spent communicating with external sensors is lengthy, high-performance cores maintain

high operational power while waiting for sensor responses, whereas slower, lower-power cores like the All-Night core are advantageous for such tasks. Consequently, the highest energy savings were observed in case [5], which demands the simplest operations and minimal computation, while case [9], requiring intense computations with relatively less time spent on sensor communication, showed the least energy savings. A comparison between our self-developed test app and [6] shows that despite similar execution times for the two applications on the All-Night core, the test app's use of more sensors increases the proportion of time spent on sensor communication, resulting in significantly greater energy savings for the All-Night core in the test app compared to [6].

Furthermore, the energy-saving results in Table 5 consider only the scenario where the All-Night core remains constantly active and not clock-gated, representing the minimum savings achievable by the All-Night core. In other words, if we were to reduce the sample rate below the values in the table, allowing the All-Night core to enter a standby state and consume only static power for extended periods, as reported in Fig. 12, the significant difference in static power between the All-Night and other cores should be reflected in the energy savings results. However, reducing the sample rate could compromise the accuracy of the anomaly detection app, so this aspect was excluded from the analysis in our paper.

Finally, we derived the energy savings results for the entire processor architecture. The execution times for each application remained consistent with those used in the earlier comparison between the All-Night core and other cores: the results for the ORCA were used for the baseline processor, and the results for the All-Night were applied to the Day–Night architecture. Table 6 underscores the remarkable energy savings of our proposal: the Day–Night processor achieved a maximum energy saving of 57.5% (when running application [5]) and a minimum of 56.2% (when running application [9]).

5. Conclusion

Healthcare has always been at the forefront of leveraging technology for better patient outcomes. In this pursuit, the role of anomaly detection in wearable devices holds a cardinal significance. This research introduced the *Day–Night* architecture for ULP processors tailored to enhance the efficiency of wearables dedicated to this critical application. The proposed architecture allows the processor to selectively clock-gate the system interconnect, a major power consumer, during anomaly detection, resulting in substantial power savings. The introduction of the *All-Night* core as an optimized micro-core further underscores the processor's energy efficiency. To demonstrate the viability of our proposed architecture, we developed a prototype processor on an FPGA board. Using the self-developed anomaly detection application, we showcased its functionality. Synthesizing the developed processor prototype with 45 nm process technology and analyzing the energy consumption for four types of anomaly detection applications resulted in achieving energy savings of up to 57.5%. This substantiates our claim of significant energy reduction in the proposed processor architecture.

CRediT authorship contribution statement

Eunjin Choi: Writing – original draft. **Jina Park:** Writing – original draft. **Kyeongwon Lee:** Writing – original draft. **Jae-Jin Lee:** Methodology. **Kyuseung Han:** Writing – review & editing. **Woojoo Lee:** Writing – review & editing, Writing – original draft, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] J. Park, E. Choi, K. Lee, J.-J. Lee, K. Han, W. Lee, Developing an ultra-low power RISC-V processor for anomaly detection, in: Design, Automation & Test in Europe Conference & Exhibition, DATE, 2023, pp. 1–2.

[2] A. Mahajan, G. Pottie, W. Kaiser, Transformation in healthcare by wearable devices for diagnostics and guidance of treatment, ACM Trans. Comput. Healthc. 1 (1) (2020) 1–12.

[3] J. Hua, Y. Xu, J. Tang, J. Liu, J. Zhang, ECG heartbeat classification in compressive domain for wearable devices, J. Syst. Archit. 104 (2020) 101687.

[4] S.M.A. Iqbal, I. Mahgoub, E. Du, M.A. Leavitt, W. Asghar, Advances in healthcare wearable devices, Npj Flex. Electron. 5 (1) (2021).

[5] A. Bourke, J. OBrien, G. Lyons, Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm, Gait Posture 26 (2) (2007) 194–199, http://dx.doi.org/10.1016/j.gaitpost.2006.09.012, URL https://www.sciencedirect.com/science/article/pii/S0966636206001895.

[6] S.B. Khojasteh, J.R. Villar, C. Chira, V.M. González, E. De la Cal, Improving fall detection using an on-wrist wearable accelerometer, Sensors 18 (5) (2018).

[7] S.S. Fakhrulddin, S.K. Gharghan, An autonomous wireless health monitoring system based on heartbeat and accelerometer sensors, J. Sensor Actuator Netw. 8 (3) (2019).

[8] J.-S. Lee, H.-H. Tseng, Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers, IEEE Sens. J. 19 (18) (2019) 8293–8302.

[9] Y.-H. Nho, J.G. Lim, D.-E. Kim, D.-S. Kwon, User-adaptive fall detection for patients using wristband, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2016, pp. 480–486.

[10] G. Sivapalan, K.K. Nundy, S. Dev, B. Cardiff, D. John, ANNet: A lightweight neural network for ECG anomaly detection in IoT edge sensors, IEEE Trans. Biomed. Circuits Syst. 16 (1) (2022) 24–35.

[11] M. Gu, Y. Zhang, Y. Wen, G. Ai, H. Zhang, P. Wang, G. Wang, A lightweight convolutional neural network hardware implementation for wearable heart rate anomaly detection, Comput. Biol. Med. 155 (2023) 106623.

[12] G. Sivapalan, K.K. Nundy, A. James, B. Cardiff, D. John, Interpretable rule mining for real-time ECG anomaly detection in IoT edge sensors, IEEE Internet Things J. 10 (15) (2023) 13095–13108.

[13] C. Zhuo, S. Luo, H. Gan, J. Hu, Z. Shi, Noise-aware DVFS for efficient transitions on battery-powered IoT devices, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 39 (7) (2020) 1498–1510.

[14] T. Zhou, M. Lin, CPU frequency scheduling of real-time applications on embedded devices with temporal encoding-based deep reinforcement learning, J. Syst. Archit. 142 (2023) 102955.

[15] P. Mercati, R. Ayoub, M. Kishinevsky, E. Samson, M. Beuchat, F. Paterna, T.Š. Rosing, Multi-variable dynamic power management for the GPU subsystem, in: 2017 54th ACM/EDAC/IEEE Design Automation Conference, DAC, 2017, pp. 1–6.

[16] Y. Pu, C. Shi, G. Samson, D. Park, K. Easton, R. Beraha, A. Newham, M. Lin, V. Rangan, K. Chatha, D. Butterfield, R. Attar, A 9-mm2 ultra-low-power highly integrated 28-nm CMOS SoC for internet of things, IEEE J. Solid-State Circuits 53 (3) (2018) 936–948.

[17] S. Umesh, S. Mittal, A survey of techniques for intermittent computing, J. Syst. Archit. 112 (2021) 101859.

[18] J. Lee, Y. Zhang, Q. Dong, W. Lim, M. Saligane, Y. Kim, S. Jeong, J. Lim, M. Yasuda, S. Miyoshi, M. Kawaminami, D. Blaauw, D. Sylvester, A self-tuning IoT processor using leakage-ratio measurement for energy-optimal operation, IEEE J. Solid-State Circuits 55 (1) (2020) 87–97.

[19] E. De Giovanni, F. Montagna, B.W. Denkinger, S. Machetti, M. Peón-Quirós, S. Benatti, D. Rossi, L. Benini, D. Atienza, Modular design and optimization of biomedical applications for ultralow power heterogeneous platforms, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 39 (11) (2020) 3821–3832.

[20] D. Rossi, F. Conti, M. Eggiman, A.D. Mauro, G. Tagliavini, S. Mach, M. Guermandi, A. Pullini, I. Loi, J. Chen, E. Flamand, L. Benini, Vega: A ten-core SoC for IoT endnodes with DNN acceleration and cognitive wake-up from MRAM-based state-retentive sleep mode, IEEE J. Solid-State Circuits 57 (1) (2022) 127–139.

[21] M. Janveja, R. Parmar, G. Trivedi, P. Jan, Z. Nemec, An energy efficient and resource optimal VLSI architecture for ECG feature extraction for wearable healthcare applications, in: 2022 32nd International Conference Radioelektronika, RADIOELEKTRONIKA, 2022, pp. 1–6.

[22] C. Jie, I. Loi, L. Benini, D. Rossi, Energy-efficient two-level instruction cache design for an ultra-low-power multi-core cluster, in: DATE, 2020, pp. 1734–1739.

[23] A. Suyyagh, Z. Zilic, Energy and task-aware partitioning on single-ISA clustered heterogeneous processors, IEEE Trans. Parallel Distrib. Syst. 31 (2) (2020) 306–317.

[24] E. Shamsa, A. Kanduri, P. Liljeberg, A.M. Rahmani, Concurrent application bias scheduling for energy efficiency of heterogeneous multi-core platforms, IEEE Trans. Comput. 71 (4) (2022) 743–755.

[25] J. Park, K. Han, E. Choi, S. Lee, J.-J. Lee, W. Lee, M. Pedram, Florian: Developing a low-power RISC-V multicore processor with a shared lightweight FPU, in: IEEE/ACM International Symposium on Low Power Electronics and Design, ISLPED, 2023, pp. 1–6.

[26] C. Tan, M. Karunaratne, T. Mitra, L.-S. Peh, Stitch: Fusible heterogeneous accelerators enmeshed with many-core architecture for wearables, in: ACM/IEEE ISCA, 2018, pp. 575–587.

[27] Y. Wei, Q. Cao, L. Hargrove, J. Gu, A wearable bio-signal processing system with ultra-low-power SoC and collaborative neural network classifier for low dimensional data communication, in: EMBC, 2020, pp. 4002–4007.

[28] J. Nunez-Yanez, N. Howard, Energy-efficient neural networks with near-threshold processors and hardware accelerators, J. Syst. Archit. 116 (2021) 102062.

[29] L. Mei, P. Houshmand, V. Jain, S. Giraldo, M. Verhelst, ZigZag: Enlarging joint architecture-mapping design space exploration for DNN accelerators, IEEE Trans. Comput. 70 (8) (2021) 1160–1174.

[30] SiFIVE, Accessed 17 February 2024, https://github.com/chipsalliance/rocket-chip.

[31] Vectorblox, Accessed 17 February 2024, https://github.com/riscveval/orca-1.

[32] RISC-V, Accessed 17 February 2024, https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf.

[33] YosysHQ, Picorv32-risc-v, 2024, https://github.com/YosysHQ/picorv32, Accessed 17 February 2024.

[34] SI-RISCV, E203-risc-v, 2024, https://github.com/SI-RISCV/e200_opensource, Accessed 17 February 2024.

[35] onchipuis, Mriscvcore-risc-v, 2024, https://github.com/onchipuis/mriscvcore, Accessed 17 February 2024.

[36] liangkangnan, Tinyriscv-risc-v, 2024, https://github.com/liangkangnan/tinyriscv, Accessed 17 February 2024.

[37] NCSU, FreePDK45, 2024, https://eda.ncsu.edu/freepdk/freepdk45, Accessed 17 February 2024.

[38] K. Han, S. Lee, K.-I. Oh, Y. Bae, H. Jang, J.-J. Lee, W. Lee, M. Pedram, Developing TEI-aware ultralow-power SoC platforms for IoT end nodes, IEEE Internet Things J. 8 (6) (2021) 4642–4656.

[39] K. Han, S. Lee, J.-J. Lee, W. Lee, M. Pedram, TIP: A temperature effect inversion-aware ultra-low power System-on-Chip platform, in: IEEE/ACM International Symposium on Low Power Electronics and Design, ISLPED, 2019, pp. 1–6.

[40] R. Höller, D. Haselberger, D. Ballek, P. Rössler, M. Krapfenbauer, M. Linauer, Open-source RISC-V processor IP cores for FPGAs — Overview and evaluation, in: MECO, 2019, pp. 1–6.

[41] H. Jang, K. Han, S. Lee, J.-J. Lee, S.-Y. Lee, J.-H. Lee, W. Lee, Developing a multicore platform utilizing open RISC-V cores, IEEE Access 9 (2021) 120010–120023.

[42] S. Pinto, P. Machado, D. Oliveira, D. Cerdeira, T. Gomes, Self-secured devices: High performance and secure I/O access in TrustZone-based systems, J. Syst. Archit. 119 (2021) 102238.

[43] Digilent, Arty A7, 2024, https://digilent.com/reference/programmable-logic/arty-a7/start, Accessed 17 February 2024.

[44] Synopsys, Design compiler, 2024, https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html, Accessed 17 February 2024.

Eunjin Choi is a graduate researcher at the Low-power SoC lab in the School of Electrical & Electronics Engineering, Chung-Ang University, Seoul, Korea, pursuing her M.S. degree. Her research interests revolve around real-time systems, processor architectures, and low power SoC designs. Eunjin received the Ministerial Award from the Ministry of Trade, Industry, and Energy of the Republic of Korea for her outstanding performance in the Korea Semiconductor Design Competition in both 2021 and 2022.

Jina Park is a graduate researcher at the Low-power SoC lab in the School of Electrical & Electronics Engineering, Chung-Ang University, Seoul, Korea, pursuing her M.S. degree. Her research interests revolve around ultra-low power design, SoC architecture, and computer-aided design automation. Jina received the Ministerial Award from the Ministry of Trade, Industry, and Energy of the Republic of Korea for her outstanding performance in the Korea Semiconductor Design Competition in both 2021 and 2022.

Kyeongwon Lee is a graduate researcher at the Low-power SoC lab in the School of Electrical & Electronics Engineering, Chung-Ang University, Seoul, Korea, pursuing his M.S. degree. His research interests revolve around low power design, processor architecture, and embedded systems. Kyeongwon received the Ministerial Award from the Ministry of Trade, Industry, and Energy of the Republic of Korea for his outstanding performance in the Korea Semiconductor Design Competition in 2022.

Jae-Jin Lee received the B.S., M.S., and Ph.D. degrees in computer engineering from Chungbuk National University, Cheongju, South Korea, in 2000, 2003, and 2007, respectively. He is currently a Project Leader with the low-power AI system-on-chip (SoC) Design Research Division, Electronics and Telecommunications Research Institute, Daejeon, Korea. His research interests include ultra-low-power deeply embedded RISC-V processor designs and event-driven neuromorphic computing architectures for brain-inspired spiking deep neural networks (SNNs).

Kyuseung Han received the B.S. and Ph.D. degrees in Electrical Engineering and Computer Science from Seoul National University (SNU), Seoul, Korea, in 2008 and 2013. At SNU, he researched on computer architecture and design automation. From 2014, Dr. Han has been working at Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, and he currently belongs to the SoC Design Research Group as a senior researcher. His current research interests include reconfigurable architecture, network-on-chip, and ultra-low power techniques in embedded systems.

Woojoo Lee received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2007, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2010 and 2015, respectively. He was a Senior Researcher with the Electronics and Telecommunications Research Institute, Daejeon, South Korea, from 2015 to 2016, and an Associate Professor with the system on chip (SoC) Design Research Group, Department of Electrical Engineering, Myongji University, Seoul, from 2017 to 2018. He is currently an Associate Professor with the School of Electrical and Electronics Engineering, Chung-Ang University, Seoul. His research interests include ultralow-power VLSI designs, SoC designs, spiking neural network designs, and system-level power and thermal management.