A Silicon-Proof Controller System for Flexible Ultra-Low-Power Energy Harvesting Platforms

Moritz Weißbrich*, Holger Blume† and Guillermo Payá-Vayá*

*Chair for Chip Design for Embedded Computing, Technische Universität Braunschweig, Braunschweig, Germany
{m.weissbrich, g.paya-vaya}@tu-braunschweig.de

†Institute of Microelectronic Systems, Leibniz Universität Hannover, Hannover, Germany
blume@ims.uni-hannover.de

***Abstract*—In this paper, a heterogeneous controller system and its first-silicon ASIC implementation are presented, where the use of a programmable NanoController next to a general-purpose microcontroller enables more efficient and flexible power management strategies than typical timer-based, periodical power-up of a single microcontroller in state-of-the-art IoT devices. The NanoController features a compact, control-oriented 4-bit ISA, which is used to continuously pre-process data in order to decide when to power-up the microcontroller required for infrequent complex processing, e.g., encrypted wireless communication. Despite its programmability, the required silicon area and power consumption are very small and enable the use in the always-on domain of SoCs for energy harvesting platforms, instead of much simpler and constrained timer circuits. The first-silicon ASIC implementation of such a controller system using a 65nm UMC low-leakage process is presented and evaluated for a real home automation application intended to operate on harvested energy, i.e., electronic door lock, reducing the average power consumption of reference microcontrollers by up to 20x.**

***Index Terms*—ASIC, application-specific microcontroller, embedded system, energy efficiency, energy harvesting platform, ultra-low-power**

I. Introduction

Energy Harvesting (EH) can be an effective method to operate embedded systems in remote sites that are difficult to reach by cables, and to eliminate expensive battery replacement cycles. Examples of such systems may be IoT devices, distributed wireless sensor networks [1] or devices in home automation, e.g., electronic door locks or room temperature control [2]. In completely battery-less devices powered by EH, power is provided solely by harvesting energy from environmental sources, e.g., solar, thermal or RF energy. The energy income via EH is often unpredictable and intermittent [3], making energy a precious resource and energy efficiency an absolute necessity in these systems [1]. For electronic door locks in indoor environments, it was observed that continuous system operation must be ensured at ultra-low power budgets (below 1 µW average for the microcontroller system) in phases of low energy income [2]. The above-mentioned applications have in common that there are relatively simple events with frequent execution, e.g., checking for key cards, or waiting for incoming wireless transmissions multiple times per second. But only in significantly less frequent cases, a complex system reaction is required, e.g., encrypted wireless data exchange only few times per day.

A conventional system approach is to use a single low-power microcontroller core to detect and process all frequent and infrequent events at discrete points in time, triggered by a periodical timer. This scheme can be applied to a majority of applications, however, it can be very challenging to achieve the necessary energy efficiency for EH platforms. Although there are *sleep modes*, which disable the instruction fetch, data path and unused peripherals in order to reduce the always-on power consumption, the complete general-purpose microcontroller periodically wakes up from sleep in order to catch every possible event that could occur. As an example, the EnOcean Dolphin Core for EH-based IoT [4] requires 1.3 µW in sleep mode with a periodical wake-up timer running, but consumes up to 7 mW for wake-up at every timer overflow, even if no event reaction is required. Depending on the application and event frequency, this can increase the average power consumption to several microwatts or more, possibly exceeding the available power budget for continuous operation. By using an always-on timer device like the *NanoTimer* and additionally applying *power gating* techniques [5], the always-on power consumption can be further decreased, however, this concept does not address the issue of periodical wake-up. Consequently, it has been identified that, for wireless sensor networks, a *conditional* wake-up mechanism, based on received packet information and modeled available energy, increases the energy efficiency compared to periodical wake-up [1]. The concept can be applied to other application fields as well. However, this approach requires a second controller instead of just a periodical timer to implement the intelligent strategy for power gating and conditional wake-up of the general-purpose microcontroller. If this controller is supposed to replace a small and simple periodical timer in the always-on power domain of a system-on-chip (SoC), *ultra-low* silicon area, leakage and active power consumption on the one hand, but also sufficient programming flexibility for implementing power management strategies on the other hand are required.

In this paper, a silicon-proof, heterogeneous controller system is presented in order to eliminate periodical wake-up and power consumption of a large general-purpose processor core. For the always-on power domain of the system, a fully programmable *NanoController* with *very small* silicon area and a compact, control-oriented 4 bit instruction-set architecture (ISA) is used, which has been designed for control flow and implementing power management strategies beyond sleep modes and periodical wake-up actions. Compared to general-purpose controllers, the architecture features a very small code size and power consumption in active operation. The required logic resources for control and data path are *only 220 logic gates* (UMC 65 nm standard cell technology). Based on decisions under program control of the *NanoController*, the large power-gated microcontroller in an on-off domain is activated only for absolutely necessary infrequent events, e.g., encrypted wireless data transmission dependent on specific sensor values, in order to minimize the power consumption.

The contributions of this paper are as follows:

- Presentation and description of the silicon-proof *NanoController* system concept,
- presentation of the manufactured standard cell ASIC implementation in UMC 65 nm low-leakage technology, applying the *NanoController* concept coupled with a TTA-based general-purpose microcontroller,
- and evaluation of the power consumption with real-world measurements in a home automation application intended to operate on EH (electronic door lock), showing power consumption reduction of up to 20x compared to a reference implementation.

The paper is organized as follows: In Section II, related work and state-of-the-art approaches for power-sensitive energy harvesting applications are presented. The *NanoController* system concept is described in Section III. Details on the manufactured ASIC and evaluation application are given in Section IV. In Section V, the power measurement results are presented and compared to reference values. Section VI concludes the paper.

II. Related Work

As a reference for the later evaluation of the presented system, the typical power consumption in sleep mode of general-purpose microcontrollers is of particular interest. Since it cannot be the scope of this paper to present a comprehensive overview of the numerous low-power controller architectures on the market, this section is limited to a selection of microcontroller implementations known to be used in products powered by EH. On the one hand, there are traditional 8 or 16 bit RISC and multi-cycle architectures, e.g., AVR, PIC or MSP430 [4], [6]–[9]. On the other hand, IoT devices with higher performance requirements are based on more modern 32 bit controllers, mostly using ARM Cortex processor cores [10]–[12]. In all compared cases, *sleep modes* are provided to disable the core CPU (instruction fetch and data path) via clock gating. Commonly, five to seven different stages of sleep are provided for fine-grained control of active IO pins and peripherals in an application [8], [9], [11], [12]. More advanced power reduction mechanisms include internal *dynamic voltage scaling* according to the current sleep mode in the STM32L432 controller [12], or adaptive *back-bias voltage control* to minimize leakage current in the Renesas RE01 controller [11]. Taking all above-mentioned controllers and mechanisms into account, the power consumption in sleep mode is in the range between 0.5 and 4 µW when a watch dog timer is running for periodical wake-up, which will be the reference range considered in this paper. This range is also supported by two silicon-proof ultra-low-power processor implementations [13], [14], using 65 nm CMOS technologies comparable to this work, with 1.7 and 1 µW, respectively. However, it should be mentioned that the focus of these publications is sub-threshold voltage operation at 0.4 to 0.5 V, which is not the scope of this paper. From an architectural point of view, standard implementations of a MSP430 controller core with conventional sleep modes are used. A separate statement on each individual reference is presented in Table II for the evaluation in Section V.

Beyond the integration of sleep modes and adaptive voltage regulation, the *NanoTimer* approach [5] applies *power gating* to a complete general-purpose microcontroller core. For this, the *NanoTimer* integrates analog timer circuitry to periodically power up the processor again after a specific time interval, achieving less than 0.4 µW power consumption of the always-on domain. Comparable concepts have been integrated in [15] and [16] for a 65 nm low-leakage sub-threshold and a 28 nm FDSOI implementation of an ARM Cortex M0+ core, respectively. These SoC designs for IoT devices have an always-on power domain, including a real-time clock (RTC) and the power management controller, achieving low always-on power consumption of 0.08 µW and 0.7 µW, respectively. However, all mentioned implementations are limited to deterministic periodical power-up of the core, or power-up on every external interrupt event, as intelligent power management strategies with dynamic power-up conditions are not supported. If processing is only required in specific cases in the application, this can cause the core to power up more frequently than necessary, consuming additional power. In contrast, the programmable *NanoController* approach presented in this paper can enable flexible power-up strategies for an EH platform, e.g., based on external sensor values or the current amount of available energy, while requiring only very small silicon area and power consumption in the always-on domain.

III. Architecture Design and Embedded System Concept

The presented controller system concept uses two processor cores (*NanoController* in always-on power domain, general-purpose controller (GPC) in on/off domain), targeting an ASIC/SoC implementation. Handshaking interfaces between the cores and a power management are provided to implement the control concept for the system and power supply, which is described in this section.

A. NanoController

The simplified architectural block diagram of the *NanoController* is depicted in Fig. 1. Based on the class of one-operand

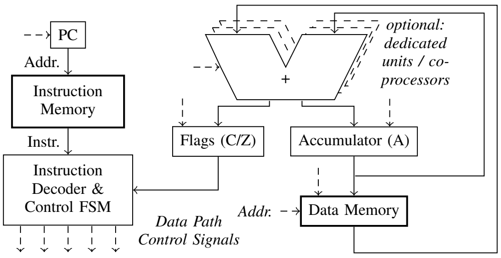
PC
Addr.
Instruction Memory
Instr.
Instruction Decoder & Control FSM
Data Path Control Signals
optional: dedicated units / co-processors
+
Flags (C/Z)
Accumulator (A)
Addr.
Data Memory

Fig. 1. Simplified block diagram of the NanoController architecture

accumulator/flag architectures, it has been designed with a minimal instruction set using 16 compact, control-oriented instructions encoded in 4 bit. These instructions combine load/store, increment/decrement, comparison and conditional branch operations for a memory-efficient software implementation of finite state machines (FSMs) and programs for system state control. This way, the silicon area and the consumed power of the instruction memory can be significantly reduced while maintaining a fully programmable controller for FSMs, power management control, etc. The limited design space size of compact 4 bit instructions also enables the designer to effectively optimize the binary instruction encoding for a certain class of control applications. This is achieved by minimizing the switching activity, e.g., in the instruction decoder and at the instruction memory output, for sequences of instructions in a program, further reducing power consumption [17]. Furthermore, *variable-length encoding* of literal values is applied as a key feature in order to reduce the size of small immediates and addresses in the instruction memory. The control unit of the *NanoController* utilizes multi-cycle instruction execution to implement this feature. Combining all these features, the *NanoController* partition of the control application shown in Section IV-B requires 21 % less instructions and 61 % smaller total code size than an implementation of the same functionality for the PIC12 accumulator architecture with a comparable 8 bit wide data path [7]. The complete *NanoController* data path including ALU, all registers, instruction decoder and control FSM fits into *only 220 standard cell gates* of the UMC 65 nm ASIC technology, emphasizing its minimal size.

An extensive architectural analysis is out of scope of this paper, which focuses on the ASIC implementation of an embedded system using the *NanoController*, and will be subject of a separate publication. Finally, it should be mentioned that the *NanoController* data path is not limited to control-oriented operations by design, but is prepared for specialization with additional functional units or co-processors. Therefore, it can also be interesting to apply the *NanoController* architecture as a platform for a tiny ASIP for specific processing requirements, e.g., continuous digital pre-filtering of audio samples, which is subject of current research.

B. General-Purpose Controller (GPC)

Generally, the system concept does not dictate any specific requirements for the general-purpose controller (GPC). Any processor architecture may be used, e.g., MIPS, RISC-V implementations, etc. The system implementation presented in this paper uses a minimum basic configuration of a 32 bit *transport-triggered architecture* (TTA) equipped with protocol peripherals (SPI, I2C, UART). The processor hardware description is specified and generated using the open-source *TTA Co-Design Environment* (TCE) toolset, which also provides the LLVM-based C compiler target [18]. TCE has been developed as a tool chain for designing high-performance and energy-efficient application-specific instruction-set processors (ASIPs) using the TTA template, which can be easily specialized for various IoT and complex signal processing applications [19], [20]. However, GPC specialization has not been the scope for the first silicon in this work, and as the minimum basic TTA configuration already features sufficient performance for 128 bit AES-encrypted RFID communication in the targeted door lock application [2], no further customization has been applied.

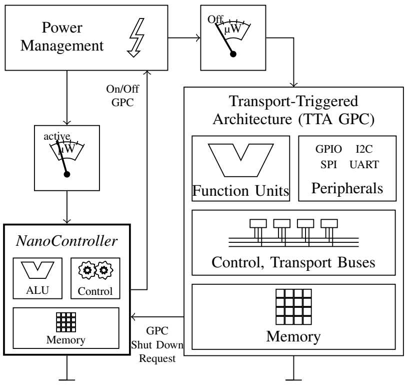

Fig. 2. Block diagram of the power management concept of the embedded system

C. System Control & Power Management Concept

Fig. 2 illustrates the power management control concept of the system. There are two power domains, i.e., one separate domain for each core. While the *NanoController* domain can be always-on for system control and controlling the power management, the GPC on/off domain is supposed to be off by default in order to avoid power consumption. In case of an infrequent event requiring complex processing, the GPC will be temporarily switched on, e.g., for RFID tag communication. The power gating is under program control of the *NanoController*, which receives shut-down requests from the GPC and sends on/off control signals to the power management.

IV. ASIC Evaluation Setup

The presented controller system has been implemented and manufactured as first silicon for demonstration and measurement purposes. Fig. 3 is a photograph showing the ASIC evaluation setup for the power measurements in this paper. In addition to the ASIC breakout and application peripheral boards on the left, there are four Amprobe AM-540 digital multimeters to capture voltage and current of both core logic power domains described in Section III-C. The evaluation

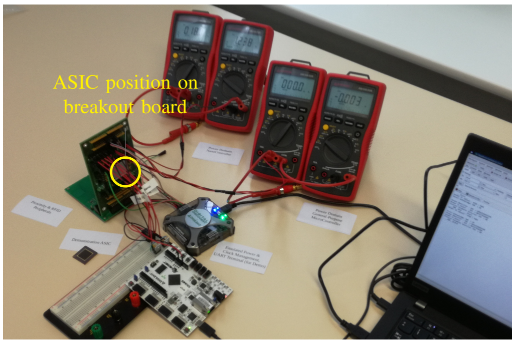

Fig. 3. ASIC power measurement setup

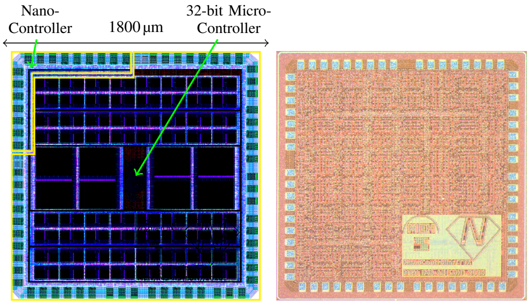

Fig. 4. Layout view (left) and microscope photograph (right) of the manufactured ASIC

ASIC implements the digital-domain controller system only, so the analog and mixed-signal power and clock management are out of scope of this work. For this paper, these components are emulated using FPGA-based development devices (Digilent Digital Discovery signal generator, Digilent Arty-A7) shown at the lower edge of the photograph. The notebook is used to monitor the UART debugging interface and to supply power to the system. As a consequence of the emulated power management, real EH sources are not in use in this first-silicon evaluation setup. Nonetheless, the intended use case is an EH-powered device platform, and activities on integrating the ASIC into such a platform are currently initiated. In the following, details on the manufactured ASIC and the evaluation application (electronic door lock) are presented.

A. Manufactured ASIC

Fig. 4 depicts the CAD layout view of the evaluation ASIC and a photograph of the die surface of one of the samples, which have been manufactured using the UMC 65 nm low-leakage process. Table I summarizes the key properties of both controller domains. Due to the very compact size of the *NanoController* and its ISA, only a small region in the upper left die corner is reserved, which will be the always-on domain of the system. This includes 64 B of instruction memory (up to 128 instructions), which is generously dimensioned for the evaluated application. Instruction and data memories are implemented as standard cell memories (SCM, flip-flop arrays) due to the area and power inefficiency of SRAM macro blocks at small memory capacities. It should be noted that most of the *NanoController* domain is occupied by IO cells for evaluation and debugging (external clock, reset, SPI and GPIO interfaces) in this ASIC prototype, which would not be required in a final product SoC integration. The actual contributing logic and memory area is below 0.01 mm^2 and thus *very small*. Already the core logic (without any memories) of the GPC is 2.6x larger in this implementation.

TABLE I
KEY PROPERTIES OF EVALUATION ASIC AND CONTROLLER DOMAINS

	NanoController		32-bit GPC	
Silicon Area [mm^2]	*Total Die:* 3.2674 mm^2 (100.00 %)			
- Physical (incl. IO)	0.1876	5.74 %	3.0798	94.26 %
- Core Logic	0.0018	0.05 %	0.0260	0.80 %
- Memory Cells	0.0081	0.25 %	1.7876	54.71 %
On-Chip Memory				
- Instruction	64 B	SCM	96 KiB	SRAM
- Data	16 B	SCM	128 KiB	SRAM
Power Supply	nom. 1.2 V on		nom. 1.2 V on/off	

The majority of the die area is covered by the GPC implementation, the memories and peripheral interfaces (SPI, I2C, UART, GPIO). Large SRAM macro blocks for instruction and data memory provide enough headroom for general processing and communication parts within the application. The GPC on/off domain is supposed to be in power shut-down by default and is only activated processing complex infrequent events. SRAM implementation of the memories is a design compromise made due to the participation in a resource-limited multi-project wafer run for fabrication, which had no preferred non-volatile memory IP available. Therefore, a hardware bootloader is implemented to load required pages of program and data memory on demand from an external SPI Flash IC after each power-up. However, this is not a weakness of the system concept and only caused by the restricted fabrication preliminaries.

B. Application

The evaluation application used in this paper is an RFID tag-based electronic door lock application from a project dealing with a processing platform for home automation devices powered by EH [2]. Fig. 5 shows the flow diagram and the partitioning onto the two processor cores. The control tasks mapped to the *NanoController* are

- *proximity event detection* in order to detect the presence of an RFID transponder key card via a capacitive sensor,
- *real-time clock* (RTC) for time-of-day-based access permissions and to detect RFID *time-out conditions*,
- and *power-off request* from the GPC domain after successful RFID communication.

Dashed lines in Fig. 5 represent the actions that cause power-up or power-down of the GPC under program control of the *NanoController* via the power management. If an attempt to lock or unlock a door with a key card is detected by the proximity sensor, the GPC will be supplied with power to communicate with the RFID card (AES-128-encrypted).

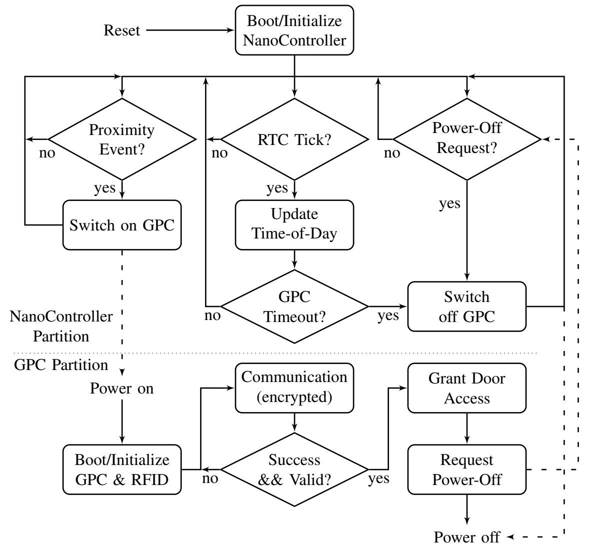

Fig. 5. Flow diagram of the electronic door lock evaluation application

After successful read-out and door access, or after failure and time-out, the GPC power domain will be switched off again. RFID communication with a key card is expected to happen only few times a day, so that the average power consumption of the system will be dominated by the energy-efficient *NanoController* and its application partition. A typical requirement for office doors, as defined in the aforementioned project, is an average of five lock cycles per work day [2].

V. Results and Comparison

With the setup[1] described in Section IV, power measurements on the *NanoController* domain result in a dynamic power consumption of 3.6 µW MHz^{-1}, measured at 6.25 MHz and room temperature, and less than 0.2 µW static leakage power. These are worst-case values, taking measurement inaccuracy of the equipment into account (3.1 % maximum relative error ±0.2 µW). For the evaluated application, a clock frequency of 32 kHz is sufficient for the *NanoController* partition, which is also a reference value for RTCs in always-on power domains. This results in 0.12 µW dynamic power and, consequently, maximum 0.32 µW total power consumption, which is the always-on power dominating the average power consumption of the processor system.

For the GPC domain, values of 40 µW MHz^{-1} dynamic power, measured at 6.25 MHz, and 80 µW static leakage power consumption are obtained when the core is powered up. Static leakage is considerably high due to the large total amount of 224 KiB generic, non-power-gated SRAM macro IP used in this domain, which is confirmed by data sheet values from the IP vendor. Compared to this naïve memory implementation approach limited by the multi-project wafer run options, it is expected that static leakage can be reduced to the order of magnitude of commercial microcontrollers (0.5 to 4 µW) when using properly dimensioned non-volatile memories, e.g., integrated low-power Flash for the instruction memory. However, with five lock cycles per work day on average [2], the TTA domain is active for maximum 10 s per day, so that the high leakage can be neglected for this application case. In fact, the resulting total average power of the heterogeneous processor system is 0.35 µW, confirming the previous assumption that the average system power will be dominated by the power consumption of the *NanoController* domain.

In Table II, the measurement results are put into context with data sheet values and references from Section II. While the technologies and architectures are too diverse to compare them in detail, the table shows a typical range of 0.5 to 4 µW for the always-on power consumption where at least an internal watch-dog timer is running and the CPU is shut down in sleep mode. The *NanoController* leakage of less than 0.2 µW is less than for all presented implementations except [15], for which it is in a comparable region, however, a more precise evaluation could not be performed due to measurement inaccuracy. Also, the dynamic power consumption of 3.6 µW MHz^{-1} outperforms the reference values or approaches the very small numbers of state-of-the-art near-threshold operation [16], which has been out of scope of this work. Consequently, *NanoController* code can be executed with a clock frequency of 0.08 to 1.06 MHz without leaving the always-on power budget of the references, which are not able to perform any computations in sleep mode. For the specific door lock application described in Section IV-B, the reference implementation on an AVR ATmega1284 microcontroller [8] consumed 7 µW on average in measurements, which is 7x more than the sleep mode consumption due to frequent periodical wake-up [2]. With the presented heterogeneous system, the total average power consumption of 0.35 µW for the same application increases only by 9.3 % compared to the always-on *NanoController* domain. The total power consumption is 20x less than the AVR reference, because frequent wake-up of the GPC is avoided.

VI. Conclusion

A silicon-proof heterogeneous controller system has been presented in order to enable more efficient and flexible power management strategies for platforms powered entirely by energy harvesting. Instead of a periodical timer, the system uses a very small and energy-efficient programmable architecture, the *NanoController*, in the always-on power domain, whose logic requires *only 220 logic gates* (UMC 65 nm standard cell technology). A larger, general-purpose controller (GPC) is power-gated completely to eliminate any power consumption when inactive, and is only enabled for infrequent events of complex processing, based on intelligent control decisions made by the *NanoController*. Therefore, as opposed to state-of-the-art IoT devices, data pre-processing, application control, and intelligent power management control, e.g., based on specific sensor data or modeled available energy, can be performed without periodical power-up of the GPC from sleep.

The presented system has been implemented and manufactured as a standard cell ASIC using the UMC 65 nm low-

[1] Demonstration video of concept & measurement setup: *Nanocontroller* – `https://youtu.be/jElXvPlH-O4`

TABLE II
COMPARISON OF ASIC IMPLEMENTATION RESULTS TO COMMERCIAL AND PUBLISHED LOW-POWER CONTROLLER IMPLEMENTATIONS

Microcontroller	Technology	Core Voltage	Architecture	Power Consumption	
				Norm. Dyn.	*Always-On / Sleep incl. Leakage*
SiLabs EFM32 [10]	-	1.8 V	32 bit ARM Cortex M4	>110 µW MHz^{-1}	>2 µW
Renesas RE01 [11]	65 nm SOTB	1.8 V	32 bit ARM Cortex M0+	550 µW MHz^{-1}	2 µW
ST STM32L432 [12]	90 nm	1.8 V	32 bit ARM Cortex M4	200 µW MHz^{-1}	0.8 µW
TI MSP430L092 [9]	130 nm Low-Leakage	0.9 V	16 bit Multi-Cycle	50 µW MHz^{-1}	4 µW
Atmel/Microchip ATmega1284 [8]	-	1.8 V	8 bit AVR RISC	720 µW MHz^{-1}	1 µW
Microchip PIC12LF1840T39A [7]	-	1.8 V	8 bit accumulator, 14 bit RISC instructions	80 µW MHz^{-1}	1 µW
EnOcean Dolphin V4 [4]	-	1.8 V	8 bit Intel 8051	430 µW MHz^{-1}	1.3 µW
EM EM6682 [6]	-	0.9 V	4 bit RISC	125 µW MHz^{-1}	0.5 µW
Kwong 2009 [14]	65 nm	0.5 V	16 bit MSP430	>27 µW MHz^{-1}	1 µW
Bol 2013 [13]	65 nm Low-Power	~0.4 V	16 bit MSP430	>7 µW MHz^{-1}	1.7 µW
Myers 2015 [15]	65 nm Low-Leakage	~0.4 V	32 bit ARM Cortex M0+	>12 µW MHz^{-1}	0.08 µW
Lallement 2018 [16]	28 nm FDSOI	0.5 V	32 bit ARM Cortex M0+	>2.7 µW MHz^{-1}	0.7 µW
NanoController (this work)	65 nm Low-Leakage	1.2 V	8 bit accumulator, 4 bit multi-cycle instructions	**3.6 µW MHz^{-1}** **Combined <0.32 µW at 32 kHz**	**Leakage ≪0.2 µW**
32-bit GPC (this work)	65 nm Low-Leakage	1.2 V	32 bit based on TTA	40 µW MHz^{-1}	80 µW (naïve SRAM implementation, optimization not scope of this work)

leakage process. In a real home automation application, i.e., electronic door lock, power measurements show 20x less power consumption than a reference implementation on an AVR microcontroller. The measured average power consumption on first silicon is maximum 0.35 µW, which is a promising result with respect to the limited power budget of energy harvesting platforms. In order to increase the potential of the *NanoController* concept, refinements on the ISA and the instruction compaction for even less memory (and, therefore, area and power) requirements, as well as an extended evaluation of further application cases requiring pre-processing of sensor samples, will be the scope of upcoming work.

REFERENCES

[1] F. Ait Aoudia, M. Gautier, M. Magno, O. Berder, and L. Benini, "Leveraging energy harvesting and wake-up receivers for long-term wireless sensor networks," *Sensors*, vol. 18, no. 5, 2018.

[2] M. Neujahr, S. Möller, J. Passoke, and H. Blume, "Flexible Plattform für Energiesammelsysteme für die Gebäudeautomation - MEH *(project report in German language)*," Tech. Rep., 2021, in acquisition, accessible via tib.eu.

[3] M. Gorlatova, J. Sarik, G. Grebla, M. Cong, I. Kymissis, and G. Zussman, "Movers and shakers: Kinetic energy harvesting for the internet of things," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1624–1639, 2015.

[4] EnOcean, "DOLPHIN V4 core description," https://www.enocean.com/wp-content/uploads/Knowledge-Base/Dolphin_V4_Core_Description_V2.1.pdf, 2014, accessed on 12.05.2022.

[5] P. Pickering, "Designing ultra-low-power sensor nodes for IoT applications," https://www.electronicdesign.com/power-management/article/21802213/designing-ultralowpower-sensor-nodes-for-iot-applications, 2016, accessed on 12.05.2022.

[6] EM Microelectronic-Marin, "EM6682: World's first sub-1V, 8-pin microcontroller," https://www.emmicroelectronic.com/sites/default/files/products/datasheets/em6682_fs.pdf, 2008, accessed on 12.05.2022.

[7] Microchip, "PIC12LF1840T39A 8-bit Flash microcontroller with XLP technology datasheet," https://ww1.microchip.com/downloads/en/DeviceDoc/40001636B.pdf, 2014, accessed on 12.05.2022.

[8] ——, "Atmega1284 datasheet," http://ww1.microchip.com/downloads/en/devicedoc/atmel-42718-atmega1284_datasheet.pdf, 2016, accessed on 12.05.2022.

[9] Texas Instruments, "MSP430L092 mixed signal microcontroller datasheet," https://www.ti.com/lit/ds/symlink/msp430l092.pdf, 2010, accessed on 12.05.2022.

[10] Silicon Labs, "EFM32 MCU overview," https://www.silabs.com/mcu/32-bit, 2021, accessed on 12.05.2022.

[11] Renesas, "RE01B group product with 1.5-Mbyte Flash memory," https://www.renesas.com/eu/en/document/dst/re01b-group-product-15-mbyte-flash-memory-datasheet, 2020, accessed on 12.05.2022.

[12] ST Micro, "STM32L432 ultra-low-power ARM Cortex-M4 32-bit MCU (DS11451 Rev 4)," https://www.st.com/resource/en/datasheet/stm32l432kb.pdf, 2018, accessed on 12.05.2022.

[13] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J.-D. Legat, "SleepWalker: A 25-MHz 0.4-V sub-mm^2 7-μW/MHz microcontroller in 65-nm LP/GP CMOS for low-carbon wireless sensor nodes," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 20–32, 2013.

[14] J. Kwong, Y. K. Ramadass, N. Verma, and A. P. Chandrakasan, "A 65 nm sub-V_t microcontroller with integrated SRAM and switched capacitor DC-DC converter," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 115–126, 2009.

[15] J. Myers, A. Savanth, R. Gaddh, D. Howard, P. Prabhat, and D. Flynn, "A subthreshold ARM Cortex-M0+ subsystem in 65 nm CMOS for WSN applications with 14 power domains, 10T SRAM, and integrated voltage regulator," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 31–44, 2016.

[16] G. Lallement, F. Abouzeid, M. Cochet, J.-M. Daveau, P. Roche, and J.-L. Autran, "A 2.7 pJ/cycle 16 MHz, 0.7 μW deep sleep power ARM Cortex-M0+ core SoC in 28 nm FD-SOI," *IEEE J. Solid-State Circuits*, vol. 53, no. 7, pp. 2088–2100, 2018.

[17] M. Weißbrich, J. A. Moreno-Medina, and G. Payá-Vayá, "Using genetic algorithms to optimize the instruction-set encoding on processor cores," in *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2021, pp. 1–6.

[18] P. Jääskeläinen, T. Viitanen, J. Takala, and H. Berg, *HW/SW Co-design Toolset for Customization of Exposed Datapath Processors*. Springer International Publishing, 2017, pp. 147–164. [Online]. Available: https://doi.org/10.1007/978-3-319-49679-5_8

[19] H. Kultala, T. Viitanen, H. Berg, P. Jääskeläinen, J. Multanen, M. Kokkonen, K. Raiskila, T. Zetterman, and J. Takala, "LordCore: Energy-efficient OpenCL-programmable software-defined radio coprocessor," *IEEE Trans. VLSI Syst.*, vol. 27, no. 5, pp. 1029–1042, 2019.

[20] J. Multanen, H. Kultala, K. Tervo, and P. Jääskeläinen, "Energy efficient low latency multi-issue cores for intelligent always-on IoT applications," *Journal of Signal Processing Systems*, vol. 92, no. 10, pp. 1057–1073, 2020.