An Energy Consumption Benchmark for a Low-Power RISC-V Core Aimed at Implantable Medical Devices

Roberto Molina-Robles, *Graduate Student Member, IEEE*, Alfredo Arnaud, *Senior Member, IEEE*, Matías Miguez, *Member, IEEE*, Joel Gak, *Member, IEEE*, Alfonso Chacón-Rodríguez, *Senior Member, IEEE*, and Ronny García-Ramírez, *Member, IEEE*

***Abstract*—In this work, Siwa, a micropower 32-bits RISC-V core aimed at implantable medical SoCs is presented. The core was fabricated in a 180-nm CMOS-HV technology to directly drive biological stimuli circuits within the same ASIC. A complete set of power consumption measurements is presented; the core properly operated up to 30 MHz with a current consumption of 52 μA/MHz at 1.8-V supply voltage, and <20-nA leakages at room temperature. Since the existing benchmarks are not completely adequate to compare Siwa performance to other microcontrollers used on implantable medical devices (IMDs), a simple, specific benchmark inspired in a pacemaker operation was developed. The new benchmark considers both the CPU current consumption and performance, sleep and run states, and allows to compare broadly different CPUs and operating conditions for specific IMD applications. Siwa's performance was compared using this benchmark with 8 and 16-bits MCUs.**

***Index Terms*—CPU benchmark, medical devices, RISC-V.**

I. Introduction

IMPLANTABLE medical devices (IMDs) are one of the most challenging embedded systems due to the low-current consumption constraints and reliability and safety issues. A block diagram of a generic IMD is shown in Fig. 1, including electrodes connecting the biological tissue to the device through one or multiple pass switches, sensors, amplifiers, and filters for the signal conditioning of body's natural electrical activity, an intelligent control logic (CPU) processing data and deciding when and for how long a stimulus should be applied to the tissue, a telemetry block communicating data out when necessary, and a stimuli section composed of voltage and/or current sources. In Fig. 1, an off-the-shelf microcontroller can be utilized either combined or not with an analog ASIC for sensing/stimuli functions [1], [2], but multiple chips unnecessarily increase the power consumption, PCB size, and per-IMD production cost. Thus, integrating the CPU and specific analog circuits in the same HV-CMOS technology voltage, because up to 25 V can be necessary for tissue stimulation) is a highly desirable option. In the case of mature products, such as pacemakers or cochlear implants and large IMD manufacturers, both the CPU and sensing/stimuli circuits are integrated in the same SoC. But IMDs are very low-volume production devices and this characteristic usually limits the development of a custom CPUs. In this context, an opensource scalable processor like RISC-V would be helpful to allow, especially small companies and research groups, to develop their own IMD-SoCs. While a 32-bits CPU may sound complex or complex or power hungry, recent advances allow to implement very efficient RISC-V controllers [3], [4], but their application to IMDs is yet underexplored. In [4] and [5], a power-efficient core was presented aimed at medical applications, removing unnecessary features, such as floating point capabilities, predictive execution modes, superscalar pipelines, multilevel cache and/or DMA, etc. As pointed by Strydis *et al.* [6], this kind of design strategy enhances reliability and makes simpler to make the system safe. But it is still hard to answer if an RISC-V core is efficient enough in terms of current consumption, or die area, or if just an 8-bit processor is a better option. Furthermore, there is a lack of specific CPU benchmarks in the case of IMDs to compare different cores. In this work, the first fabricated version of the core in [4] named *Siwa*, utilizing LV transistors in a 0.18μm HV-CMOS technology is presented, including several current consumption measurements. The ASIC integrates a custom V_{DD} = 1.8-V RISC-V core, 8-kB SRAM, and an 8-bit programmable +6 trimmable

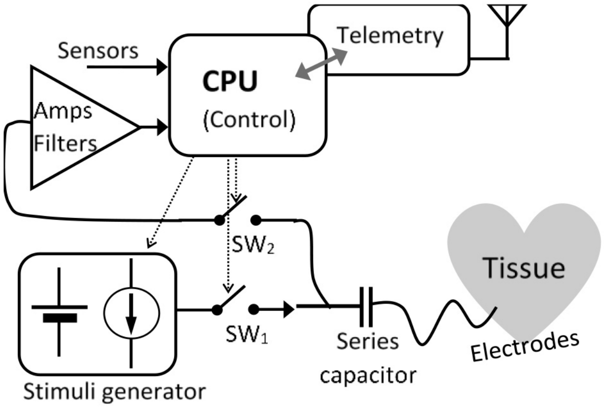

Fig. 1. Simplified block diagram of an IMD, including CPU, telemetry, stimuli outputs, electrode and sensors inputs, and series safety capacitor [1].

Manuscript received 15 March 2022; revised 13 June 2022; accepted 5 July 2022. Date of publication 12 July 2022; date of current version 29 May 2023. This work was supported in part by the Agencia Nacional de Investigación e Innovación (ANII)-Uruguay under Grant FMV_1_2017_136543, and in part by the Universidad Católica del Uruguay (UCU) and Tecnológico de Costa Rica (TEC). This manuscript was recommended for publication by L. De Micco. *(Corresponding author: Roberto Molina-Robles.)*

Roberto Molina-Robles, Alfonso Chacón-Rodríguez, and Ronny García-Ramírez are with the Escuela de Ingeniería Electrónica, Instituto Tecnológico de Costa Rica, Cartago 30101, Costa Rica (e-mail: rmolina@itcr.ac.c).

Alfredo Arnaud, Matías Miguez, and Joel Gak are with the Departamento de Ingeniería, Universidad Católica del Uruguay, Montevideo 11600, Uruguay.

Digital Object Identifier 10.1109/LES.2022.3190063

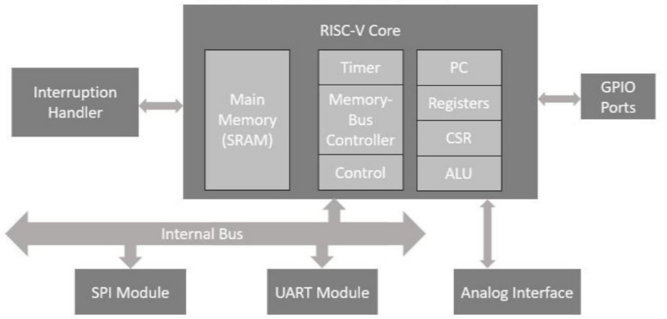

Fig. 2. Siwa RISC-V CPU organization.

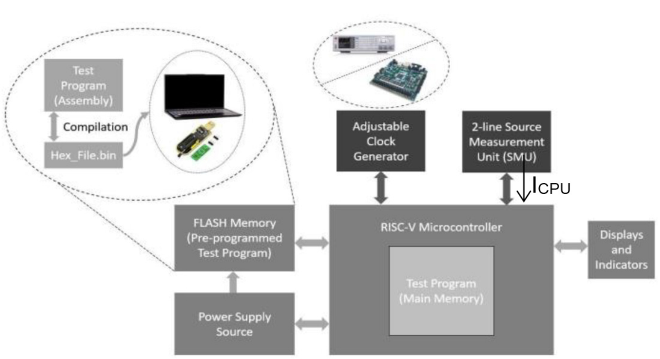
Test Program (Assembly)
Compilation
Hex_File.bin
Adjustable Clock Generator
2-line Source Measurement Unit (SMU)
ICPU
FLASH Memory (Pre-programmed Test Program)
RISC-V Microcontroller
Test Program (Main Memory)
Displays and Indicators
Power Supply Source

Fig. 3. Scheme of Siwa measurement setup.

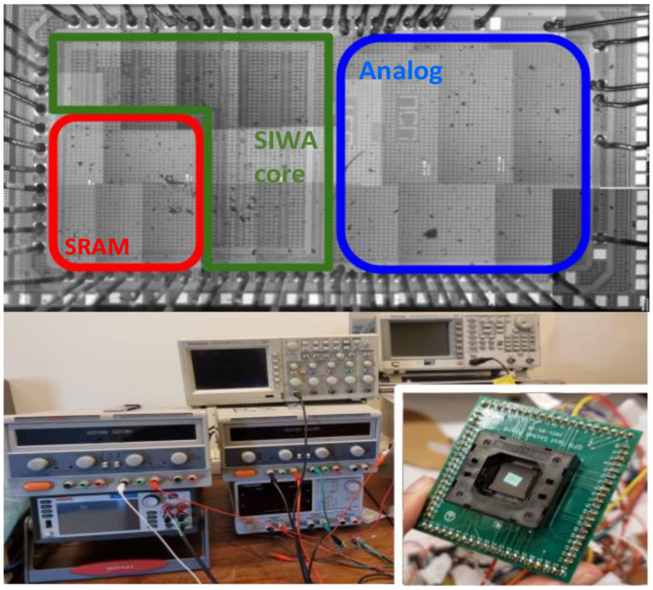

Fig. 4. Siwa microphotograph (top) sized 3 mm × 1.5 mm including pads; SRAM, RISC-V core, and analog circuits indicated in red, green, and blue, respectively. Testbench and QFN80 DUT in a socket (bottom).

current source aimed at biological tissue stimulation up to 25 mA–15 V among other analog circuits in the HV domain. For simplicity, analog In this section, the developed RISC-V core is introduced and, next, a complete set of power consumption measurements is presented. Finally, inspired in the pacemaker, a specific benchmark is developed aimed at comparing very different cores for IMDs.

A. Siwa CPU Architecture

The developed processor is a full-custom RISC-RV32I-based architecture, designed for low-power consumption with a concise and simple organization, but can execute the complete RV32I Instruction Set Architecture (ISA). Siwa organization is presented in Fig. 2. The CPU has a 32-bit register bank and an integrated 8-kB SRAM memory, 3.3-V digital I/O ports to drive the analog circuit blocks, and several types of interrupts and timers as necessary to implement the IMDs, UART, and SPI ports. The SPI is used to boot up the core upon reset, loading the main program from an external Flash memory. A pipelined or superscalar implementation was avoided for the sake of area and power consumption reduction. Siwa was verified using a UVM framework and fully tested on a FPGA. The ASIC version was implemented using low-power library cells provided by the foundry. The total cell count was 11.8k gates. Siwa has a sleep mode, where it turns off the internal bus, SPI and UART modules, but a programmed timer is still running to wake up the CPU with an interrupt call. The sleep mode is important for a normal IMD operation and will be addressed in the developed benchmark.

B. CPU Benchmarks

A benchmark is a kind of report to compare different CPUs. It is generally a set of programs running in a device under test (DUT) with different tasks to get a score. Dhrystone and Whetstone [7] are classic synthetic benchmarks using integer and floating-point calculations; *synthetic* means that they are not real applications but a set of processes representing the CPU usage. But these classic benchmarks are very limited and hardly fit an IMD's characteristic. CoreMark is probably the best known modern benchmark for embedded applications incorporating a broader realistic set of tasks. ULPMark is a low-power version for MCU energy analysis [8]. The latter is suitable for battery-powered IoT devices approaching the needs of IMDs. These benchmarks are generic and there are many examples of specific benchmarks for wireless, satellite, MCUs under radiation effects, ultralow-power machine learning, among others [9]–[11] that yield more detailed results and conclusions. There are almost no references, however, for CPU aimed at implantable applications, which was the objective of this work. An ideal benchmark for IMDs should be energy-performance oriented and compiler-independent, should adequately represent a specific IMD (e.g., a pacemaker) but easily scalable to new applications.

II. Siwa Measurement Results

The measurement setup to characterize Siwa is shown in Figs. 3 and 4. Benchmark programs are compiled using the RISC-V opensource toolchain [12] (assembly, C) and stored in an external boot Flash memory; a Keithley2400 SMU is used to separately power the RISC-V core including memory and measure the current consumption, and an independent supply powers auxiliary and analog circuits. A function generator and an FPGA are employed to generate a variable frequency main clock f_{clk} from a few Hz to 30 MHz. All the measurements were performed at $V_{DD} = 1.8$-V supply voltage. The first Siwa measurements were on power consumption while varying f_{clk}, and for different instructions. The results are presented in Fig. 5 for the main functional groups: arithmetic/logic, control, and memory instructions. The power consumption was measured separately for most ISA instructions with a simple program containing ∼1024 times the same instruction and a final loop jump. The results are averaged for each group in Fig. 5. Additionally, Fig. 5(a) includes power consumption when not booted, and Fig. 5(b) includes the power consumption of

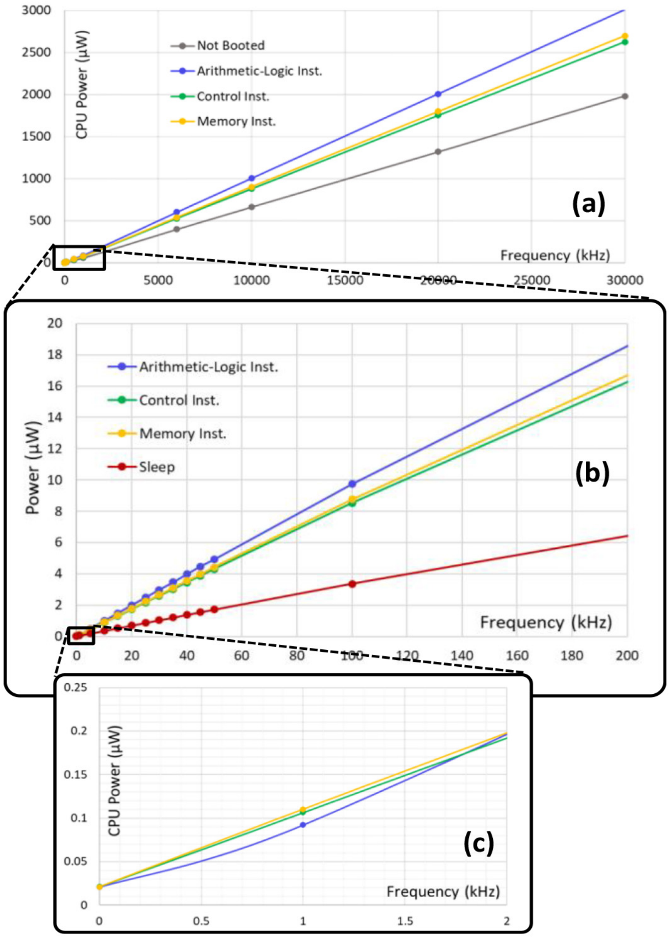

Fig. 5. Measured average power consumption of Siwa core, for different instruction types while varying the clock frequency (a) up to 30 MHz, (b) zoomed-in view from dc to 200 kHz, and (c) zoomed-in view of only 2 kHz.

TABLE I
NORMALIZED SIWA POWER

Data Set	Avr. CPI	Normalized Power
Arithmetic-Logic Inst.	13.9	1.39 mW/MIPS
Control Inst.	13.1	1.32 mW/MIPS
Memory Inst.	10	1.00 mW/MIPS

Siwa's sleep mode as described in Section I-B. Note the power consumption is almost linear with f_{clk} from dc to 30 MHz with an average 94-μW/MHz slope. Note also the very low leakage as $f_{\text{clk}}\rightarrow$dc with a core static power consumption of 20 nW in Fig. 5(c). Siwa was not optimized in performance; being a 32-bits core many instructions take several clock cycles. Thus, power measurements were normalized to Million-Instructions per second (MIPS) in Table I to better represent the core performance. MIPS $= f_{\text{clk}}/(\text{CPI} \times 10^6)$ where CPI is the number of clock cycles per instruction. A simulated CoreMark benchmark resulted in 0.2 CoreMark/MHz, a lower result in comparison to the survey in [13] of RISC-V simple cores but using advanced technologies.

III. Specific Benchmark Aimed at IMDs

Active IMDs can be very different from a basic old pacemaker to complex vision aids yet under development. To develop the CPU core benchmark, the pacemaker case was studied and adapted to different IMDs. A basic pacemaker essentially steadily listens to the natural electrical activity of the heart through implanted electrodes, and only if it detects an abnormal activity does it electrically stimulate to regulate the heartbeat normally using the same electrodes. A pacemaker stimulus consists of a single-voltage pulse of amplitude V_{pace} and duration t_{pace}, and a passive balance stage a few ms later [1], [16]. Rate-adaptive pacemakers adapt their pacing rate to the patient's physical activity using at least an always-on micro-power accelerometer [14]. Modern pacemakers also incorporate additional functions, such as ICD capabilities, intracavital ECG recording option, and complex telemetry functions among others [1], [14]–[17]. All these features fit the scheme in Fig. 1. Some aspects to consider while developing the benchmark.

1) A pacemaker behaves like a state machine, the CPU is most of the time in "sleep" either running from a low frequency clock or directly halted and waiting for an interrupt. Upon an external or a timer interrupt event, the pacemaker state changes, and the CPU executes as fast as possible (to reduce latency) several simple tasks, such as triggering a pace, programming timers (pace, charge balance, and wait times), get accelerometer data or a single electrode's impedance, and returns to sleep again to reduce battery consumption.
2) No complex calculations are required permanently from an IMD's CPU; always-on complex operations if required, are generally implemented either as analog or digital custom circuits [14], [15]. Thus, complex multipliers and floating-point math will be avoided in the proposed benchmark.
3) Other high-level functions, such as telemetry or ECG vector processing are rarely executed, e.g., during a visit to the cardiologist, and have much less impact in the battery life.- Pacemakers are normally powered by a primary lithium iodine cell ranging from 2.8 to 2.0 V (beginning to end of battery life). *Battery current consumption* instead of power will be used in the benchmark since in most of the cases, the CPU is connected using an LDO instead of a buck converter because of simplicity, reliability, and because the efficiency in the μW range.

To develop a specific benchmark, consider the pacemaker's average base current I_{base} necessary just to stimulate the heart's muscle with a single electrode in a reference pacing condition [17] with $f_{\text{pace}} = 60$ ppm $=$ 1-Hz pacing frequency, $V_{\text{pace}} = 5$ V, $t_{\text{pace}} = 1$ ms, and a reference electrode impedance $R_{\text{load}} = 500\ \Omega$

$$\overline{I_{\text{base}}} \approx f_{\text{pace}} \frac{2\cdot V_{\text{pace}}\cdot t_{\text{pace}}}{R_{\text{load}}}\Bigg|_{\substack{V_{\text{pace}}=5\text{ V}, t_{\text{pace}}=1\text{ ms}\\ f_{\text{pace}}=1\text{ Hz},\ R_{\text{load}}=500\ \Omega}} = 20\ \mu\text{A}. \tag{1}$$

In (1) factor 2 is included, as a charge pump is necessary to generate a 5-V pulse from the 2.7 V_{typ} battery [15], [16]. A "Basic Task Set" (BTS) was defined assuming like in Fig. 6 that the CPU is most of the time sleep mode. The BTS contains a set of pseudo-assembler instructions corresponding to a complete interrupt code from the sleep-to-sleep condition. Examining a previously developed pacemaker firmware, the selected BTS code includes: 8-b register read and write operations, 32-b sums and 16-b compare, memory operations (LUT and data memory read and write), and conditional jumps. Note the operations in Fig. 6 are defined with a CPU-independent

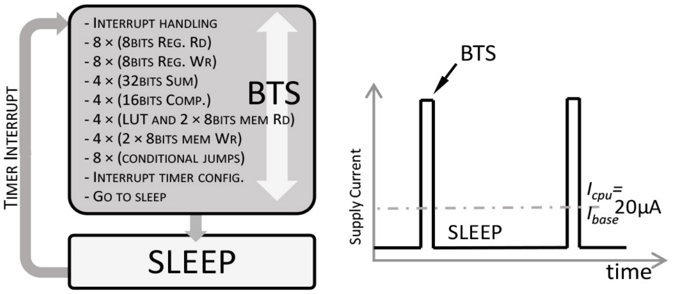

Fig. 6. Scheme of the developed benchmark. The DUT continuously run a (Basic-Task-Set + sleep) cycle until the average current consumption is equal to a base current specific to the device (20 μA in the case of a pacemaker). The BTS is a CPU-independent pseudo assembler code.

TABLE II
MICROCONTROLLER BENCHMARK COMPARISON

	Siwa	MSP430	ATmega328P
Processor bits	32	16	8
BTS freq. [MHz]	20	1	1
Supply Voltage [V]	1.8	1.8	2.2
Sleep clck [kHz]	32	32	32
Sleep current [µA]	0.63	0.5	4.3
BM_{IMD} [BTS/s]	681	438	162

number of bits; an 8-b CPU probably will require extra instructions to execute a 16-b comparison but a 32-b Siwa has a larger CPI. Finally, the benchmark BM_{IMD} aimed at micropower IMDs, is defined like in Fig. 6, as the number of BTS per second the CPU can execute @$I_{\text{CPU}} = 20\mu$A while running a BTS→sleep continuous cycle using its own timer to wake-up. Note this figure of merit is almost independent of the active CPU clock, the bus width in bits, the sleep strategy or sleep clock, interrupt response time, or the instruction set, thus allowing to compare very different cores. In the case of the measured version of Siwa, $BM_{\text{IMD}} = 681$ BTS/s @$V_{DD} = 1.8$ V, and $f_{\text{clk}} = 20$ MHz. Table II shows a comparison to two low-power of-the-shelf MCUs.

The proposed BTS/s-benchmark is core-independent, but it is application centered; thus, the BTS definition is of course rather arbitrary. The benchmark is a tool for the IMD designer, and must be adapted to different IMDs following the procedure.

1) Estimate a base current consumption I_{base} for the IMD like in (1) or using critical analog circuits consumption instead.
2) Define a BTS for a representative IMD interrupt code.
3) Develop a testbench to measure the core current consumption I_{CPU} with a realistic V_{DD}, and program the core to alternatively execute the BTS → sleep → BTS → sleep cycle.
4) Adjust an internal timer to periodically wake-up the CPU, and tune t_{sleep} so that the average CPU current is $I_{\text{CPU}} = I_{\text{base}}$.
5) At this point, calculate the BM_{IMD} benchmark as the number of (BTS → sleep) cycles per second.

IV. CONCLUSION

In this work, the first fabricated version of Siwa, a micropower 32-bits RISC-V core in a 180-nm CMOS process with HV capabilities as necessary to implement IMDs, was presented. The core was measured from dc up to a 30-MHz clock frequency showing a very low leakage $I_{\text{leak}} = 12$ nA at room temperature, and an average current consumption $I_{\text{CPU}} = 52\mu$A/MHz. The core uses approximately 11.8k gates with an area of 0.7 mm^2. The occupied area (considering the technology scaling), and CoreMark results are consistent with other power-constrained RISC-V cores [13]. To compare Siwa performance with other cores and microcontrollers used on IMDs, a specific benchmark, named BM_{IMD}, was developed. The new benchmark considers the CPU executing a continuous (BTS → sleep) loop cycle, in comparison to a specific IMD's base current I_{base}, expressed as the number of cycles per second @ $I_{\text{base}} = I_{\text{base}}$, allowing to compare very different cores, and execution conditions. A pacemaker example was presented and Siwa performance was compared with two low-power, of-the-shelf, 8 and 16-b microcontrollers. The results showed that they perform similarly, with a slight advantage for Siwa but that can be attributed to fewer peripherals implemented in it. Back to the original question of how efficient a 32-b processor is for an IMD, Siwa was shown to be comparable to 8 and 16-b processors, and the area overhead if any, is much less than the area expected for analog HV circuits such as reported in [14]–[16] or the single-channel current source in Fig. 4. Thus, since there are plenty free tools and resources, working with the RISC-V is a promising way to allow small companies and research groups to develop their own IMD SoCs.

REFERENCES

[1] D. Prutchi and M. Norris, "Cardiac pacing and defibrillation" in *Design and Development of Medical Electronic Instrumentation*. Hoboken, NJ, USA: Wiley, 2005.

[2] M. Sayahkarajy, E. Supriyanto, M. H. Satria, and H. Samion, "Design of a microcontroller-based artificial pacemaker: An internal pacing device," in *Proc. Int. Conf. Robot. Autom. Sci. (ICORAS)*, 2017, pp. 1–5.

[3] A. Pullini, D. Rossi, I. Loi, G. Tagliavini, and L. Benini, "Mr.Wolf: An energy-precision scalable parallel ultra low power SoC for IoT edge processing," *IEEE J. Solid-State Circuits*, vol. 54, no. 7, pp. 1970–1981, Jul. 2019.

[4] R. Garcia-Ramirez *et al.*, "Siwa: A custom RISC-V based system on chip (SoC) for low power medical applications," *Microelectron. J.*, vol. 98, Apr. 2020, Art. no. 104753.

[5] A. Arnaud *et al.*, "A RISC-V based medical implantable SoC for high voltage and current tissue stimulus," in *Proc. IEEE 11th Latin Amer. Symp. Circuits Syst. (LASCAS)*, 2020, pp. 1–4.

[6] C. Strydis, R. M. Seepers, P. Peris-Lopez, D. Siskos, and I. Sourdis, "A system architecture, processor, and communication protocol for secure implants," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 4, pp. 1–23, 2015.

[7] W. J. Price, "A benchmark tutorial," *IEEE Micro*, vol. 9, no. 5, pp. 28–43, Oct. 1989.

[8] "CoreMark Home." [Online]. Available: https://www.eembc.org/coremark (Accessed: Dec. 5, 2021).

[9] B. Sudharsan *et al.*, "TinyML benchmark: Executing fully connected neural networks on commodity microcontrollers," in *Proc. IEEE 7th World Forum Internet Things (WF-IoT)*, 2021, pp. 883–884.

[10] M. Götz, S. Khriji, R. Chéour, W. Arief, and O. Kanoun, "Benchmarking-based investigation on energy efficiency of low-power microcontrollers," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 7505–7512, Oct. 2020.

[11] K. P. Gnawali, H. M. Quinn, and S. Tragoudas, "Developing benchmarks for radiation testing of microcontroller arithmetic units using ATPG," *IEEE Trans. Nucl. Sci.*, vol. 68, no. 5, pp. 857–864, May 2021.

[12] "RISC-V GNU Toolchain." RISC-V. [Online] Available: https://github.com/riscv-collab/riscv-gnu-toolchain (Accessed: Dec. 5, 2021)

[13] I. Elsadek and E. Y. Tawfik, "RISC-V resource-constrained cores: A survey and energy comparison," in *Proc. 19th IEEE Int. New Circuits Syst. Conf. (NEWCAS)*, 2021, pp. 1–5.

[14] J. Gak, M. R. Miguez, and A. Arnaud, "Nanopower OTAs with improved linearity and low input offset using bulk degeneration," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 3, pp. 689–698, Mar. 2014.

[15] L. S. Y. Wong, S. Hossain, A. Ta, J. Edvinsson, D. H. Rivas, and H. Naa, "A very low-power CMOS mixed-signal IC for implantable pacemaker applications," *IEEE J. Solid-State Circuits*, vol. 32, no. 12, pp. 2446–2456, Dec. 2004.

[16] J. Gak, M. Miguez, and A. Arnaud, "CMOS level shifters from 0 to 18 V output," *Analog Integr. Circuits Signal Process.*, vol. 107, no. 3, pp. 617–628, 2020.

[17] *Active Implantable Medical Devices—Part 2-1 (Cardiac Pacemakers)*, CEN/CENELEC Standard EN 45502-2, 2005.