Siwa: a RISC-V RV32I based Micro-Controller for Implantable Medical Applications

R. Garcia-Ramirez*, A. Chacon-Rodriguez*, R. Castro-Gonzalez *, A. Arnaud†, M. Miguez†, J. Gak†,
R. Molina-Robles*, G. Madrigal-Boza*, M. Oviedo-Hernandez*, E. Solera-Bolanos*, D. Salazar-Sibaja*,
D. Sanchez-Jimenez*, M. Fonseca-Rodriguez*, J. Arrieta-Solorzano*, and R. Rimolo-Donadio*
*Escuela de Ingeniería Electrónica, Instituto Tecnológico de Costa Rica
†Depto. de Ingeniería Eléctrica, Universidad Católica del Uruguay
Email: {rgarcia, alchacon, rmolina, rrimolo}@tec.ac.cr, {aarnaud, mmiguez, jgak}@ucu.edu.uy

***Abstract*—The design of Siwa[1], a compact low power custom system on chip (SoC), targeted for implantable/wearable applications, is reported in this paper. Siwa is based on a RISC-V RV32I architecture. It has a centrally controlled non-pipelined structure, and it includes a control interface for an integrated sensing and stimulation device for biological tissues as well as standard communication interfaces. Siwa was developed from scratch using System Verilog, and implemented in a 180nm CMOS technology; Siwa includes a latch based register file capable to read and write in one clock cycle with an area 30% smaller and a power consumption 25% lower with respect to an equivalent flip flop implementation; also, it has an estimated average power consumption of 70μW (48pJ/cycle) which is comparable to other micro-controllers commonly used in IMD applications.**

***Index Terms*—IMD, RISC-V, Micro-Architecture, System-on-Chip, Digital VLSI.**

I. Introduction

The requirements of small form factor and reliability imposed by implantable medical devices (IMD) applications are hard to achieve using field programmable arrays (FPGAs). Also, these have worse power consumption and performance compared to an application specific integrated circuit (ASIC) using the same technology, and normally additional discrete circuits are required for the analog actuators and sensors. The use of discrete commercial micro-controllers is common in implantable medical applications like pacemakers. However, using independent processing, sensors, and actuators tied to a printed circuit board (PCB) requires extra validation for the final product, as well as a higher form factor and power consumption due to the additional capacitance required to drive the signals out of the ICs through the PCB. This additional power consumption and validation effort can be minimized if both the processing unit and the actuators reside in the same integrated circuit.

Even though there are plenty of micro-controller intellectual property (IP) available for the development of ASICs, the use of general purpose IP adds overhead, since in order to accommodate multiple fields of application in one general IP, additional or oversized structures are often included.

[1]Broadly meaning wisdom in Cabecar, an indigenous language still spoken in Costa Rica

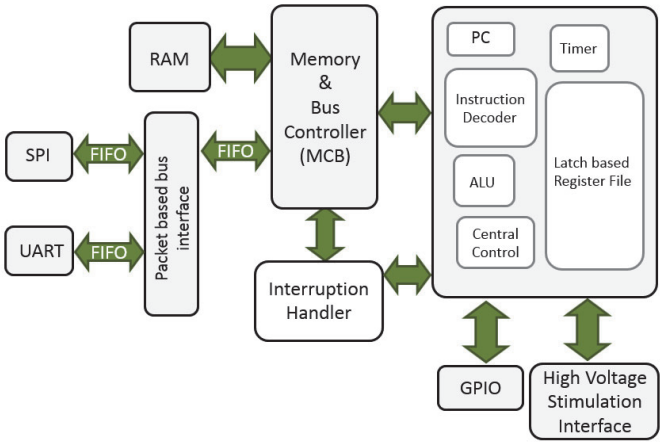

Fig. 1. High-level description of Siwa. The dedicated high-voltage I/O is used as interface with the analog tissue stimulus block of the SoC. An elastic, packet based bus adds flexibility and fault tolerance to communications with the core.

Also, the use of these blocks is tied to restrictive and expensive licenses and generic IPs are not open for customization, making them unsuitable for small development teams with restricted design targets and budgets.

In this paper, the micro-architecture of a custom low power RISC-V [1] based micro-controller, called Siwa, is presented. The selection of RISC-V over other instruction set architectures (ISA) options was done based on its modularity, which allows to implement only the core integer functionalities while still being compliant with the standard; its open philosophy allows for the free development of hardware without any royalties associated to its use, and its fast growth and support from mayor players in the industry envision a strong technological ecosystem in the future.

Since the micro-controller is intended to serve as programmable controller core (PCC) for biological tissue stimulators, a high voltage, reliable process such as XFAB's 0.18 μm CMOS was selected. The use of SystemVerilog (SV) for the RTL description of the micro-controller was chosen based on its finer-grain control over micro-architectural details, against higher abstraction languages such as Chisel or Bluespec SystemVerilog (BSV).

978-1-7281-3427-7/20/$31.00 ©2020 IEEE

This paper is organized as follows: section II provides an overview of the functional details of the proposed architecture, as well as a description of its micro-architecture; section III presents a comparison of the proposed micro-architecture with regard to other RISC-V implementations and other processors commonly used in IMDs; finally, section IV gives some remarks of this work, as well as some prospects of its further development.

II. Siwa's Proposed Micro-Architecture

Figure 1 depicts the main functional blocks of Siwa. The system bus is parameterizable at the RTL level, in the number of devices connected to the bus, the number of bits transferred per transaction, and the depth of the FIFO interfaces; this enables the easy addition or removal of peripherals for future IMD applications on the proposed micro-architecture. The bus is package based and capable of broadcasting messages, allowing for the use of snooping protocols and data redundancy for fault tolerance in critical applications. The arbitration method implemented in the bus is "round robin", in order to ensure fair usage of the bus and avoid starvation of the devices. The bus and its peripherals can be disabled using a control and status register (CSR) bit to enable power savings in master-slave applications.

All the peripherals are managed as memory mapped devices (MMIO), meaning that the CPU communicates to each device by writing to specific memory addresses assigned to the desired peripheral; however, in order to avoid power hungry memory polling on slow devices, whenever a device send a message to the central processing unit (CPU), it triggers an interrupt. The control interface of the biological tissue stimulation device is implemented using custom CSR, interfaced with level shifters to convert between the power domain of the PCC and the one of the analog front end. The stimulation interface is not connected to the system bus to avoid the overhead of the interruption handler, since this is the main peripheral for the target application, and frequent communication is expected with this device.

There are two sources of non-maskable interrupts in the system: those generated by an invalid instruction in the instruction decoder and those generated by an invalid address transaction; there are three sources of maskable interrupts: those coming from external devices connected to the bus, the ones from the internal timer, and those coming from the external interruptions, which can be triggered by and external pin.

The CPU consist of four general blocks: the memory and bus controller (MBC), an instruction decoder(ID), the register file (RF), and a arithmetic and logic unit (ALU); these blocks are coordinated by a central finite state machine (FSM). The system also includes standard communication interfaces like a universal asynchronous receiver/transmitter /UART, general purpose input/output ports (GPIO), and a serial peripheral interface (SPI). A centrally controlled micro-architecture was chosen because of its smaller footprint, and thus, lower energy requirements. This entails no segmentation

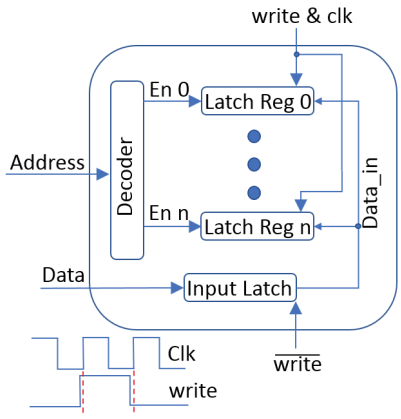

Fig. 2. General description of the latch-based Register File used in Siwa. Notice the small delay required in the Clk signal. This solution porvides savings of about 30% in area and 25% in power against a typical flip-flop implementation.

registers or feedback paths, and allows for the re-utilization of processing units. The micro-architecture's inherent low performance (in terms of clock speed and instructions per time unit) is not determinant for most IMD, considering that biological signals have bandwidths typically under the kHz range. The system is bounded to 8kB of SRAM in the current implementation; this is enough for the target application, and since the implementation of the MBC is parameterizable, the design may be adapted for any memory size. Constraints in the projects' area and power restricted the system to only one memory level (as main memory for both data and program). The systems bootstrap routine is carried out through the SPI interface that loads the main program from an external 16MB flash memory into Siwa's internal SRAM. After the bootstrap, the flash memory may be also used as secondary storage.

The RF includes CSRs and general purpose registers (GPRs), and it was built with latches instead of flip-flops; an extra latch at the input interface of the RF is required to hold data, in order for it to be captured at falling clock edges, as Fig. 2 shows, with the clock delayed with respect to the write signal. Using latches accounts for 30% savings in the RF area, and about 25% in power, against a flip-flop implementation; this, based on post synthesis evaluations, being equivalent to the use of SRAM in both area and power [6].

III. Evaluation of Siwa's performance

A methodology for feature size scaling, as proposed in [2], was used in order to be able to compare Siwa's power, area, and performance against micro-controllers with different fabrication technologies and features used for IMD applications, as if all the implementations were made in 180nm.

Two metrics were selected for the analysis. The first one is average clock cycles per instruction (CPI), that is, the average time in clock cycles required for the completion of an instruction in a given micro-architectural implementation. Information required to estimate this metric was found in [7]–[9] (even though DMIPS are very common in the measurement

TABLE I
COMPARISON OF SIWA AGAINST OTHER RISC-V CORES.
PROCESS RELATED VARIABLES ARE SCALED ACCORDING TO [2]. DATA USED FOR SIWA IS BASED ON THE POST-PLACE AND-ROUTE AREA AND TIMING REPORTS.

Core	Siwa	Mriscv [3], [4]	Riscy [5]	Zero-Riscy [5]	Micro-Riscy [5]
Technology	180 nm	130 nm	65 nm	65 nm	65 nm
Frequency	20MHz	160MHz	ND	ND	ND
ISA	RV32I	RV32IM	RV32IM+DSP	RV32IM	RV32E
Program Memory	8 kB	4 kB	ND	ND	ND
Average CPI	4	ND	1.27	1.49	1.49
pJ/cycle	48.31	850	63.68	26.12	23.61
Core Area (μm^2)	672146	350250.4	703296	326592	200448

of microprocessors, this implies using the Drhystone benchmark, which is too large for small micro-controllers due to its memory requirements).

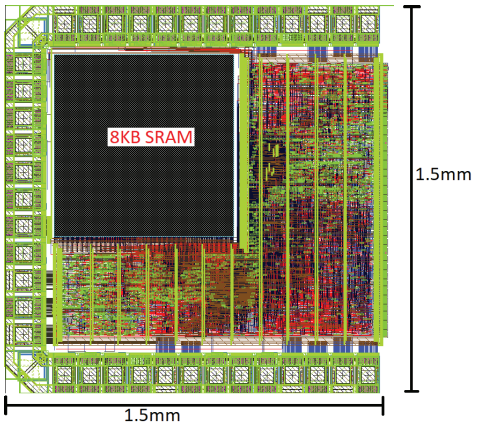

Fig. 3. Siwa's layout sent to fabrication. To the right, 3.3V ports provide interfacing with the analog stimulus circuits (not shown).

The second metric is energy per cycle (EC); that is, the average energy consumed by cycle for a micro-architecture running a task (sometimes found in the literature in terms of μW/MHz). Data was extracted from the data sheets of the implemented devices, where the average total consumption is published for a specific frequency and voltage. For Siwa, power consumption was estimated assuming a balanced test, using all the instructions in the same proportion. It would have been ideal to base this estimate on a software profiling from implantable applications instead of a balanced test, however, since we are not addressing an specific application in the IMD ecosystem in this paper, we leave the specific profiling of software as future work. For the power calculation in the reference designs, we relied on published data.

Siwa was designed on XFAB's CMOS 0.18 μm low leakage, 1.8V standard cells library. Figure 3 shows Siwa's first spin layout, including the space for the SRAM IP. This 8KB SRAM IP represents the biggest block in the implementation and could run faster that the clock cycle used in this iteration, allowing for future improvement of the speed of the system if needed. As the target application in this work is IMD systems, common tasks expected are trimming analog stimulus circuitry and responding to pre-processed input signals from analog filters, which do not require a large amount of memory.

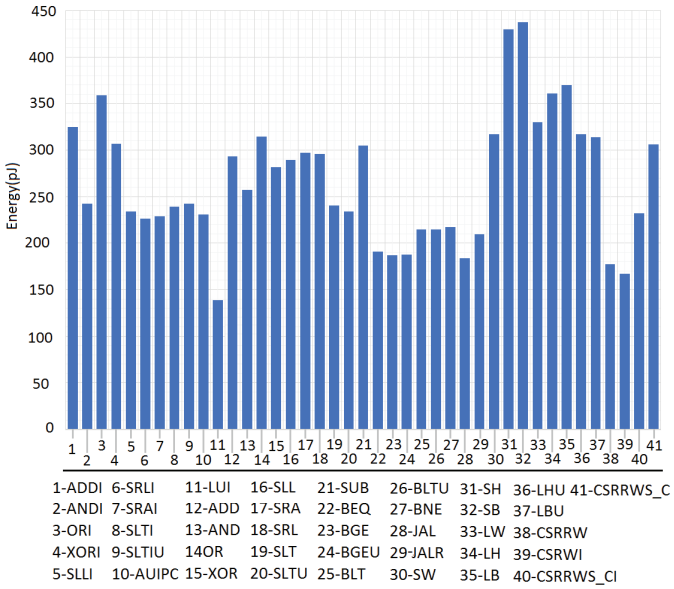

Fig. 4. Energy consumption per instruction. Refer to [1] for instruction details.

A post-place-and-route estimation of the energy consumed by each instruction is presented in Fig. 4, with average energy per clock cycle of 48.31pJ/cycle. To get the energy consumption for each instruction, a test with 1000 iterations of the instruction using random operators was executed and the average power consumption from the post-layout model is reported. To obtain the power consumption, the test is first loaded in the cache; this implies the boot load process is not taken into account. Consistently, the higher energy consumption of all the implemented families of instructions is attained by the load and store instructions due to the additional burden of memory access imposed by these operations. In particular, the load and store byte and half word have the highest energy consumption of all the implemented instructions due to the unaligned access to the memory, taking into account that the micro-architecture is optimized to favor full word (32-bit) accesses. It is important to notice that even when load and store instructions require a bigger energy budget, these instructions also require more clock cycles to be executed so the average power consumption is not severely affected by these instructions.

Table I compares different RISC-V micro-architectures,

TABLE II
COMPARISON OF THE PROPOSED MICRO-ARCHITECTURE WITH OTHER IMPLEMENTATIONS IN THE IMD FIELD

Core	Siwa	8051-Compatible [10]	Atmega328p [7]	PIC16LF1823 [8]	MSP430 [9]
Technology	180 nm	180 nm	ND	ND	ND
Instruction word size (bits)	32	ND	16	8	16
Program Memory	8 kB	ND	32 kB FLASH	2 kB	8 kB RAM
Clock freq.	20 MHz	13 MHz	0-20 MHz	31 kHz- 32 MHz	4-16 MHz
Average CPI	4	ND	1.62	1.1837	≈ 1
pJ/cycle	48.31	70.6	360	54	803
Average power (μW)	70@1MHz@1.8V	918@13MHz@ND	360@1MHz@1.8V	54@1MHz@1.8V	803@1MHz@2.2V

with similar characteristics to Siwa. Table II provides a comparison between Siwa and other micro-architectures reported in IMD's related literature. No Data "ND" fields in these tables correspond to parameters that could not be inferred from the published data. The architecture reported in [3], [4] was developed in a 130nm technology, implements a RV32IM ISA, an AXI-LITE, and an APB bus. In [5], three cores: Riscy(RV32IM+DSP), Zero-Riscy(RV32IM) and Micro-Riscy(RV32E), implemented in 65nm and embedded in a PULPino platform are reported. The first of these cores was intended for DSP, the second was intended for integer operations, and the last one was customized for administrative tasks only.

As it can be noticed, based on the results presented in Tables I and II, parameters for the proposed architecture are comparable to the references, being the third lower in energy per cycle in Table I and the second in Table II.

IV. CONCLUSIONS

Siwa, a custom SOC micro-architecture based on RISC-V, intended for low power medical devices, is presented and evaluated against other RISC-V implementations and other micro-controllers used in the literature for IMDs. Even when this is the first version of the core processor, it presents the lower average energy consumption per clock cycle of all the evaluated micro-architectures commonly used in the IMD field, when scaled using the method presented in [2] to compensate the differences in the fabrication technology and based on post layout simulations.

Since this micro-architecture is built from scratch, and not using third party IPs, except for the SRAM, it provides a base for further customization and improvement, which may be targeted in the future for different applications that may require lower power consumption, higher performance or both.

The chip is expected to be received for physical testing on February 2020. Several optimizations may be taken into account for future iterations of this work, such as reducing the core power supply to 1.2V. This improvement alone implies a reduction in power consumption by a factor of 2.25, but with an unavoidable area penalty due to the need of extra logic for interfacing more voltage domains.

ACKNOWLEDGMENTS

This work was supported by the ANII-Uruguay grant FMV_1_2017_136543. Thanks to Universidad Catolica del Uruguay and Instituto Tecnologico de Costa Rica. Thanks also goes to the Europractice IC Service, and the Synopsys Electronic Design University program.

REFERENCES

[1] A. Waterman and K. Asanovic, "The risc-v instruction set manual-volume i: User-level isa-document version 2.2," *RISC-V Foundation (May 2017)*, 2017.

[2] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of cmos device performance from 180nm to 7nm," *Integration*, vol. 58, pp. 74 – 81, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167926017300755

[3] C. Duran, L. E. Rueda G., A. Amaya, R. Torres, J. Ardila, L. Rueda D., G. Castillo, A. Agudelo, C. Rojas, L. Chaparro, H. Hurtado, J. Romero, W. Ramirez, H. Gomez, H. Hernandez, and E. Roa, "A system-on-chip platform for the internet of things featuring a 32-bit risc-v based microcontroller," in *2017 IEEE 8th Latin American Symposium on Circuits Systems (LASCAS)*, Feb 2017, pp. 1–4.

[4] C. Duran, D. L. Rueda, G. Castillo, A. Agudelo, C. Rojas, L. Chaparro, H. Hurtado, J. Romero, W. Ramirez, H. Gomez, J. Ardila, L. Rueda, H. Hernandez, J. Amaya, and E. Roa, "A 32-bit risc-v axi4-lite bus-based microcontroller with 10-bit sar adc," in *2016 IEEE 7th Latin American Symposium on Circuits Systems (LASCAS)*, Feb 2016, pp. 315–318.

[5] P. Davide Schiavone, F. Conti, D. Rossi, M. Gautschi, A. Pullini, E. Flamand, and L. Benini, "Slow and steady wins the race? a comparison of ultra-low-power risc-v cores for internet-of-things applications," in *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, Sep. 2017, pp. 1–8.

[6] W. M. Elsharkasy, H. E. Yantir, A. Khajeh, A. M. Eltawil, and F. J. Kurdahi, "Work-in-progress: efficient pulsed-latch implementation for multiport register files," in *2017 International Conference on Compilers, Architectures and Synthesis For Embedded Systems (CASES)*, Oct 2017, pp. 1–2.

[7] Z. T. Irwin, D. E. Thompson, K. E. Schroeder, D. M. Tat, A. Hassani, A. J. Bullard, S. L. Woo, M. G. Urbanchek, A. J. Sachs, P. S. Cederna, W. C. Stacey, P. G. Patil, and C. A. Chestek, "Enabling low-power, multi-modal neural interfaces through a common, low-bandwidth feature space," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 5, pp. 521–531, May 2016.

[8] S. Rodriguez, S. Ollmar, M. Waqar, and A. Rusu, "A batteryless sensor asic for implantable bio-impedance applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 533–544, June 2016.

[9] C. S. Mestais, G. Charvet, F. Sauter-Starace, M. Foerster, D. Ratel, and A. L. Benabid, "Wimagine: Wireless 64-channel ecog recording implant for long term clinical applications," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 10–21, Jan 2015.

[10] X. Zhang, H. Jiang, L. Zhang, C. Zhang, Z. Wang, and X. Chen, "An energy-efficient asic for wireless body sensor networks in medical applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 4, no. 1, pp. 11–18, Feb 2010.