

# PUPPY microprocessor: a RISC-V MCU for IoT applications

Gabriel Gouveia

*Department of Electronic Systems*  
*Universidade de São Paulo*  
São Paulo, Brazil  
gabrielgouveia@usp.br

Ivan Hirata

*Digital IC Designer*  
*LSITEC*  
São Paulo, Brazil  
ivantakahira@usp.br

Catherine Pancotto

*Chief Design Engineer*  
*LSITEC*  
São Paulo, Brazil  
catherine.pancotto@lsitec.org.br

Laisa de Biase

*Department of Electronic Systems*  
*Universidade de São Paulo*  
São Paulo, Brazil  
laisa.costa@lsitec.org.br

Bruno Sanches

*Department of Electronic Systems*  
*Universidade de São Paulo*  
São Paulo, Brazil  
bruno.csanches@usp.br

Wilhelmus Noije

*Department of Electronic Systems*  
*Universidade de São Paulo*  
São Paulo, Brazil  
wilhelmus.noije@usp.br

Marcelo Zuffo

*Department of Electronic Systems*  
*Universidade de São Paulo*  
São Paulo, Brazil  
mkzuffo@usp.br

**Abstract**—In the rapidly advancing field of Internet of Things (IoT), there is a growing need for flexible and power-efficient microcontrollers. Addressing this demand, we introduce Puppy, a RISC-V microprocessor designed for low power consumption and offering frequency flexibility. To ensure the reliability of the design, a comprehensive verification environment utilizing the Universal Verification Methodology (UVM) was implemented for functional and timing tests. Successful logical and physical tests were conducted following the physical implementation. Puppy was synthesized using UMC’s 65 nm technology and is currently undergoing fabrication. This project aims to develop an optimized microcontroller unit (MCU) with a novel architecture to cater to future IoT applications, enabling enhanced flexibility in IP integration. The circuit occupies a total (including Pads) small area of 6.68 mm<sup>2</sup>, and from the simulations we got that the power consumption is below 14 mW at 20 MHz.

**Index Terms**—ASIC, Puppy, PULPissimo, low-power, logic synthesis, physical synthesis, microelectronics

## I. INTRODUCTION

THE growing adoption of IoT applications enables seamless connection between physical and digital devices. Engineers design specialized microprocessors optimized for IoT [2], minimizing power consumption while enhancing device functionality. This energy efficiency focus shapes the future of the connected world as IoT technology evolves.

In this context, the RISC-V architecture is gaining attention for its power-efficient optimization in IoT devices. Its simplified instruction set and modular design enable efficient execution with minimal power consumption, extending battery life. Customizability enhances performance in resource-constrained environments. Being a free and open-source ISA,

This study was financed in part by: Banco Nacional de Desenvolvimento Econômico e Social: Smart Cities Pilots, grant no. 6.168.378; Ministério da Ciência, Tecnologia e Inovações and Softex: SBCs Nacionais para IoT Fase II, grant no. 01245.0101 12/2020-11; Ministério da Ciência, Tecnologia e Inovações and Softex: Projeto e desenvolvimento de plataformas not 2.0 com técnicas de manufatura aditiva microeletrônica 3D em sistemas empacotados, grant no. 02/softex/lsi-tec/microeletronica1b; FUNDEP ROTA 2030 - Segurança Veicular 4.0; grant no. 27192\*47.

RISC-V hardware is widely available, making it common and secure for IoT applications. [1].

With a focus on flexibility and innovation, we proudly present Puppy, a Brazilian IoT-oriented microprocessor. Developed for the Caninos Loucos Program, an initiative dedicated to open Single-Board Computers (SBCs) for IoT applications [3], Puppy aims to enhance performance by offering adjustable core and peripheral clock frequencies. This advancement in clock flexibility empowers developers to optimize the microprocessor’s capabilities according to their specific IoT project requirements.

Puppy, a RISC-V microprocessor, is designed for energy efficiency in IoT. Derived from the Pulpissimo project, it offers diverse peripherals, memory and, with a single core fabric controller and optional core clusters, it optimizes IoT end-node performance. [4], [6], [5]. Despite its advantages, the project’s internal frequency-locked loop (FLL) module restricts clock frequency flexibility. Puppy resolves this limitation by externalizing clock signals, providing enhanced flexibility. The development process followed an iterative design approach, incorporating logic, synthesis, and gate-level verification for optimal results.

The present paper is organized as follows: Section II provides an overview of the state of the art and related works. Section III describes the Puppy SoC. The synthesis process is explained in Section IV, while Section V covers the evaluation. Finally, Section VI concludes the paper. This structure allows for a comprehensive understanding of Puppy’s development and its significance in the field of IoT microprocessors.

## II. RELATED WORK

State-of-the-art microprocessors for IoT applications have experienced significant advancements, enhancing performance, efficiency, and integration while exhibiting low power consumption, compact size, and robust computational capabilities [7]. These microprocessors incorporate advanced archi-

technologies like ARM and RISC-V, striking a balance between high performance and energy efficiency.

A popular choice is the nRF52 series by Nordic Semiconductor [8], implementing ARM architecture, and Pulpissimo chips based on the RISC-V instruction set. The nRF52840 in the nRF52 series features a 64 MHz, 32-bit ARM Cortex-M4F processor with rich peripherals and features.

Pulpissimo chips represent the cutting-edge of open-source processor designs for IoT applications. It offers high performance while remaining energy-efficient, which makes it ideal for devices that have limited resources. These chips use the RISC-V instruction set. The PULP platform provides on their website several silicon-proven projects designed in different technologies nodes [9]. Projects like Artemis [10] and Plink [11], based on Pulpissimo, demonstrate low-power consumption and optimized performance in ultralow-power IoT applications. These advancements contribute to the development of efficient microprocessors, catering to the specific requirements of IoT devices.

### III. PUPPY SOC

Puppy SoC is an implementation based on the Pulpissimo architecture, utilizing the 65 nm UMC technology. It features a 32-bit in-order single-issue 4-pipeline stages RV32IMC RISC-V processor. The SoC includes 8 kB ROM for boot code storage and an L2 memory consisting of four 64 kB interleaved banks and two 32 kB private banks. It incorporates an I/O DMA (uDMA) for direct memory access and supports various peripherals such as Quad SPI, I2S, I2C, UART, JTAG, GPIOs, and a camera interface.

Pulpissimo, recently released as open source, provides a comprehensive package including the set of IPs, top-level SystemVerilog hardware description language (RTL level), simulation files, and C-based runtime software, all available for free download [12]. Figure 1 illustrates the Pulpissimo architecture.

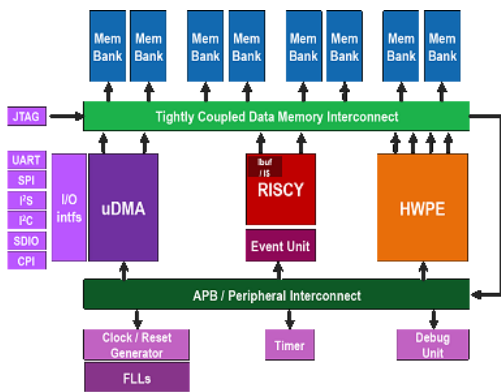


Fig. 1. Pulpissimo architecture [12].

As previously mentioned, our methodology involved rigorous testing, synthesis, and addressing any failures that arose. In this chapter, we delve into the modifications made to the original RTL, the development of tests, and the analysis of

parameters to ascertain their success or failure. By examining these aspects, we aim to provide a comprehensive understanding of the changes implemented and their impact on the overall system.

#### A. Development

To enhance clock flexibility in Pulpissimo, we opted to bypass the internal Frequency-Locked Loop (FLL) module and externalize the core and peripheral clock signals. This allows for independence from the reference signal and facilitates easy modification using external oscillators. Two new clock pads were added to the die to accommodate this change. Subsequently, the synthesis process was initiated, involving the determination of timing parameters such as clock definitions, transition time, and delays. These parameters were specified in the constraints file, which will be further discussed in the subsequent section.

#### B. Tests

Initially, the implemented changes at the RTL level were validated using the provided testbench and original test cases. The logical results matched those of the original system, ensuring the expected behavior was maintained. A verification environment was then developed to achieve high coverage testing, focusing on GPIOs, UART, SPI, I2C, and SRAM modules. Tests were conducted to analyze logical and timing behavior in both RTL code and netlists. Parameters specific to each IP were scrutinized to verify their functionality. Timing tests checked for setup, hold, recovery, and removal violations. Physical verification involved using a DRC tool and technology rulefiles to ensure error-free manufacturing.

### IV. SYNTHESIS PROCESS

This section explains the procedures and techniques used in the synthesis process for both the logic and physical synthesis.

#### A. Logic Synthesis

To perform logic synthesis and generate the netlist we utilized all the RTL files that describes the submodules and top-level, the library files for informations about standard cells and a constraints file, that describes a series of time requirements that the system must meet. For this step, we used Genus, a software from the Cadence chip design support Framework.

To determine the timing constraints, we established the clock period as our reference point and defined values of transition and input/output delay as a small percentage of the clock period.

Throughout the synthesis process, we conducted multiple iterations, consistently refining and updating these parameter values. This approach ensured that the timing requirements were satisfactorily met, resulting in a well-optimized circuit design.

## B. Physical Synthesis

Physical synthesis is the process of transforming a logical design into a physical layout that can be fabricated in silicon. For this step, we used Innovus, a software from the Cadence chip design support Framework, in which we input the netlist and constraints files generated in logic synthesis along with library information of the standard cells and technology.

1) *IO cells*: In this first step we decided what IO cells would be used and determined their order in the die, since it will affect routing. The IO cells include digital signal bidirectional pads, power pads, clock pads, corner cells and fillers. Table I summarizes the IO cells used in this project:

TABLE I  
IO CELLS SUMMARY

| Cell type        | Number | Function                          |
|------------------|--------|-----------------------------------|
| <b>Total</b>     | 316    | -                                 |
| <b>VDD</b>       | 2      | 1.2 V VDD core cell               |
| <b>Ground</b>    | 2      | Ground core cell                  |
| <b>IO VDD</b>    | 2      | 3.3 V VDD IO cell                 |
| <b>IO ground</b> | 2      | Ground IO cell                    |
| <b>Digital</b>   | 41     | Bidirectional signal digital cell |
| <b>Clock</b>     | 3      | Clock crystal signal cell         |

2) *Floorplanning*: Floorplanning is a crucial step in physical synthesis, involving the placement of IO cells, fillers, welltap cells, and macros within the core. The design includes seven macros for SRAM and ROM modules, impacting system routing. A rectangular die shape was chosen to accommodate multiple blocks and maintain density. A 100  $\mu\text{m}$  margin was left between the IO ring and the core to accommodate the power ring placement. This careful floorplanning ensures proper connectivity, power distribution, and optimal space utilization in the design.

3) *Power planning*: At this stage, it was necessary to make decisions about the power and ground distribution.

At first we distributed the VDD/Ground pads evenly around the pad ring, with one pair of VDD/Ground or VDDIO/VSSIO in each side. Then we created the power ring around the core site for VDD and ground and added stripes for both nets evenly inside the core. At last, we used "Global Connect" to make the connections of VDD and Ground with the pins and routed the power nets with the Special Route.

4) *Place and Route*: In this step, we made a few tests with density blockages to avoid placement and routing congestion. After analysing the results, we decided to place a partial blockage with density of 75% where the layout was very congested. This improved the routing results and minimized DRC errors. Also, it was decided to leave sufficient space in the design to accommodate additional IPs that will be implemented in future versions of Puppy. This strategic decision did not impact the tapeout cost, as two sub-blocks were already being allocated."

5) *Final Layout*: After the place and route stage, we can move to the final part of the physical synthesis, which consists of running timing verifications, adding core fillers in order to improve mechanical stability and reliability of the circuit and running verifications such as DRC, Antenna and connectivity.

With a clean layout, we generated reports of parasitic resistance and capacitances, and generated the SDF file and netlist to run timing and functional verification. Since all tests and verification showed no errors, we exported the layout to a GDSII file. Figure 2 shows the final layout.

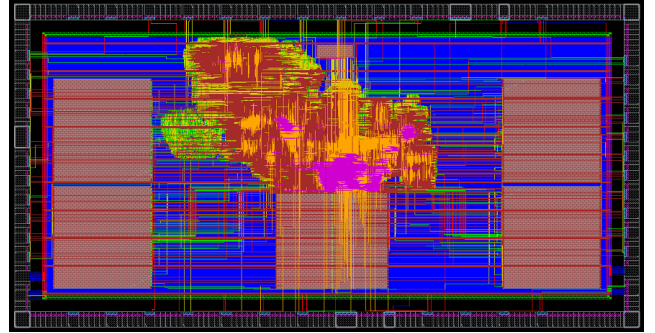


Fig. 2. Final layout

## V. EVALUATION

In this section, we present the coverage rate and the results of the tests conducted using a UVM. All the tests performed yielded positive results, which enhances the reliability of the final chip. Also, we present a comparison with other chips and a summary report.

### A. Tests results

We verified the design at register-transfer level and gate-level (both for the netlists generated at logic and physical synthesis). The tests checked the design for timing violations, such as setup and hold slack, and for the correct functionality of the core and each peripheral. Since no timing violations were reported and the system performed as expected, we concluded that verification was successful.

TABLE II  
FUNCTIONAL AND TIMING TESTS RESULTS

| Unit                             | RTL    | Log. Netlist | Phy. Netlist |
|----------------------------------|--------|--------------|--------------|
| <b>UART - with bit parity</b>    | PASSED | PASSED       | PASSED       |
| <b>UART - without bit parity</b> | PASSED | PASSED       | PASSED       |
| <b>SPI - half duplex</b>         | PASSED | PASSED       | PASSED       |
| <b>SPI - full duplex</b>         | PASSED | PASSED       | PASSED       |
| <b>QSPI</b>                      | PASSED | PASSED       | PASSED       |

We also sampled the data packets sent by the UVM components to analyse the coverage of each test (except for GPIO, since every pin is already verified individually). This way we can guarantee that the design has been thoroughly tested. Table III shows coverage results.

During logic synthesis a few reports were generated about timing properties. These reports showed that timing was met. Table IV shows the slack results.

### B. Comparison with other projects

Table V presents a comparison of a few parameters to show the differences between different implementations of the Pulpissimo architecture.

TABLE III  
TESTS COVERAGE

| Unit                      | Coverage |
|---------------------------|----------|
| UART - with bit parity    | 100.00%  |
| UART - without bit parity | 100.00%  |
| SPI - half duplex         | 98.83%   |
| SPI - full duplex         | 96.51%   |
| QSPI                      | 96.51%   |
| I2C                       | 100.00%  |

TABLE IV  
SETUP SLACK RESULTS OF THE LOGIC SYNTHESIS

| Setup           | R2R   | R2O   | I2O   | CG    |
|-----------------|-------|-------|-------|-------|
| WNS (ns)        | 18.67 | 18.72 | 17.98 | 19.83 |
| TNS (ns)        | 0.00  | 0.00  | 0.00  | 0.00  |
| Violating paths | 0     | 0     | 0     | 0     |

### C. Summary report

The physical verification showed no layout errors, so we can consider the layout ready. Table VI shows the final summary report generated about the properties and estimations of the layout.

## VI. CONCLUSIONS

This paper presents Puppy, a microprocessor implementation based on Pulpissimo, designed for low-power IoT applications with frequency flexibility. Compared to Pulpissimo, Puppy offers enhanced clock frequency definition and lower power consumption (at 20 MHz) than similar works in the same technology node. Synthesis estimations demonstrate Puppy's energy-efficient operation, with power consumption below 14 mW. Its compact total area of 6.68 mm<sup>2</sup> makes Puppy a promising solution for various applications. Extensive testing confirmed the success of Puppy's synthesis, with no errors or critical warnings reported. The chip is currently undergoing fabrication, and will be assembled in QFN56 packages. As soon as they arrive, they will be tested experimentally.

### ACKNOWLEDGMENTS

This project was supported by MCTIC through the PPI-PNM-SBC project; Pr6-Reitoria de Pós-Graduação da Universidade de São Paulo; Laboratório de Sistemas Integráveis Tecnológico (LSI-TEC); SBMicro and Centro Interdisciplinar de Tecnologias Interativas da Universidade de São Paulo (CITI-USP), which was essential for the success of the project.

### REFERENCES

- [1] Electronics Project Focus. [Online]. Available: <https://www.elprocus.com/risc-v-processor/>
- [2] T. Adegbija, A. Rogacs, C. Patel and A. Gordon-Ross, "Microprocessor Optimizations for the Internet of Things: A Survey," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 1, pp. 7-20, Jan. 2018, doi: 10.1109/TCAD.2017.2717782.
- [3] Caninos Loucos Program. [Online]. Available: <https://caninosloucos.org/en/program-en/>
- [4] ] D. Rossi, F. Conti, A. Marongiu, A. Pullini, I. Loi, M. Gautschi, G. Tagliavini, A. Capotondi, P. Flatresse, and L. Benini, "Pulp: A parallel ultra low power platform for next generation iot applications," in 2015 IEEE Hot Chips 27 Symposium (HCS), Aug 2015, pp. 1-39

TABLE V  
COMPARISON OF PARAMETERS BETWEEN MICROPROCESSORS

| Parameter                   | Artemis | Plink | Puppy |
|-----------------------------|---------|-------|-------|
| Die area (mm <sup>2</sup> ) | 1.57    | 3.29  | 6.68  |
| Technology node (nm)        | 65      | 65    | 65    |
| Supply voltage (V)          | 1.2     | 1.2   | 1.2   |
| Clock frequency (MHz)       | 100     | 1.0   | 20    |
| Power (mW)                  | 23.5    | 5.2   | 13.8  |

TABLE VI  
FINAL SUMMARY REPORT

| General design information   |                      |
|------------------------------|----------------------|
| Design name                  | Pulpissimo           |
| Design status                | Routed               |
| Instances                    | 714051               |
| Hard Macros                  | 7                    |
| Std Cells                    | 713728               |
| Gate count                   | 2.236.605            |
| Pads (non-fillers)           | 56                   |
| Total internal memory        | 320 KB               |
| Routing layers               | 8 (ME1 - ME8)        |
| Floorplan Information        |                      |
| Total area of std cells      | 3.02 mm <sup>2</sup> |
| Total area of macros         | 1.72 mm <sup>2</sup> |
| Total area of Pad cells      | 0.94 mm <sup>2</sup> |
| Total area of Core           | 4.76 mm <sup>2</sup> |
| Total area of Chip           | 6.68 mm <sup>2</sup> |
| Core density (after fillers) | 99.52%               |
| Power Information            |                      |
| Total Internal Power         | 7.4990 mW            |
| Total Switching Power        | 4.5730 mW            |
| Total Leakage Power          | 1.7409 mW            |
| Total Power                  | 13.8129 mW           |

- [5] A. Waterman, Y. Lee, D. A. Patterson, K. Asanovic, V. I. U. level Isa, A. Waterman, Y. Lee, and D. Patterson, "The risc-v instruction set manual," 2014.
- [6] P. D. Schiavone, D. Rossi, A. Pullini, A. Di Mauro, F. Conti and L. Benini, "Quentin: an Ultra-Low-Power PULPissimo SoC in 22nm FDX," 2018 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), Burlingame, CA, USA, 2018, pp. 1-3, doi: 10.1109/S3S.2018.8640145.
- [7] M. A. El-Razek, M. B. Abdelhalim and H. H. Issa, "Dynamic power reduction of microprocessors for IoT applications," 2016 28th International Conference on Microelectronics (ICM), Giza, Egypt, 2016, pp. 297-300, doi: 10.1109/ICM.2016.7847874.
- [8] NRF52840 Datasheet. [Online]. Available: [https://infocenter.nordicsemi.com/pdf/nRF52840\\_PS\\_v1.1.pdf](https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.1.pdf)
- [9] PULP Platform. Pulpissimo's silicon proven designs. [Online]. Available: <https://pulp-platform.org/implementation.html/>
- [10] M. Gautschi, M. Schaffner, F. K. Gürkaynak and L. Benini, "An Extended Shared Logarithmic Unit for Nonlinear Function Kernel Acceleration in a 65-nm CMOS Multicore Cluster," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 98-112, Jan. 2017, doi: 10.1109/JSSC.2016.2626272.
- [11] H. Okuhara et al., "A Fully Integrated 5-mW, 0.8-Gbps Energy-Efficient Chip-to-Chip Data Link for Ultralow-Power IoT End-Nodes in 65-nm CMOS," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 10, pp. 1800-1811, Oct. 2021
- [12] PULP Platform. Pulpissimo. [Online]. Available: <https://pulp-platform.org/>