# Sargantana: An Academic SoC RISC-V Processor in 22nm FDSOI Technology

Max Doblas\*, Gerard Candón\*, Xavier Carril\*, Marc Domínguez\*, Enric Erra\*, Alberto González\*, César Hernández<sup>†</sup>, Víctor Jiménez\*, Vatistas Kostalampros\*, Rubén Langarita\*, Neiel Leyva<sup>†</sup>, Guillem López-Paradís\*, Jonnatan Mendoza\*, Josep Oltra\*, Julián Pavón\*, Cristóbal Ramírez\*, Narcís Rodas\*, Enrico Reggiani\*, Mario Rodríguez\*, Carlos Rojas\*, Abraham Ruiz\*, Hugo Safadi\*, Víctor Soria\*, Alejandro Suanes<sup>‡</sup>, Iván Vargas\*, Fernando Arreza\*, Roger Figueras\*, Pau Fontova-Musté\*, Joan Marimon\*, Ricardo Martínez<sup>‡</sup>, Sergio Moreno<sup>¶</sup>, Jordi Sacristán<sup>‡</sup>, Oscar Alonso<sup>¶</sup>, Xavier Aragonés<sup>§</sup>, Adrián Cristal\*, Ángel Diéguez<sup>¶</sup>, Manuel López<sup>¶</sup>, Diego Mateo<sup>§</sup>, Francesc Moll<sup>\*§</sup>, Miquel Moretó<sup>\*§</sup>, Oscar Palomar\*, Marco A. Ramírez<sup>†</sup>, Francesc Serra-Graells<sup>||‡</sup>, Nehir Sonmez\*, Lluís Terés<sup>‡</sup>, Osman Unsal\*, Mateo Valero<sup>\*§</sup>, Luis Villa<sup>†</sup> \*Barcelona Supercomputing Center (BSC), Barcelona, Spain. Email: name.surname@bsc.es <sup>†</sup>Centro de Investigación en Computación, Instituto Politécnico Nacional (CIC-IPN), Mexico City, Mexico. <sup>‡</sup>Institut de Microelectrònica de Barcelona, IMB-CNM (CSIC), Spain. Email: name.surname@imb-cnm.csic.es

<sup>§</sup>Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. Email: name.surname@upc.edu ¶Universitat de Barcelona (UB), Barcelona, Spain. Email: name.surname@ub.edu ∥Universitat Autònoma de Barcelona (UAB), Barcelona, Spain. Email: name.surname@uab.cat

*Abstract*—This paper describes the Sargantana System on chip (SoC), a 64-bit RISC-V single core processor designed by a number of academic institutions and manufactured in 22 nm FDSOI technology: BSC, UPC, UB, UAB, CIC-IPN and IMB-CNM (CSIC). The SoC includes the processor as well as, among other components, a Phase Locked Loop (PLL) operating up to 2 GHz, interfaces to HyperRAM and a Serdes up to 8 Gbps. The processor has demonstrated experimental correct operation at 800 MHz.

## I. INTRODUCTION

The RISC-V open source architecture, originated in 2010 at the University of California at Berkeley [1] is a great opportunity for processor designers to implement powerful processors. In particular, academic groups can experiment with novel microarchitectures without liabilities associated to proprietary architectures.

This paper describes the design, verification and physical implementation of Sargantana, a 64-bit RISC-V single core processor capable of booting Linux. The application-specific integrated circuit (ASIC) was taped-out in Globalfoundries 22FDX (22nm FDSOI) technology. jointly developed in the framework of the DRAC project [2], It contains a 2-level cache hierarchy based on open-source hardware. In addition, a HyperRAM memory controller, interfaces to an ancillary Field-Programmable Gate Array (FPGA) and on-chip clock generators are integrated in the chip.

The structure of the paper is as follows: Section II describes the chip architecture and IP components. Section III explains the main analog blocks designed specifically for this chip. Next, Section IV details the physical design of the chip. Section V displays the results of FPGA emulation of the system and performance of the processor in comparison with similar designs. Section VI shows the measurements performed on the ASIC. Finally, Section VII concludes this work.

## II. CHIP ARCHITECTURE

The Sargantana SoC contains several components and features such as computation capabilities, access to storage, peripherals, input/output communication as well as on-chip clock generation with a PLL. These components are connected via AXI bus in a System-on-Chip (SoC).

Figure 1 shows the system architecture inside the chip as well as connections with other components on the PCB and the FPGA. In the following subsections we will break down the architecture of these components.



Fig. 1. Sargantana SoC block diagram.

<sup>© 2023</sup> IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. https://dx.doi.org/10.1109/DCIS58620.2023.10335976

# A. RISC-V core

Sargantana is a 64-bit RISC-V core developed for the DRAC project. The core is the third generation of the Lagarto family. It supports hardware integer multiply and divide operations (M extension), atomic memory operations (A extension), single and double precision floating point hardware support (F and D extensions), as well as the full privileged instruction set extension (version 1.11), making the core compliant with the RV53G instruction set. Additionally, the core partially supports the Vector extension (version 0.7.1) through a 128-bit packed SIMD. It is targetting bioinformatics and AI workloads with a comparatively reduced power consumption.

The core makes use of a highly optimized 7-stage datapath that issues one instruction per cycle in program order. To achieve a higher performance, the core introduces the outof-order write-back, register renaming, and a non-blocking memory pipeline to reduce the backend stalls. This way, the core can take advantage of the advanced 16 KiB Data L1 design that features up to 16 memory operations in-flight and up to 2 cache line misses. The system implements a 39 bit, page-based virtual memory scheme (SV39), so every access to the first level caches requires a translation from the MMU.

These optimizations allow to achieve 1.2 GHz in the typical corner according to synthesis results.

#### B. Cache Hierarchy

The majority of the Cache Hierarchy is adopted from the cache design developed in the Untethered lowRISC SoC version 0.2 [3]. The lowRISC cache hierarchy has two levels of caches. The private caches of the core are located in the first level (L1), with separate instruction and data caches. In the second level (L2) there is a shared instruction and data cache.

In this case, cache cocherence applies to: 1) maintaining data coherence between the instruction and data L1 in cases of self-modifying code, 2) maintaining the inclusive data policy between L2 and L1 caches (every data present in L1 must also be in L2). The MESI cache coherence protocol [4] is used, as was in DVINO [5], the previous version of the processor. A directory located in L2 is used to maintain the coherence.

The communication protocol between L1 and L2 caches is based on a 128-bit wide TileLink, while AXI and AXI Lite protocols are used to access main memory and peripherals, respectively.

#### C. HyperRAM controller

The on-chip HyperRAM controller was designed for this project following the specifications for this type of memory. The communication between the processor and the Hyper-RAM controller follows the AXI protocol, with a 128-bit data bus. The architecture of the controller is made of a HyperRAM-specific controller block to manage the external memory chips, an AXI wrapper of 32-bit data bus and a FIFO to send or receive blocks of 32 bits from the AXI wrapper. In total we have 6 HyperRAM chips of 8 MB for a total of

48 MB of memory. The HyperRAM controller is configured to work with the HyperRAM chips at up to 100 MHz.

#### D. Communication with ancillary board

Communication with the ancillary FPGA board provides access to DDR3 on that board, as well as other peripherals such as UART and SDcard. This implies to split the system into two parts: one on the ASIC implementing a custom interface to the FPGA connected to the processor via AXI, and another one on the FPGA that uses the integrated DDR3 PHY and on-board memory, among other devices. Both parts are physically connected with an FPGA Mezzanine Card (FMC) connector. This split is depicted in Fig. 1.

There are two such communication interfaces: the *Packetizer* and the *Serdes*, in a concept similar to the Hurricane 2 chip by UC Berkeley [6].

1) Low-speed interface: The Packetizer supports up to 50 MHz transfer speed, and also the 128-bit bus must be split into 16 transactions of 8 bits each, so that it provides a low-speed interface. On the FPGA side, a depacketizer reconstructs these packets back into AXI transactions on the FPGA side.

2) *High-speed interface:* There is a prototype Serdes designed to operate at up to 8 Gbps. In this design, the connection to the core is partial, being able only to transfer pseudorandom data when the core instructs it to do so, instead of transmitting actual memory requests. The *Serdes* interface is described in more detail in Section III-B.

# E. Other peripherals

The ASIC has standalone IO controllers connected to the internal AXI-Lite bus for UART, SDcard and SPI. In addition to that, the Packetizer system has been designed to support the full AXI protocol of 128 bits data width as well as the AXI-Lite of 32 bits. In this way, we can use in the FPGA not only the DDR3 memory controllers connected to the full AXI interface, but also the FPGA IO controllers, such as UART and SPI, connected to the AXI-Lite interface.

#### F. Clocking

The SoC has different clock sources and different frequency configurations that are detailed in this section. Each clock domain source comes from a dedicated output in a Clock Selection Unit (CSU.) The CSU generates the different clocks by selecting and dividing between several possible inputs: external input pin, main PLL (*m-PLL*) or backup PLL (*b-PLL*.) The external clock can be selected from an on-board 100 MHz oscillator (also acting as the *m-PLL* reference) or from an external clock source by an SMA connector, allowing to set an arbitrary frequency for the main clock. The *m-PLL* can be configured to different frequencies up to 2.2 GHz. The *b-PLL* has a fixed output frequency of 1 GHz, with a separate clock reference of 125 MHz. The clock domains are:

1) Main clock domain: It covers the processor core, caches and associated logic. The CSU does not divide the frequency of the selected input source, so the possible values for this domain frequency span 100 MHz (less if an arbitrary source is used) up to 2 GHz.



Fig. 2. Layout of the two PLLs on the chip. Top: *m-PLL*. Bottom: *b-PLL*.

2) *HyperRAM:* The HyperRAM interface supports up to 100 MHz. The CSU has a configurable divider that depending on selected input frequencies it generates 100, 200 or 400 MHz. The HyperRAM controller further divides by 4 this frequency for the interface with the external chips.

*3) Peripherals:* It covers the Packetizer, SPI and UART. The corresponding CSU output divides the selected input frequency by a configurable factor such that it obtains 100 MHz. The Packetizer further divides this signal by 2 (50 MHz), the SPI and uART divide it by 10 (10 MHz.)

## G. Debug infrastructure

We developed a debug infrastructure to verify at runtime the state of the SoC, and to be able to inject internal tests without using the access to main memory. This system is referred in RISC-V literature as a Debug Ring [7]. We developed a custom infrastructure that allows to control the SoC initialization for Linux booting operations.

Our implementation is based on the Open SoC Debug Library (OSD) [8], which provides a plug and play communication interface with a base architecture, and the Generic Logic Interfacing Project (GLIP) [9], which gives a generic data exchange protocol based on FIFO queues.

## III. ANALOG IPS

There are several analog IPs that were designed specifically for this project: on-chip clock generators based on PLL, and a Serdes for communication with an ancillary board. These IPs are configured using a dedicated SPI port.

#### A. PLLs

The *m-PLL* is an Integer-N PLL with a current-controlled ring oscillator based on a nominal external reference of 100 MHz that is capable of generating a clock signal between 1,200 MHz and 2,200 MHz in steps of 100 MHz direct output, or divided output from those frequencies depending on its configuration parameters using the SPI port. The default configuration after a reset is a divided frequency of 400 MHz. The PLL can also operate with other reference clock frequencies, with its output frequencies being consequently scaled. The *b-PLL* is a second PLL that can be used as an alternative clock source and it was included to have a backup option in case of malfunctioning of either PLL. It is an instantiation of the PLL used in the Serdes. This PLL has a fixed output frequency of 1 GHz with an external reference of 125 MHz, the same as the one used for the Serdes.

Fig. 2 shows the layout of both PLLs: the *m*-PLL at the top has dimensions of 77 x 80  $\mu$ m, while the *b*-PLL at the bottom has dimensions of 22 x 31.5  $\mu$ m.

## B. Serdes

A Serdes (Serializer-Deserializer) interface in 22FDX technology was newly developed. It is intended to act in future tapeouts as a high-speed interface between the processor and the ancillary FPGA board, although in this chip it acts almost standalone only for testing purposes, sending pseudo-random data sequences when instructed by the core processor. This sequence is generated by a Pseudo-Random Binary Stream (PRBS) generator in the digital interface.

The Serdes has two distinct parts: the digital interface for the Physical Coding Sublayer (PCS) layer and the high-speed analog frontend for the Physical Medium Attachment (PMA) layer, as shown in Fig. 3.

The PMA is able to establish a communication link at 2, 4 or 8 Gbps by selecting appropriate configuration registers. It features a parallel user data interface of 32-bits running at 125 MHz (8 Gbps), 62.5 MHz (4 Gbps) or 31.25 MHz (2 Gbps). The transmitter is a Current Mode Logic (CML) driver with configurable current through a DAC and Feed Forward Equalizer (FFE.) At the receiver, the signal is filtered with a Continuous Time Linear Equalizer (CTLE), amplified with a StrongArm and then equalized with a 5-stage Decision-Feedback Equalizer (DFE.) Then, a Clock and Data Recovery circuit (CDR) based on a PLL discriminates the clock signal from the data stream and generates the necessary clock signals for the deserializer. Once the clock signal is locked in the desired frequency by the CDR, the deserialization takes place and the 32 bits are transmitted to the PCS at the core.

Signal integrity is expected to be an issue at these speeds and for this reason it was decided to have a large number of tuning parameters for the different blocks in the PMA. There are 64 bits to digitally tune the transceiver behavior with associated Digital to Analog Converters (DAC): bias current for the CDR, sign bits and bias current for the DFE and for the FFE.

The PMA analog frontend layout is shown in Fig. 4. The block has a size of 280 x 76  $\mu$ m.

The PCS digital interface is integrated with the rest of the core.

## IV. CHIP PHYSICAL DESIGN

The Sargantana chip layout is shown in Fig. 5. The design measures 2.49 mm width by 1.17 mm height, for a total area of 2.91 mm<sup>2</sup>. Fig. 5 highlights the different IPs in the chip.

Since the circuits with the largest area and complexity are digital, the chosen strategy is DoT (Digital On Top). This



Fig. 3. Serdes description.



Fig. 4. Layout of the Serdes analog frontend (PMA.)

means that we work as if the chip were purely digital, and at the mid-point we add the information of the analog blocks that were designed separately.

There are 3 independent power domains: Serdes (also including the *b-PLL*,) *m-PLL* (including ADC and SPI port) and Core (rest of the chip.) All of them have a nominal voltage of 0.8 V. In addition, there is a separate power domain for the IO ring, at 1.8 V.

In total, there are 116 pins in a 144 CPGA package. 22 of the 116 pins are for power and ground, and 2 pins are for applying forward body biasing. he rest of pins, 92, are for IO.

A static timing analysis (STA) was performed. Three timing constraints were set depending on the corner: 800 MHz for the slow corner (SSG), 1,200 MHz for the typical corner (TT) and 1,600 MHz for the fast corner (FFG.) Fig. 6 shows the maximum frequency for several corners and conditions, including forward body bias and temperature.

# V. FPGA EMULATION

The deployment platform for FPGA emulation is the KC705 board from Xilinx. It contains a Kintex-7 FPGA with 326,080 logic cells, 840 DSP slices, 16,020 memory cells, on-board Ethernet, UART, QSPI, 2 FMC expansion connectors, DDR3 controller and 1GB DDR3 SODIMM storage.



Fig. 5. Chip area distribution.



Fig. 6. Max frequency analysis of the design in different corners, body bias and temperature conditions.



Fig. 7. Instructions per cycle (IPC) of different benchmarks comparing the processor to a previous version.

The FPGA results in this section have been obtained by running several bare-metal benchmarks on the emulated SoC. The instructions and cycles values have been extracted from the hardware counters in the Performance Monitoring Unit (PMU) of the core.

Figure 7 shows the instructions per cycle (IPC) obtained in each benchmark by the processor in this paper compared to the DVINO [5]. On top of the bars we can see the configuration or data size used in each run. The rightmost column represents the geometric mean across all benchmarks and configurations. It is observed a significant improvement thanks to the changes introduced in the architecture.

# VI. POST-SILICON RESULTS

To test the ASIC we designed a PCB with all the required circuits and connectors (FMC for the FPGA, JTAG, UART, micro SD).

Below is a summary of measurements made so far, keeping in mind that at this stage they are focused on functionality.

# A. PCB verification

Fig. 8 shows a picture of the ASIC board connected to the FPGA board. The on-board regulators provide the correct voltages 3.3 V, 1.8 V, 0.8 V from the external 5 V supply. The on-board oscillator for the *m-PLL* gives the correct 100 MHz



Fig. 8. Experimental setup connecting the ASIC board to the FPGA board.

clock signal. The PCB power consumption without the chip is 1 W: 200 mA total current measured from the 5 V source. With the Chip attached to the PCB and selecting as the clock source the 100 MHz oscillator, the total current consumption of the PCB rises to 255 mA, equivalent to a power consumption of 1.275 W. The consumption of the chip under these conditions can thus be estimated to be 275 mW.

## B. Core execution via Bootrom and Debug Ring

The Bootrom controller was verified by loading a small test program to an EEPROM, consisting on writing in the 32 registers of the core. The controller successfully transferred the instructions to the core and after that we used the debug ring to verify the correct execution of the instructions.

The Debug ring module was verified using the JTAG interface placed on the PCB board, the following procedure was applied:

- 1) The JTAG interface was connected to a computer running Linux OS through a FTDI cable.
- The Debug ring software back-end was executed in a terminal and connection with the SoC was successfully established (Chip ID was given by the processor)

The debug ring allows access to the L2 cache and in this way indirectly access the program counter and the state of execution.

#### C. Clock Selection Unit (CSU)

The CSU can select between the external 100 MHz oscillator and the PLL. In both cases, the correct clock frequency for the processor main clock, the hyperram and the fmc out pin were verified.

Using the *m-PLL* clock output, The CSU works properly within the range of 400 MHz to 1,000 MHz (PLL divided clock output.)

Applying the *b-PLL* clock output to the CSU results in frequency shifts, The processor and the peripherals can't operate with this clock source.

#### D. m-PLL operation

The *m-PLL* was tested under the following conditions:

- 100 MHz reference clock (on board clock bypassed to PLL input) 0.8 V External dedicated voltage supply and 0.8 V shared internal voltage supply.
- Input clock applied to the CSU coming from the PLL's divided output.
- Even number to set the internal division factor.
- Bias voltage higher than 0.5 V to start the PLL.
- Clock duty cycle at 50%.

To test each PLL clock frequency, the processor was accessed through the Debug ring and several write/read instructions were executed to verify a correct operation.

Table I shows the power consumption of the *m-PLL* for several PLL frequencies. The power was measured using a high precision Source and Measurement Unit, SMU, powering the *m-PLL* domain.

TABLE I POWER CONSUMPTION FOR THE M-PLL.

PLL Out	PLL Current	PLL Power
(MHz)	(mA)	(mW)
800	1.1	0.88
1200	1.52	1.22
1400	1.7	1.36
1600	1.8	1.44
2000	1.98	1.58

The power consumption of the processor was estimated at each clock frequency by measuring the current consumption of the 5 V source powering the PCB and subtracting the PCB power consumption without the chip The *m*-*PLL* and Serdes have independent power supplies at 0.8 V. Also note that this estimation includes IO activity of the chip.

Table II summarizes the power consumption of the chip according to the clock frequencies generated by the PLL and the processor's clock that is driven by the CSU. To achieve frequencies in the range of 200 - 500 MHz for the processor clock (main clock domain) the CSU divides by 2 the frequency of the PLL divided clock, thus applying a factor of 4 between the PLL frequency and main clock frequency. On the other hand, to achieve frequencies in the range of 600 - 800 MHz for the processor's clock the CSU drives the PLL's divided clock without further division.

 TABLE II

 Power consumption (estimated) for the processor.

Core Clock (MHz)	Processor Power (mW)
200	570
300	725
400	850
500	1,000
600	1,085
700	1,150
800	1.330

# E. PCS FPGA-FPGA test

The PCS functionality was tested by connecting two FPGAs through a bridge PCB custom designed for this purpose, as



Fig. 9. Test setup for the PCS between two FPGA boards.

shown in Fig. 9. The interface board contains a clock generator and two FMC connectors.

One of the FPGAs emulated the complete processor design including the SerDes-PCS submodule, while the other FPGA only contained the Serdes-PCS design. The Kintex Serdes GTX on the FPGAs was the PMA used in the communication.

The FPGA-interface board connects the TX and RX serial pins and shares a reference 125 MHz clock to each of the FPGA boards.

On the side of the processor a test program is executed to access its AXI memory and check the number of packets received from the other FPGA, packet ID and the number of transmission errors via UART. Once the processor receives the packets, it retransmit them back to the other FPGA in which the TX, RX and control signals can be seen though waveforms in Vivado.

After powering up the set-up and executing the test program, the serial transmission took place correctly. The channels were paired and aligned and 1,000,000 packets containing PRBS data were transmitted at 8 Gbps without any error.

## F. On-chip Serdes PMA preliminary test

The PMA on the ASIC has many calibration and configuration settings and it has not been thoroughly tested yet. Using the reset settings a preliminary test communicating with an FPGA has been tried.

On the FPGA side the PCS is programmed to retransmit data from the ASIC in a loopback configuration. The two boards are connected through the FMC connector, sharing the TX, RX lines and the 125 MHz clock reference from the ASIC board.

On the FPGA side an Integrated Logic Analyzer (ILA) is instantiated in the decoder to check the RX serial data coming from the ASIC. In the waveform screen it could be seen that both SerDes are aligned and paired. Also there is a 64-bit sequence of 1010 as expected.

Further characterization work is currently under way.

#### VII. CONCLUSIONS AND FUTURE WORK

The presented processor is a new implementation of the RISC-V architecture in Globalfoundries FDSOI 22nm tech-

nology targeting bioinformatics and AI workloads thanks to its tightly coupled SIMD unit.

The core presents a significant improvement in IPC with respect to its predecessor, and the technology used together with the on-chip PLL allows to reach 800 MHz clock frequency in the laboratory measurements in the preliminary tests done so far. This corresponds to the slow corner in the timing analysis.

The analog IPs were developed from scratch for this technology. The *m-PLL* has demonstrated its functionality, while the Serdes testing is still ongoing effort. The large number of calibration settings was designed to have many options given the uncertainties of the experimental conditions. However, this requires a long time to find the optimal values for reliable transmission.

There is still work to do regarding the Linux capability, that depends on the Bootrom, HyperRAM demonstration and several other interfaces.

#### ACKNOWLEDGMENTS

The DRAC project is co-financed by the European Union Regional Development Fund within the framework of the ERDF Operational Program of Catalonia 2014-2020 with a grant of 50% of total eligible cost. The authors are part of Red-RISCV which promotes activities around open hardware. The Lagarto Project is supported by the Research and Graduate Secretary (SIP) of the Instituto Politécnico Nacional (IPN) from Mexico, and by the CONACyT scholarship for Center for Research in Computing (CIC-IPN).

#### REFERENCES

- Andrew Waterman, Yunsup Lee, David A. Patterson, and Krste Asanovic. "The RISC-V Instruction Set Manual, Volume I: Base User-Level ISA". In *Tech. Report UCB/EECS-2011-62, EECS Dept., UC Berkeley*, pages 97–116, 2011.
- "Designing RISC-V-based Accelerators for next generation Computers". https://drac.bsc.es/, 2019-2022.
- [3] lowRISC. "Untethered lowRISC v0.2". https://www.lowrisc.org/docs/ untether-v0.2/. [Online; accessed 27-August-2019].
- [4] Mark S. Papamarcos and Janak H. Patel. "A Low-Overhead Coherence Solution for Multiprocessors with Private Cache Memories". In *Proceedings of the 11th Annual International Symposium on Computer Architecture*, ISCA '84, page 348–354, 1984.
- [5] Guillem Cabo et al. "DVINO: A RISC-V Vector Processor Implemented in 65nm Technology". In XXXVI Conference on Design of Circuits and Integrated Systems (DCIS), pages 1–6, 2022.
- [6] John Wright. Design of a lightweight serial link generator for test chips. Master's thesis, EECS Department, University of California, Berkeley, Dec 2017.
- [7] lowRISC. "Overview of the debug infrastructure". https://www.lowrisc. org/docs/debug-v0.3/overview/, 2018.
- [8] The Open SoC Debug Contributors. "The Open SoC Debug Documentation Library". https://opensocdebug.readthedocs.io/en/latest/index.html#, 2018.
- [9] Institute for Integrated Systems (LIS). "The Generic Logic Interfacing Project". https://www.glip.io, 2018.