

Design of Steel ASIC, a RISC-V processor

Vinicius dos Santos, Fábio Petkowicz, Rafael da Silva, Rafael Calçada, Ricardo Reis
Instituto de Informática (INF), Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, RS, Brazil
{vssantos, fabio.petkowicz, rsilva, rocalcada, reis}@inf.ufrgs.br

Abstract—The present work describes the design of an ASIC version of STEEL, a RISC-V microprocessor, developed at UFRGS. The microprocessor core called STEEL implements the RV32I and Zicsr instruction sets of the RISC-V specifications. The ASIC circuit operates with a maximum frequency of 19.61 MHz and the estimates obtained from the physical synthesis indicated an estimated power consumption of 10.09 mW.

Index Terms—ASIC, RISC, Steel Core, microprocessor, micro-electronics

I. INTRODUCTION

Integrated circuits (ICs) revolutionized electronics, and today they are present in most technology applications and in an great part of the industry [1]. According to the Statista Research Department (2021), the worldwide integrated circuit market reached 361.23 billion U.S. dollars in revenue in 2020. Furthermore, the estimated market grow in 2021 is by over 20 percent to 436.37 billion U.S. dollars [2]. Application-Specific Integrated Circuits (ASICs) are ICs manufactured for two main purposes: (1) to meet user’s specifications for the demands of a particular system; or (2) for reuse so that several other macrosystems can be designed considering the already finalized ASIC as a component [3]. A traditional design methodology for ASICs is the standard cell one, which consists of mapping a complex circuit into pre-designed logic cells. The cell library contains the description of the layout of each cell.

However, this work describes all stages of an ASIC design developed using a standard cell design methodology. It was used the Cadence EDA tools, which provides state-of-the-art services for circuits design. The circuit chosen to be designed was the Steel Core microprocessor [4], whose description RTL - Register Transfer Level was developed at UFRGS and is available on the OpenCores.org portal [5].

This paper is organized as follow: In Section II the STEEL microprocessor is described. Section III depicts the methodology of this work, including the entire process of synthesis of the microprocessor. Section IV presents the estimates obtained from the complete synthesis. Finally, the conclusions are reported in Section V.

II. STEEL CORE OVERVIEW.

There are two main architectural approaches to designing computers: CISC (Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer), which differ regarding to the quantity and complexity of the instructions. Generally speaking, RISC architectures have a significantly

smaller and simpler instruction set when compared to CISC architectures [6].

Steel is a 32-bit RISC-V microprocessor core designed to be easy to use and primarily targeted for use as a softcore in embedded system designs. Steel implements the basic RISC-V instruction set RV32I and the Zicsr [7] extension only, so that it can be used as a processing unit in small and medium size embedded systems. Although it can also be used in large systems, they usually require features absent in Steel, such as specialized instructions for integer multiplication and division (M extension), floating-point arithmetic (F extension), among others. Nevertheless, Steel is capable of running embedded software and even real-time operating systems, taking advantage that nearly all extensions can be emulated by the RV32I base instruction set. Steel has been used in one commercial application at the company where the project creator works. Its repository on the internet is downloaded five times a week on average and has been forked seven users so far.

Figure 1 presents the Steel top module architecture. There are two interfaces for memory access, an interface for connecting it to an interrupt controller and another to read from a real-time counter. Its two memory buses allow the construction of a computer system based on both the Harvard architecture (connecting it to two memories, one for data and the other for instructions) and the von Neumann architecture (connecting it to a single dual-port memory).

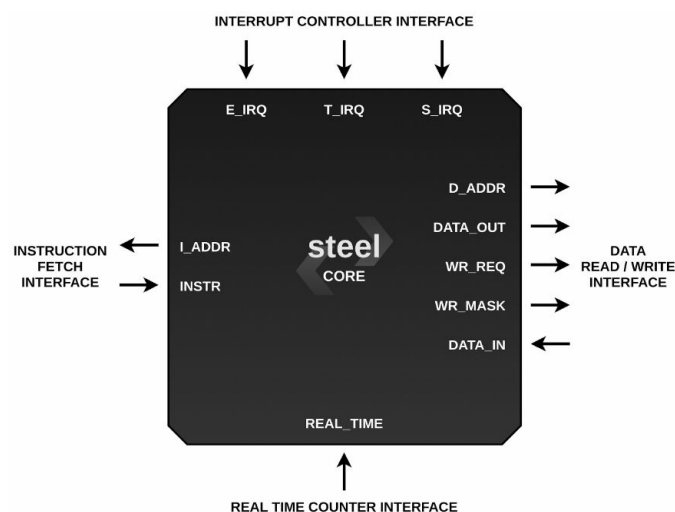


Fig. 1: Steel Core Interface [4]

A. Specification and Architecture

Figure 2 presents a detailed logic diagram of Steel's microarchitecture as described in Verilog. Steel has three pipeline stages, a single execution thread and issues one instruction per clock cycle. Therefore, all instructions are executed in program order. Its pipeline is simple, divided into fetch, decode, and execution stages. The small number of pipeline stages eliminates the need for branch predictors and other advanced microarchitectural units, like data hazard detectors and forwarding units, making the design of Steel easy to understand.

III. DESIGN METHODOLOGY

This section presents the synthesis design flow of the microprocessor based on Cadence's tools, which basically consists of: simulation of the RTL description, the logical synthesis of the register transfer level (RTL) description (mapping of the logical units of the circuit to cells of the standard cell library) and finally the design of the physical synthesis of the circuit.

A. Register Transfer Level

RTL is a high-level design abstraction that models the random processor logic through control signals and data flow between registers. We tested the RTL target description using the NClaunch and SimVision analysis environment of Cadence Design Systems for the synthesis of the microprocessor. The expected result of the simulation (confirmed after running the testbench) is the value of the sum of the constants on the output bus right after the program's last instruction. With this, the validation of the circuit is carried out.

B. Logical Synthesis

The logical synthesis maps the description in RTL for standard cells [3]. The used cell library was the XC018 MOSST Digital Core Library [8], from X-FAB Semiconductors Foundries. This cell library uses an 180nm technology. The logical synthesis was divided into three basic steps in this project's scope: Constraints, Generic Technological Mapping, and Optimized Technological Mapping.

Constraints file define the clock period, the rise and fall times, the ramp at the inputs and outputs of the circuit for rise and fall transitions, and the minimum capacitance. The main definition is the clock period, and rise and fall time, as these parameters affect the slack and clock delays such as setup and hold times. The clock parameters were defined to obtain a positive slack value being 5 percent of the used clock value. After several synthesis with different clock values, the chosen one was a clock with a period of 51 ns. This synthesis output describes the processor at the gate level and is used in the physical synthesis.

C. Physical Synthesis

After the logic synthesis, the physical synthesis use as input the netlist generated in the logic synthesis step, which describes the design with the blocks, gates, and logical connections between them. We can divide the physical synthesis

into two steps: Floorplanning and Backend Flow. In the floorplanning step, we obtain the floor plan of the processor, where we aim to minimize the area and delay. This step occurred as follows:

1) *I/O Placement*: The I/O placement consists of placing the pins of the circuit. Thus, the insertion of pads was performed during the initial stage because the positions of the pads can influence the posterior routing.

2) *Power Planning*: In addition, we defined the power planning, which adds VDD and Ground lines and creates a power ring around the core. One of the concerns of the distribution of the metal layers that generate the circuit supply is the unwanted generation of parasitic capacitances caused by the proximity of two charged electric conductors that imitate the plates of a capacitor.

3) *Floorplanning*: The floorplanning step consists of the placement the macros (memory, IPs, etc.) and PADs in the desired places to provide the best circuit performance, whether in timing, power, or area. A square floorplan was generated at this stage, with a density of 70 percent and margins of 15 μm . The choice of these numbers was based on tests carried out in the later stages of routing. The distribution of the VDD and GND tracks, both vertically and horizontally, should be as homogeneous as possible to avoid heating points on the chip.

4) *Placement and Routing*: In physical synthesis, the processes of Placement and Routing of cells are the most time-consuming. The standard cells placement step consists of distributing all of them in the layout core. The routing step refers to the physical interconnection of the standard cells pins through wires and several metal layers. The main concerns when routing are congestion and the average length of connections. In this step, we carry out the placement of the cells together with their pre-routing, or as called by Cadence's tool, a weak routing where the tool is concerned at first with the placement of the cells. Afterward, a cell routing was performed using Nanoroute. Figure 3 shows the circuit rails. The Rail Analysis step consists of verifying the power distribution in the layout to ensure that the supply voltage does IR Drop below a certain level, causing an increase in delays operating standard cells.

5) *Clock Tree Synthesis*: The step of synthesizing the clock tree consists of adding buffers in the clock signal paths [9]. Clock distribution is a meticulous task in the synthesis design flow. We work with six layers of metals, where the clock signal starts from the upper pad near the left corner and is distributed to all cells that need it. Thus, the clock skew becomes one of the main concerns: the clock signal must reach different components simultaneously to guarantee the synchronism and correctness of the operation. The addition of buffers serves to mitigate the effects caused by different wire sizes. The number of buffers in each path is adjusted to correct time violations. This synthesis step is performed with a file of information passed to the tool as a maximum and minimum delay.

6) *Filler cells and Metal Fill*: At this stage, the design is already validated. All DRC checks and other checks have already been carried out, and all reported errors or violations

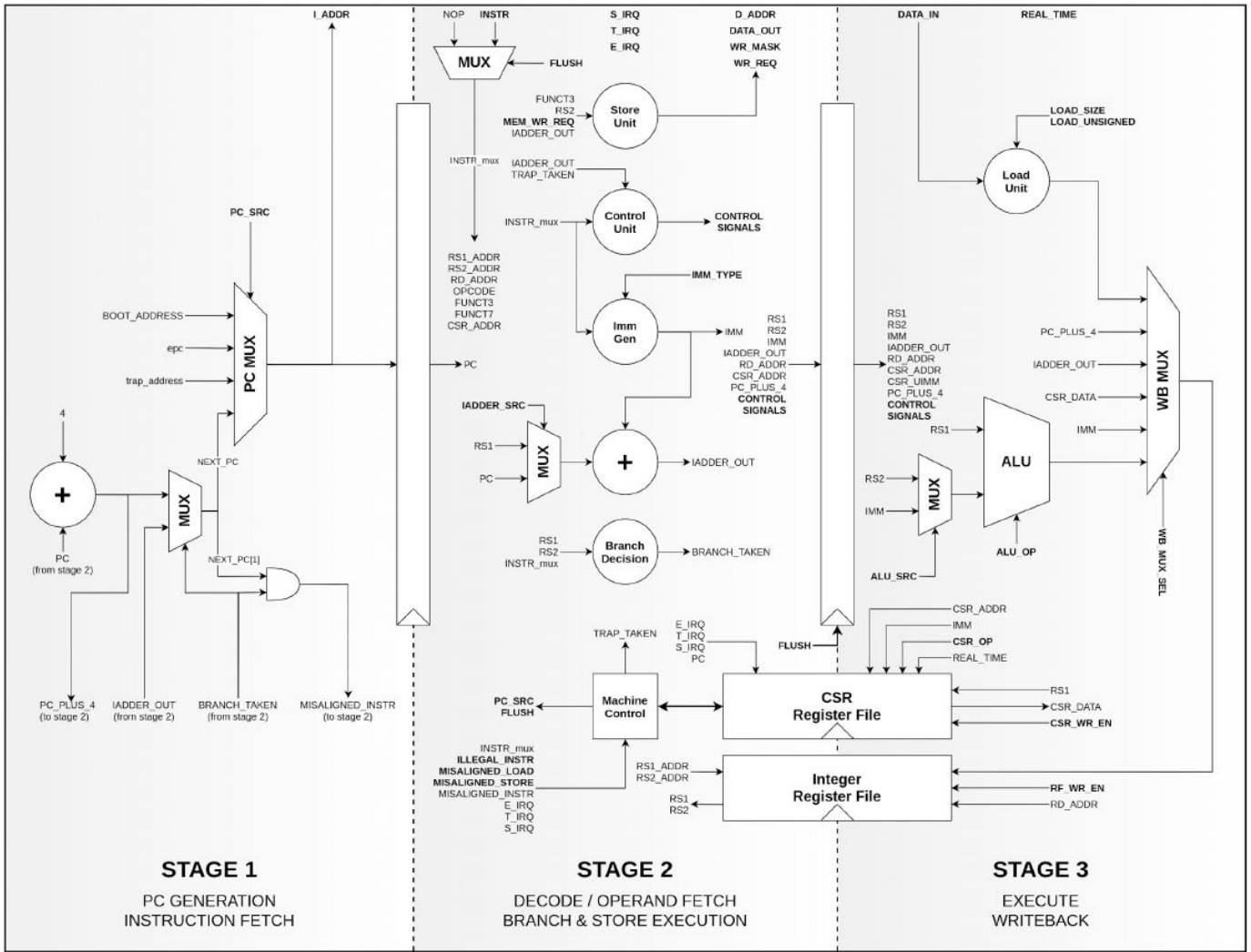


Fig. 2: Steel Microarchitecture [4]

have been resolved. With this, we can perform the insertion of filler cells, which are used to fill the empty spaces between the standard cells already placed, avoiding planarity problems. These cells do not alter the functional characteristics of the circuit in anyway. The insertion of metal fill has a similar function to that of filler cells: to help to maintain the density of metal layers, during the chemical mechanical planarization (CMP) step of the manufacturing process, with a certain percentage of metal density in each of the used metal layers, according to the foundry specifications [10]. The metal fill also avoids the likelihood of an antenna effect. Yet, the metal fill is performed to uniform the density of metal on the chip. This is the last step related to the physical synthesis of the Steel processor.

IV. RESULTS AND DISCUSSION

Table I presents the results reported by Innovus after completing the physical design. From it, we get estimates for critical design issues such as timing, power, and area.

The presented results were obtained right before GDS file generation, in the late stage of the design flow and after performing parasitic capacitance extraction and verification such as DRC, process antenna, connectivity, geometry and metal density. Innovus did not report any errors or alerts after these checkings. The Steel ASIC can reach 19.61 MHz and 10.09 mW as power consumption. The power consumption (static and dynamic) was estimated using Cadence tools, which provide these sources after the completion of the physical design, taking into account several factors such as: used PDK characteristics, supply voltage, circuit geometry and frequency of clock signals. No simulation or execution of real inputs has been performed. However, Cadence tools provide very accurate information that takes into account the variability of the input signals. Dynamic power consumption is estimated. Currently, the estimate provided by the design tools is very close to that found in real scenarios. It was not compared with [4] implementation because our work is synthesized for ASIC, and the original version was designed for FPGA.

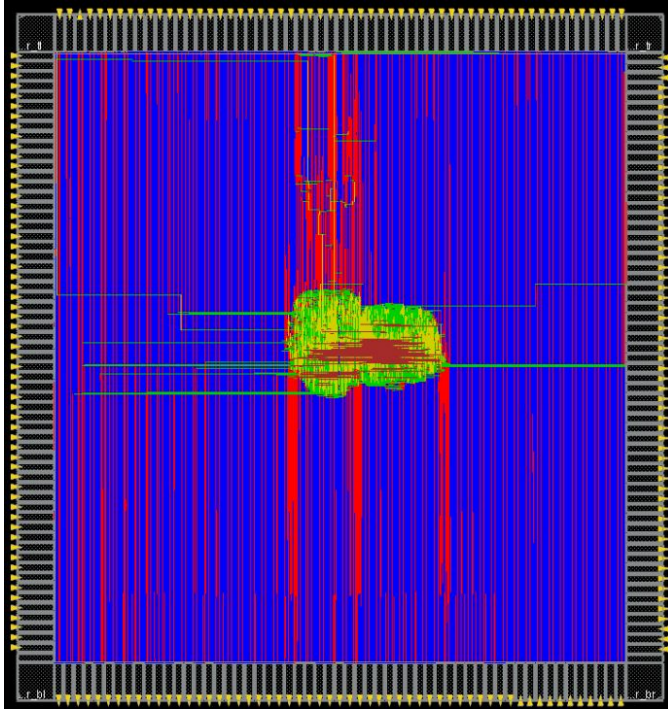


Fig. 3: Steel Processor Core and Rails

TABLE I: Result after the physical synthesis

General Design Information	
Design Status	Routed
Design Name	SteelCore top
Instances	231471
Hard Macros	0
Std Cells	231231
Pads	240
Net	7288
Special Net	2
I/O Pins	234
Pins	25234
PG Pins	463661
Average Pins Per Net (Signal)	3.462
General Cell Library Information	
Routing Layers	6
Masterslice Layers	3
Pin Layers	2
Layers	15
Netlist Information	
No of Nets (Int)	7179
No of Connections (Ext)	18242
Floorplan/Placement Information	
Total area of Standard cells	168,81 mm^2
Total area of Core	15.793,46 mm^2
Total area of Chip	20.166,83 mm^2
Core Density (w/Std Cells MACROs)	99.996%
Power Report	
Total Internal Power	7.51769609 mW
Total Switching Power	2.57942532 mW
Total Leakage Power	0.00045145 mW
Total Power	10.09757281 mW

Nevertheless, it isn't easy to compare the results obtained for Steel regarding parameters such as area, frequency and power to similar ASIC implementations of RISC-V processor cores. These parameters are heavily dependent on the manufacturing process technology and the used standard cell library. A fair comparison assumes that all variables influencing these parameters are kept constant. The works that could be compare Steel to (e.g., PicoRV32, Ibex Core, or Syntacore SCR1), cannot be as they were manufactured using other PDKs.

V. CONCLUSIONS

This paper reports the ASIC synthesis of the Steel processor, describing how the ASIC was designed. The estimates from the physical synthesis indicate that the implemented processor has an area around 20.166,83 mm^2 , switching and leakage power around 2.58 mW and 0.00045145 mW , respectively, with a clock period of 51ns, which is equivalent to a frequency of 19.61 MHz.

As future work, the routing can be optimized. When distributing the PADs around the Core, the clock signal pad is placed in the upper left corner. With that, it can be improved the performance if the pad placement is in the center. However, the actual version works and has an acceptable performance. A real implementation in silicon is being studied for the near future.

ACKNOWLEDGMENT

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) Finance Code 001, the National Council for Scientific and Technological Development – CNPq and Fapergs (Research Support Foundation of the State of Rio Grande do Sul).

REFERENCES

- [1] M. Haselman and S. Hauck, "The future of integrated circuits: A survey of nanoelectronics," *Proceedings of the IEEE*, vol. 98, no. 1, pp. 11–38, 2009.
- [2] T. Alsop. (2021, jun) Semiconductor integrated circuits global revenue 2009-2022. [Online]. Available: <https://www.statista.com/statistics/519456/forecast-of-worldwide-semiconductor-sales-of-integrated-circuits/>
- [3] G. D. Hachtel and F. Somenzi, *Logic synthesis and verification algorithms*. Springer Science & Business Media, 2007.
- [4] R. d. O. Calçada, *Design of Steel: a RISC-V Core*. UFRGS, 2020.
- [5] R. Calçada. (2020, Oct 14) Steel core. [Online]. Available: <https://opencores.org/projects/steelcore>
- [6] D. Patterson, "Reduced instruction set computers then and now," *Computer*, vol. 50, no. 12, pp. 10–12, 2017.
- [7] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovi, "The risc-v instruction set manual. volume 1: User-level isa, version 2.0," UC Berkeley Dept of EE and CS, Tech. Rep., 2014.
- [8] X.-F. S. Foundries. Xc018 cmos data sheet, x-fab semiconductor foundries. [Online]. Available: <https://www.xfab.com/technology/cmos/018-um-xc018/>
- [9] J.-L. Tsai, L. Zhang, and C. C.-P. Chen, "Statistical timing analysis driven post-silicon-tunable clock-tree synthesis," in *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005*. IEEE, 2005, pp. 575–581.
- [10] Subramanian. (2005) Performance impact from metal fill insertion. [Online]. Available: http://www.axiomweb.co.uk/cadence/MetalFill_Paper.pdf