electronics

MDPI

Article

Design and Test of Offset Quadrature Phase-Shift Keying Modulator with *GF180MCU* Open Source Process Design Kit

Emma Mascorro-Guardado [1], Susana Ortega-Cisneros [1,*], Emilio Isaac Baungarten-Leon [1,2], Luis A. Luna-Rodriguez [1], Uriel Jaramillo-Toral [1], Manuel Hernández-Aramburo [1] and Emanuel Murillo-García [1]

[1] Centro de Investigación y de Estudios Avanzados (CINVESTAV) del Instituto Politécnico Nacional (IPN), Department of Electrical Engineering, Guadalajara 45017, Mexico; emma.mascorro@cinvestav.mx (E.M.-G.); emilio.baungarten@cinvestav.mx (E.I.B.-L.); luis.luna@cinvestav.mx (L.A.L.-R.); uriel.jaramillo@cinvestav.mx (U.J.-T.); manuel.aramburo@cinvestav.mx (M.H.-A.); emanuel.murillo@cinvestav.mx (E.M.-G.)

[2] Universidad Autónoma de Guadalajara, Diseño Ciencia y Tecnologia, Ciudad Universitaria, Zapopan 45129, Mexico

* Correspondence: susana.ortega@cinvestav.mx; Tel.: +52-33-3777-3600

Abstract: This article explores the evolution of integrated circuits (*ICs*), highlighting the fundamental role of *open source* Electronic Design Automation *(EDA)* tools in their development. It describes the *IC's* design flow, differentiating between *Front-end* and *Back-end* design stages, and details the process of implementing the digital stage in offset quadrature phase-shift keying *(OQPSK)* modulation in an *IC*, including its hardware description language *(HDL)*, the implementation test in the field-programmable gate array *(FPGA)*, and the physical layout using the first manufactured open source process design kits (*PDKs*) in *Global Foundries'* 180 nm, as well as the use of *OpenLane* and *Caravel*. To conclude, the results of the physical tests obtained from the digital modulation are presented, as well as the performance of the raised cosine shaping filter.

Keywords: back-end; EDA; front-end; IC; ISI; MSB; OQPSK; open source PDKs; open source; RTL

Citation: Mascorro-Guardado, E.; Ortega-Cisneros, S.; Baungarten-Leon, E.I.; Luna-Rodriguez, L.A.; Jaramillo-Toral, U.; Hernández-Aramburo, M.; Murillo-García, E. Design and Test of Offset Quadrature Phase-Shift Keying Modulator with *GF180MCU* Open Source Process Design Kit. *Electronics* **2024**, *13*, 1705. https://doi.org/10.3390/electronics13091705

Academic Editor: Ricardo Martins

Received: 5 March 2024
Revised: 3 April 2024
Accepted: 16 April 2024
Published: 28 April 2024

1. Introduction

Modern electronics, since the advent of the transistor in 1947 by John Bardeen, Walter Brattain, and William Shockley at Bell Laboratories, have undergone a radical transformation. This milestone, followed by the development of the *IC* in 1958 by Jack Kilby and Robert Noyce, revolutionized the electronics industry through miniaturization and efficiency [1]. The increase in the complexity of *ICs* design and manufacturing led to the development of *EDA* tools by companies such as Cadence Design Systems (1988), Synopsys (1986), and Mentor Graphics (1981), which revolutionized *IC* design by establishing industrial standards [2]. *EDA* tools have become essential in the digital era, optimizing everything from design to the manufacturing of *IC* and promoting continuous innovation that has radically transformed electronics.

However, a notable limitation in the implementation of *ICs* is that most of the tools required for their development are not free of cost. This represents a significant barrier, especially for researchers, small businesses, and independent electronic design enthusiasts who may not have the financial resources to access these expensive tools. In response to this limitation, a significant shift towards the adoption of free-of-cost tools for the manufacturing of digital *ICs* has been observed. The emergence of open source hardware such as *MAGIC* (http://opencircuitdesign.com/magic/ (accessed on 15 April 2024)) [3], *Yosys* (https://yosyshq.net/yosys/faq.html (accessed on 15 April 2024)) [4], *OpenROAD* (https://theopenroadproject.org/ (accessed on 15 April 2024)) [5], and *GTKWave* (https://gtkwave.sourceforge.net/ (accessed on 15 April 2024)) has expanded access to advanced technologies and fostered an environment of collaboration and experimentation.

In this context, a pioneering development occurred with the release of the first manufacturable open source *PDKs*, specifically *SKY130* (https://skywater-pdk.readthedocs.

io/en/main/ (accessed on 15 April 2024)) [6] and *GF180MCU* (https://gf180mcu-pdk.readthedocs.io/en/latest/ (accessed on 15 April 2024)) [7]. Simultaneously, the appearance of *OpenLane* (https://github.com/The-OpenROAD-Project/OpenLane (accessed on 15 April 2024)) [8], a highly automated and freely accessible *EDA* tool, has further accelerated the accessibility and openness of hardware design processes. With a workflow that encompasses more than seventy scripts and customizable utilities, *OpenLane* facilitates the implementation of designs with any technology or *PDK*, covering key stages such as synthesis, floor planning, placement, clock tree synthesis (*CTS*), routing, tapeout, and signoff.

While *SoC* frameworks have been created to accelerate software application compilation, synthesis, cycle simulations, power analysis, and *SoC* prototyping, none have been capable of supporting a layout-level template that is manufacturable. This limitation existed due to the lack of freely usable *PDKs* until recently [9–11].

Efabless (https://efabless.com/ (accessed on 15 April 2024)) has pioneered the development of *Caravel* (https://github.com/efabless/caravel (accessed on 15 April 2024)), a versatile layout system-on-chip (*SoC*) template that facilitates the seamless integration of digital, analog, and mixed systems alongside an *RISC-V* processor [12], all ready for manufacturing with the open source *PDKs*. In a collaborative effort with industry giants *Google*, *SkyWater*, and *Global Foundries*, *Efabless* has initiated complementary free manufacturing rounds for 130 nm and 180 nm technologies. This strategic partnership not only expands the horizons of open source hardware but also fosters innovation by providing access to manufacturing processes at no cost.

In the semiconductor industry, selecting the right foundry for fabrication is pivotal, as it determines the technology node, cost, and minimum area requirements for manufacturing. Table 1 compares fabrication layouts among various foundries and multi-project wafer (MPW) programs, highlighting their technology nodes, prices per square millimeter, and minimum area requirements. This comparison is crucial for semiconductor companies seeking an ideal foundry partner for their chip designs.

MPW programs offer several advantages, including cost-effectiveness through shared manufacturing costs among multiple projects. Additionally, MPW services often provide access to advanced technologies and design libraries that may not be feasible for individual projects. This shared environment fosters collaboration and innovation, enabling companies to achieve more efficient and cost-effective chip development.

Table 1. Comparison of multi-project wafer programs.

MPW Service	Foundry	Technology	Price Range per mm^2	Minimum Area
MOSIS [13]	TSMC	12–350 nm	Available upon request [a]	4 mm^2
MOSIS [13]	Intel	16 nm	Available upon request [a]	4 mm^2
CMC M. [14]	GF	12–130 nm	USD 7.35–46 k	1 mm^2
CMC M. [14]	STM	28 nm	USD 14.01 k	1.25 mm^2
CMC M. [14]	TSMC	28–350 nm	USD 700–25.1 k	1–25 mm^2
EUROPRACTICE [15]	GF	12–180 nm	USD 3.2–147 k	1–4 mm^2
EUROPRACTICE [15]	TSMC	28–180 nm	USD 1.04–9.75 k	1–6 mm^2
EUROPRACTICE [15]	X-FAB	180 nm	USD 1.8 k	2.25 mm^2
Efabless [16]	SkyWater	130 nm	USD 0–574 [b]	17 mm^2
Efabless [16]	GF	180 nm	USD 0 [c]	17 mm^2

[a] Charge of USD 10 k per registration and minimum fabrication order of USD 25 k; [b] free manufacturing with *Google* sponsorship; [c] only available with *Google* sponsorship.

While free manufacturing rounds are ideal for any researcher or university, these rounds are limited; on the other hand, Efabless' ChipIgnite program offers a lower price than its competitors for 130 nm nodes.

The motivation for this work lies in the important advantages of implementing *ICs* with open source tools. These advantages include not only the reduction or non-existent cost

in *IC* design and fabrication, but also the promotion of broad collaboration and knowledge sharing within the *IC* design community. By providing unrestricted access to cutting-edge resources and methodologies, the use of open source tools stimulates innovation and the development of more creative solutions tailored to specific needs. This open approach has proven to be particularly valuable in the rapid advancement of technology, where adaptability and the ability to respond to new needs are essential. Furthermore, expansion in the field of *ICs* generates wide opportunities both in academia and in the professional realm, opening doors to new horizons of study and work in this dynamic field.

Additionally, it is important to highlight the significance of digital modulation within digital systems due to its revolutionary role in communication systems, improving the precision and efficiency of data transmission compared to analog methods.

The *OQPSK* [17] technique stands out in this context, being especially valuable for low-frequency communication in Internet of Things (IoT) applications, as exemplified in the *Zigbee* [18] standard, which benefits from its efficiency and reliability in IoT network data transmission.

The decision to implement *OQPSK* in *ICs* for the *Zigbee* standard was motivated not only by improved data transmission but also by the advantages of synchronous design over asynchronous digital approaches, such as reduced chip area, lower energy consumption, and simplified design. These qualities enhance system integration and optimize both the size and energy consumption of the device, ensuring its effective functionality in practical applications through verification and validation processes. The amalgamation of these benefits provides a solid foundation for the advancement and development of future communication technologies, paving the way for more sophisticated and efficient innovations.

On the other hand, open source tools face limitations such as less extensive technical support and documentation, compatibility issues with industrial standards, and limitations in advanced functionalities. In addition, they may have limited processing capacity compared to commercial tools and are based on larger-scale technologies, which restricts their applicability in cutting edge projects. The lack of tools that generate specific test modes for large corporations can also be an obstacle in advanced industrial environments.

The quality of the final design may be affected by limitations in the simulation, verification, and validation capabilities of open source tools. However, for projects that do not require the latest technology or for designs with less strict requirements, open source tools may be sufficient.

Furthermore, these limitations offer a valuable opportunity to deepen the understanding of the *IC* design process flow. They act as an educational resource in themselves, allowing designers to explore and understand the complexities of *IC* design in a practical and detailed manner, as evidenced in the case study of this work, thus enriching the learning experience and encouraging innovation within the design community.

In this paper, we present the design and implementation of an *OQPSK* modulator using open source *EDA* tools and the open source *GF180MCU PDK*, resulting in the *IC* and its respective measurement. The main contributions are summarized as follows:

- Comprehensive overview of the *IC* design flow, highlighting the use of open source tools and the *GF180MCU PDK*.
- Successful integration of the *OQPSK* modulator into the Caravel *SoC* template and its manufacturing.
- Proof of concept through simulations conducted during the design flow and physical measurement of the chip.
- Post-processing technique used to compensate for the absence of the most significant bit *(MSB)* in physical measurements.

The remainder of this article is structured as follows: Section 2 discusses related works involving these free-of-cost tools. Section 3 describes the design flow of digital *ICs*. Section 4 explains the functionality of the *OQPSK* modulator. Section 5 addresses the practical implementation of the *OQPSK* modulator in an *IC*. Section 6 provides the results derived from the simulations and experimental evaluations to which the *IC* was subjected during its design

and after its manufacturing. Section 7 provides a critical discussion of the study results and possible future research directions, and, to conclude, Section 8 summarizes the study's conclusions.

2. Related Work

This section summarizes the current state of the research on open source projects conducted in recent years.

Guthaus et al. (2016) developed *OpenRAM* [19], an open source memory compiler designed to generate, characterize, and verify customizable static random-access memories *(SRAMs)*. It facilitates research and development in memory design, offering adaptability to various technologies with its Python base. It provides functional and layout models, optimizing access times and memory densities for different configurations. Its modular approach and portability support innovation and experimentation in the field of circuit and integrated system design.

Ajayi et al. (2019) focused on the development of *OpenROAD* [5], a fully automated and open source integrated circuit design tool. Its aim is to simplify and speed up the chip design process, reducing costs and the need for specialized expertise. It combines efforts from academia and industry, promoting a more accessible and open hardware design approach.

Simon Buhr et al. (2019) achieved sub-nanosecond synchronization on a fast Ethernet physical layer (PHY) chip using 180 nm technology from *GLOBALFOUNDRIES* [20], demonstrating precision and efficiency in data transmission.

Another work by Simon Buhr et al. (2019) presented a PHY transceiver for fast Ethernet [21], optimized for low power consumption, setting a new standard for efficiency.

Zhang et al. (2022) introduced the first open source generator for temperature sensors [22] using a leakage-based design, and validated it on SkyWater 130 nm, enabling efficient design exploration.

Lucas Daudt Franck et al. (2023) developed an application-specific *IC* (ASIC) accelerator for secure hash algorithm-2 (SHA)-256 using open source electronic design tools and SkyWater's 130 nm technology [23], standing out for its optimization and compatibility with 32-bit microcontrollers.

Shah et al. (2023) presented FABulous [24], an open and highly customizable FPGA fabric generator, demonstrated on the first silicon manufactured in the open Skywater 130 nm *PDK*, sponsored by *Google*. The FPGA includes logic, register, DSP, and RAM blocks with flexible configuration options. Open source tools are used for synthesis and routing, alongside a VGA demonstration. This work, supported by EPSRC, the Google Shuttle program, and Carl-Zeiss-Stiftung, highlights the viability and effectiveness of open FPGA infrastructure in the hardware design and manufacturing process.

As observed, research in electronic engineering and *IC* design using open source technologies significantly drives innovation, efficiently optimizing both data transmission and parallel processing.

A notable example is the use of the *GF180MCU* open source PDK in the development of an *OQPSK* modulator. This particular case demonstrates how the accessibility and flexibility offered by open source tools can be leveraged to address specific technical challenges, enhancing the efficiency and precision of communication systems.

This work, like the examples previously mentioned, shows how open collaboration and innovation can transform the design of electronic systems. By detailing the unique contributions of each project, the diversity of advancements made possible through open source is highlighted, from improvements in synchronization to the optimization of energy consumption.

These initiatives mark a promising direction for future research and development, underscoring the crucial role of open source as a catalyst for technological advances in electronic engineering.

3. *IC* Design Flow

3.1. Front-End

In the development of digital *ICs*, the *Front-end* focuses on the conceptualization and logical validation of the design. This process is essential to ensure that the *IC* design meets the requirements before moving on to the *Back-end* stages, which involve the physical arrangement of components *(layout)*, chip fabrication, and post-manufacturing testing.

The *Front-end* stage includes the following:

- Specification Definition: the requirements that the *IC* must meet are determined, considering aspects such as processing speed, power consumption, noise tolerance, and storage capacity.
- High-Level Design (HLD): the general architecture of the system is defined, including the development of block diagrams and the representation of data flow, with the aim of developing and validating system-level models of the design.
- Logical Design and Register Transfer Level *(RTL)*: The functionality of the circuit is described at a higher level of abstraction than transistor-level design, using *HDLs* such as *VHDL or Verilog*. This involves translating the specifications into code that details the flow of bits through registers and corresponding logical operations.
- Functional Verification: The design is simulated to confirm that it meets the specifications and is free of logical errors. This phase is essential to validate the functionality of the circuit before proceeding to the physical design.
- *RTL* Synthesis: *RTL* synthesis transforms *HDL* code into a network of logical gates and circuit components to implement the design in silicon. This process includes logical optimization to improve performance and efficiency, mapping to specific library cells of the process technology, the generation of a netlist detailing the interconnection of these cells, the integration of design constraints to guide optimization, and validation of the netlist to ensure it meets design requirements.

3.2. Back-End

The conversion of *RTL* designs to the graphic design system *(GDSII)* format constitutes the *Back-end* process in *IC* design. This process is responsible for finalizing the physical preparation of the design, thereby facilitating its subsequent manufacture. The process includes a series of steps to follow:

- Floorplanning: during this stage, floorplans are created, an early stage in the hierarchical approach to integrated circuit design. In other words, it's a schematic representation of the tentative placement of its cells and macros.
- Placement: involves placing standard cells and macros, with subsequent placement checks to ensure the integrity of the design.
- *CTS*: involves placing clock branches and adding necessary buffers to ensure the integrity of the clock signal.
- Routing: the routing phase then begins with a two-step approach, starting with an initial routing followed by a more intricate routing process.

The final stages of the *Back-end* process involve a comprehensive set of verification, including design rule check *(DRC)*, layout versus schematic *(LVS)*, and static timing analysis *(STA)*. The successful completion of these stringent checks indicates that the design is ready for approval.

Throughout this research, the *Back-end* process was meticulously executed using *OpenLane* in conjunction with the *GF180MCU PDK*.

4. *QPSK* Functionality

4.1. Digital Modulation in Baseband

QPSK is an advanced modulation technique that enables efficient data transmission in terms of bandwidth utilization. This modulation varies the phase of a carrier signal in four discrete values, where each is represented by two bits (00, 01, 10, 11). The outcome is a

signal capable of carrying double the amount of information that could be transmitted by binary phase-shift keying *(BPSK)*, which only uses two values [25].

Figure 1 illustrates the modulator used for the *ZigBee standard*. It consists of a *serial-to-parallel converter* for mapping data in phase *(I)* and quadrature *(Q)*. In the latter, data are delayed by a bit time to ensure that the carrier signal does not have a phase change greater than 90 °, thus reducing bandwidth and abrupt changes in the temporal signal. This is achieved using a technique known as *offset QPSK*, where the *Q* sequence data are delayed by a time equivalent to one bit.

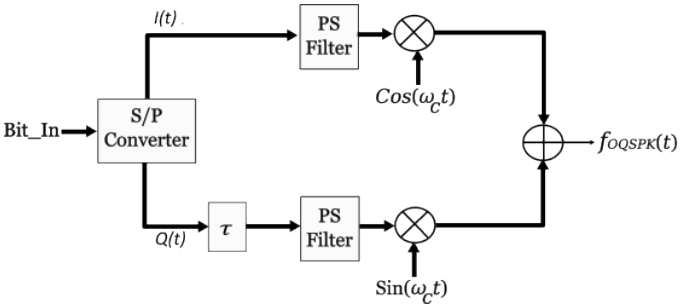


Figure 1. *OQPSK* modulator.

After phase and quadrature mapping, pulse shaping is applied to the data using a shaping filter. This process is essential to limit the signal bandwidth and prevent interference between consecutive symbols. For the *ZigBee standard, half-sine pulse shaping* is used. Mathematically, this waveform is described as Equation (1):

$$g(t) = \sin\left(\frac{2\pi}{T_s}\right), \quad 0 \leq t \leq \frac{T_s}{2} \tag{1}$$

where $\frac{2\pi}{T_s}$ refers to the angular frequency of the wave, and T_s is its period [26].

Finally, the shaped I and Q signals are multiplied by their respective carrier signals: one for the phase and another shifted by 90 ° for the quadrature. The sum of these two modulated signals results in the baseband signal that is transmitted.

4.2. Shaping Filter with a Duration Longer Than the Symbol Time

Pulse shaping plays a crucial role in the spectral efficiency of the transmitted signal. A *half-sine pulse*, lasting exactly one symbol time, can be used for this purpose. However, extending the pulse duration beyond one symbol time can further restrict the bandwidth [26]. This is achieved by a *raised cosine pulse*, which is widely recognized for its ability to smooth the signal transition in the frequency domain.

The *raised cosine pulse* is well known because, in the frequency domain, it consists of the sum of a constant and a *half-cycle of cosine*, resulting in a reduction in sidebands and, therefore, less interference with adjacent signals. This pulse shaping is described through Equation (2), which details the relationship between time and frequency.

$$g(t) = \text{sinc}(2W_t) \cdot \frac{\cos(2\pi\alpha W_t)}{1 - (4\pi\alpha W_t)^2} \tag{2}$$

where the sinc function $\text{sinc}(2Wt)$ limits the spectrum of the signal, with W representing the pulse bandwidth and t the time. The cosine modulation, expressed by $\cos(2\pi\alpha Wt)$, introduces a roll-off factor α, facilitating a smooth transition between the pass and stop bands of the raised cosine filter. The angular frequency $2\pi Wt$ modulates the sinc function, thus shaping the signal over time to meet the specific requirements of the transmission. The denominator $1 - (4\pi\alpha Wt)^2$ acts as a normalization factor, preventing mathematical singularities and ensuring the function's stability over time.

This pulse-shaping approach not only improves bandwidth efficiency but also helps to mitigate inter-symbol interference (ISI), making *OQPSK* modulation highly effective for digital communications where spectrum conservation and signal clarity are paramount [26].

In the mathematical domain, the process of filtering a signal can be effected through the convolution technique, which involves integrating the product of the input signal and the filter's impulse response. The functional relationship describing this procedure is specified by the following mathematical equation:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k) \tag{3}$$

The analysis of Equation (3) reveals that a series of sequential operations are required for the computation of the convolution:

1. Reflection: the function $h(k)$ is subjected to a reflection operation to obtain $h(-k)$.
2. Displacement: the reflected function, $h(-k)$, is displaced n positions to the right or left, depending on whether n is positive or negative, respectively, resulting in the function $h(n-k)$.
3. Multiplication: each term $x(k)$ is multiplied by the corresponding $h(n-k)$, producing the intermediate sequence $vn(k) = x(k)h(n-k)$.
4. Summation: a summation of all terms in the sequence $vn(k)$ is carried out to obtain the final value of the output signal at the moment n.

Since the raised cosine signal is even, the reflection operation does not alter its shape. In addition, the multiplication of each signal sample by the filter's impulse response will result in a positive or negative signal depending on the non-return-to-zero *(NRZ)* encoding used in the data flow.

5. *OQPSK* Implementation

5.1. Description of OQPSK Digital Modulator

Figure 2 illustrates the implementation diagram of the *OQPSK* modulator, considering the following features [27]:

- Frequency bands: 2400–2483.5 MHz.
- Modulation: *OQPSK*.
- Shaping filter: raised cosine.
- Bit rate: 2 Mbps.
- Bit frame size: 12 fractional bits and 1 sign bit.
- Samples per symbol: 50.

This modulator is structured into two main components: a digital segment, responsible for data mapping and filtering, and an analog segment, tasked with multiplying by carrier signals and summing the branches. The operation of the digital part is synchronized through a 50 MHz clock signal and receives data at a rate of 2 Mbps, which is fed through first-in, first-out memories *(FIFO)*. In parallel, a *numerically controlled oscillator (NCO)* manages the operations in the analog block and regulates the reading frequency of the *FIFO* memories, thus allowing the proper adaptation of the input data for processing in subsequent analog stages.

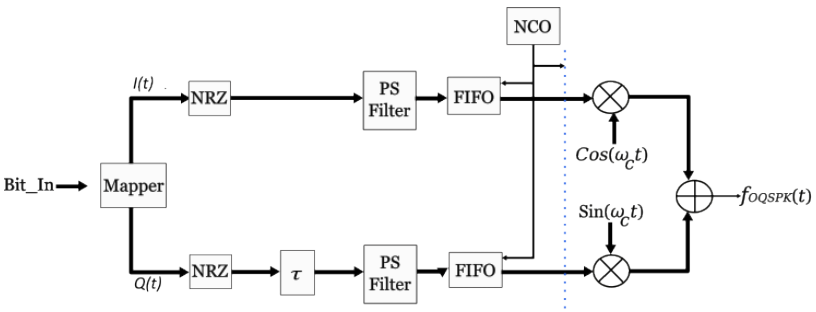


Figure 2. *OQPSK* modulation for FPGA and *IC*.

5.2. RTL Implementation of OQPSK Modulator

The approach to the modulator's [27] implementation focused exclusively on its digital component, with the intention of integrating it into an *IC*. This decision was based on leveraging the open source tools used for this process. The ensured that the digital modulator was adaptable and efficiently applicable across a broad spectrum of applications, thereby enhancing its value and applicability in the scientific and technological fields.

The *RTL* (Appendix A) of the circuit was developed using Verilog *(HDL)*, and an *FPGA* was used to verify and validate the design's behavior before proceeding with *IC* manufacturing. The use of *FPGA* is not only a common practice in electronic system design but also offers significant advantages in terms of flexibility, cost, and time. This strategy ensures that the final *IC* design is well validated and optimized, thereby enhancing its performance and reliability in practical applications.

Figure 3a illustrates the design of the digital stage implemented at the *RTL* level. It includes three main signals: *Clk*, *rst*, and *En*. When the *rst* and *En* signals are activated simultaneously, the transfer bit enters the module called *S/P buffer* (serial-to-parallel converter) at a speed of 2 Mbps, which serves as a *mapper*. The output bit from this *mapper* is then concatenated with the current value bit to form a two-bit symbol. Subsequently, the Register block operates at a frequency of 1 MHz to obtain the symbol at a speed of 1 Mbps.

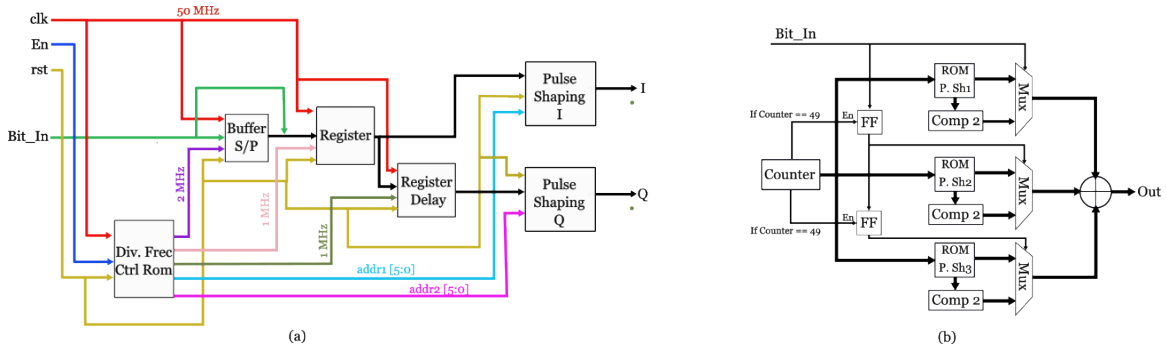


Figure 3. Block diagram of *OQPSK* modulator implemented (**a**). Structure of the pulse shaping (**b**).

The *raised cosine filter*, shown in Figure 3b, is located within the *pulse shaping* module, with a duration of three symbol periods (3 Ts), and is stored in multiple read-only memory *(ROM)* units, one for each symbol timing. This system includes a *counter* for 50 samples, which oversees the access direction to the memories simultaneously. The first incoming symbol bit is directed towards the first *multiplexer* which determines whether the sample retrieved from the memory *P. Shap1* will be sent in a positive or negative form, representing *NRZ* modulation. While a null value enters the other *multiplexors*, resulting in a zero output, these three values are then summed to generate the modulated sample.

Once the transmission of all samples corresponding to a symbol is complete, a signal is emitted that activates the *flip-flops*, allowing the newly entered symbol bit to govern the second *multiplexer* and manage the output from the *memory P. Shap2*, while the new bit oversees the *memory P. Shap1*.

When another bit is transmitted, the *flip-flops* are activated again, and the previous bits manage the subsequent *multiplexors*, while the new bit governs the first *multiplexer*, simulating the shift and multiplication before summing the output, thus obtaining the convolution of the filtered sample.

For the quadrature stage, before entering the shaping filter *multiplexer*, the symbol bit coming from the *Register* block goes into the *Register Delay* block, which is responsible for generating the *offset* with a bit time. It consists of a register that is activated with a 1 MHz *signal*, with a 2 Mbps delay. Therefore, *Q* has a time phase shift with respect to *I*.

5.3. From RTL to GDSII

The *RTL* to *GDSII* process is performed through *OpenLane*, which starts with HDL synthesis, where the *Yosys* tool optimizes the *RTL* file, creating a netlist mapped by the *PDK*. Design constraints such as clock definitions and boundary conditions can be integrated, and STA can be performed using *OpenSTA* (https://github.com/The-OpenROAD-Project/OpenSTA (accessed on 15 April 2024)).

Next is floorplanning, where the *TritonFPlan* (https://github.com/The-OpenROAD-Project/OpenROAD (accessed on 15 April 2024)) tool is used for macro-related tasks, generating a design exchange format (*DEF*) file and defining matrix and macro core sizes. Important attributes like the power distribution network (*PDN*), decoupling capacitors, and potential shunt cells are also defined.

After floorplanning, standard cell and macro placement are performed using the *RePlAce* (https://github.com/The-OpenROAD-Project/RePlAce (accessed on 15 April 2024)) tool. *CTS* is performed using *TritonCTS* (https://github.com/The-OpenROAD-Project/TritonCTS (accessed on 15 April 2024)) [28]. Routing is then conducted in two steps: first, an initial routing phase with *FastRoute* (https://github.com/The-OpenROAD-Project-Attic/FastRoute (accessed on 15 April 2024)), followed by a more detailed routing process with *TritonRoute* (https://github.com/The-OpenROAD-Project/TritonRoute?tab=readme-ov-file (accessed on 15 April 2024)).

In the final stages, the design undergoes thorough verification, including *DRC*, *LVS*, and *STA*; this workflow is illustrated in Figure 4 [8].

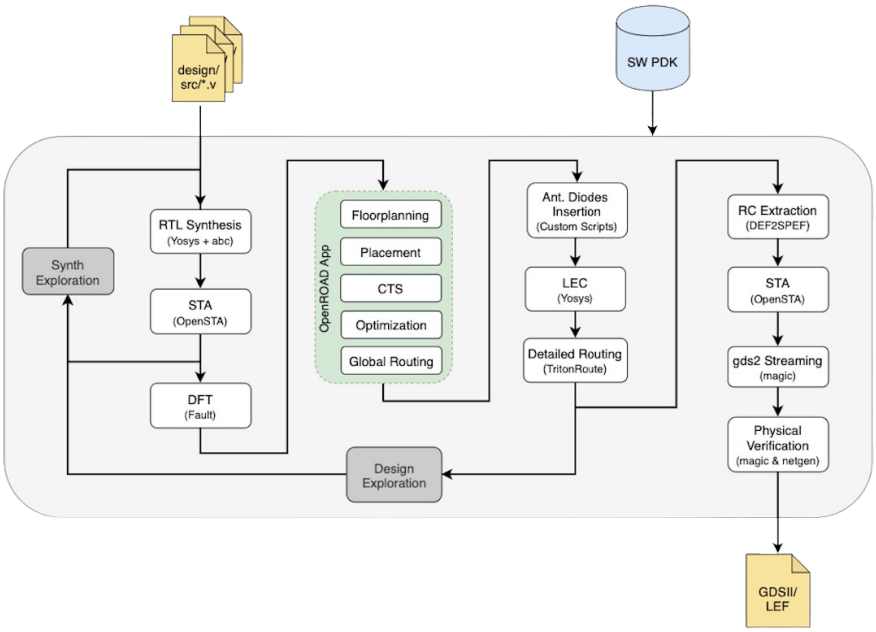


Figure 4. The OpenLane tool executes the RTL to GDSII flow, commencing with synthesis facilitated by the *Yosys* tool, and culminating in physical verification utilizing the Magic layout tool [8].

Due to its high level of automation, *OpenLane* utilizes only one constraint file. This constraint file takes the form of a java script object notation *JSON* file, incorporating various configuration variables that directly influence the flow from *RTL* to *GDSII*.

The *OQPSK* modulator was crafted using the following configuration variables in the *OpenLane* process:

- *PDK = gf180mcuC.*
- *DESIGN_IS_CORE = 0.*
- *PL_BASIC_PLACEMENT = 0*
- *PL_TARGET_DENSITY = 0.45.*
- *FP_CORE_UTIL = 40.*

- *SYNTH_MAX_FANOUT = 4*.
- *RT_MAX_LAYER = Metal4*.
- *DIODE_INSERTION_STRATEGY = 4*.
- *FP_SIZING = absolute*.
- *DIE_AREA = "0 0 300 300"*.

The specified parameters guide the *IC* design process using the GlobalFoundries 180 nm mixed-signal technology (PDK = gf180mcuC). The design is identified as a macro because it will be instantiated in an *SoC* (DESIGN_IS_CORE = 0) and must be limited to its routing layer (RT_MAX_LAYER = Metal4); because the system is not simple, and has little logical density, the basic placement of cells is disabled (PL_BASIC_PLACEMENT = 0).

The target placement density is set to 0.45 (PL_TARGET_DENSITY = 0.45), reflecting how spread the cells would be on the area, with a specific core utilization of 40% (FP_CORE_UTIL = 40). The chip's die area is established as 300 µm by 300 µm (FP_SIZING = absolute, DIE_AREA = "0 0 300 300"). The above parameters were selected to have the lowest area usage and no routing errors.

The limitation of the fan-out, the number of gates connected to the output of the driving gate, helps to reduce the power consumption (SYNTH_MAX_FANOUT = 4); on the other hand, the diode insertion strategy helps to dissipate charges accumulated on metal (DIODE_INSERTION_STRATEGY = 4).

These parameters and a set of predetermined variables by *OpenLane* collectively guide the layout design, influencing placement, routing, and utilization to meet the specific requirements of the chip design in the designated technology. The variables presented above are the key to the creation of the *OQPSK* modulator at the layout level.

5.4. SoC Integration

Caravel, an *SoC* template developed by *Efabless* and implemented with both *SKY130* and *GF180MCU* Technologies, is structured into three distinct areas: the *harness frame* and two wrapper modules referred to as the *management area* and the *user area* [12]. The *harness frame* encompasses essential components such as a clocking module, delay locked loop (DLL), user ID, housekeeping serial peripheral interface (SPI), power-on reset (POR), and a general-purpose input/output port *GPIO* controller.

The *management area* functions as a controller, housing an *RISC-V* and its peripherals. Its primary role involves the configuration and control of the *user area*. In contrast, the *user area* boasts a silicon area measuring 2.92 mm by 3.52 mm, comprising a fixed count of 38 I/O pads, 128 Logic Analyzer (LA) signals, and four power pads.

The combined architecture of the *harness frame*, *management area*, *user area*, and the *OQSPK* modulator, previously hardened, is depicted in Figure 5. This illustration provides an overview of the top architecture of the *Caravel* framework [12], including the *OQSPK* and its connections.

After the development of the macro *OQPSK*, it was seamlessly integrated with *Caravel* following the same *Back-end* process with the difference that the constrain file followed the *Caravel* constraints. While *Caravel* features an *RISC-V* microprocessor, the modulator was directly connected to the *GPIO* port.

In the concluding phase, the project underwent a series of prechecks to guarantee accurate integration and successful fabrication. Notably, the *OQPSK* modulator was designed using the 180 nm technology (*GF180MCU PDK*), marking one of the initial *ICs* developed with this specific *PDK*. For more details, the complete *OQPSK* project can be accessed at https://platform.efabless.com/projects/1685 (accessed on 15 April 2024).

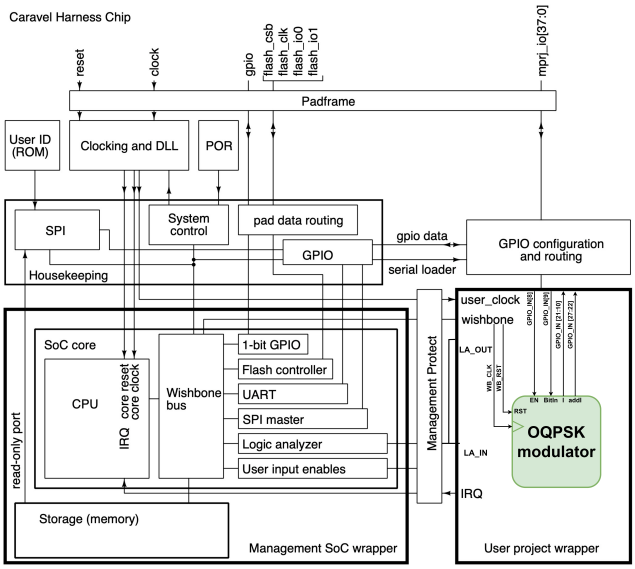


Figure 5. Caravel block diagram, divided into three sections: harness frame, management area, and user area, which instantiate the *OQPSK* modulator.

6. Results

6.1. FPGA Simulation Results

The simulation of the modulator was carried out to validate the functional behavior and verify the accuracy of the design before proceeding with its hardware implementation. This simulation stage was crucial for identifying and correcting logical errors, synchronization issues, and other potential faults at an early stage in the development process, thus ensuring that the modulator design met the required specifications and operated correctly under operational conditions. This proactive approach not only improved the reliability of the design but also established a solid foundation for more accurately predicting the expected outcomes in the physical measurements of the chip, thereby minimizing discrepancies between the simulated and actual results. For the simulation of the modulator, the construction parameters mentioned in Section 5.1 were taken into account.

As mentioned in Section 5.2, the bits intended for transmission were generated at a rate of 2 Mbps as long as the *En* and *rts* signals were activated. To visualize the waveform, sample values were collected, facilitating their graphical representation. Figure 6 displays the simulation results for each branch of the modulator. In the phase branch, the expected behavior is observed, whereas in Figure 7 in the quadrature branch, a variation is noted at the start of the signal formation. This discrepancy is due to the implementation of a delay, designed to introduce a phase shift that prevents deviations greater than 90°, which is essential for the proper functioning of the *OQPSK* modulation system.

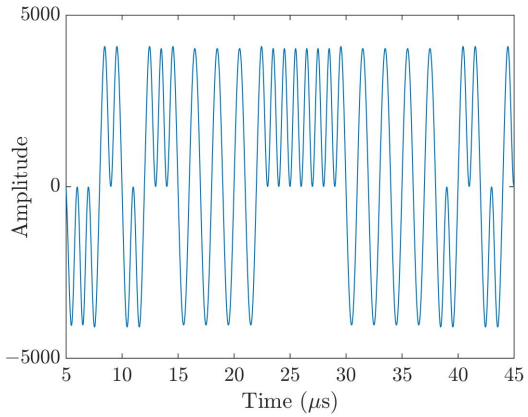


Figure 6. Simulation result of modulator with raised cosine shaping filter in phase branch.

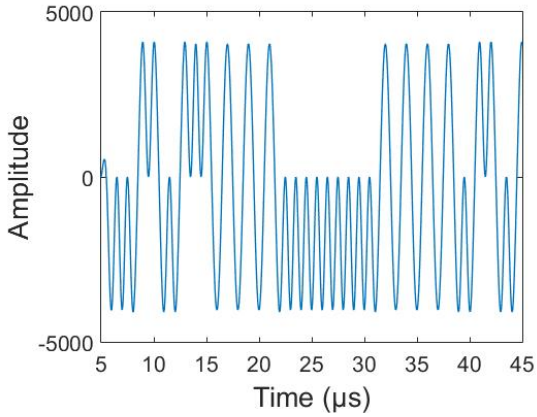


Figure 7. Simulation result of modulator with raised cosine shaping filter in quadrature branch.

FPGA Synthesis Results

The implementation of the modulator was carried out on the *Cyclone IV EP4CE115F29C7* FPGA from the Altera family using *Verilog HDL*, a popular choice for *IC* design due to its flexibility and efficiency. *FPGAs* are widely used in *ICs* for their ability to be reprogrammable, allowing designers to update functionality and fix issues without the need to redesign the hardware. Table 2 presents the results of the parameters obtained after synthesis, which are directly related to the utilization of the *FPGA*. These parameters include efficiency in the use of logic blocks, power consumption, and the total latency of the modulator system, among other important aspects. These are crucial for assessing the suitability and performance of the design within the *FPGA*, allowing us to optimize resource usage, improve energy efficiency, and minimize system response times. Furthermore, they provide a solid foundation for making informed decisions about possible adjustments and refinements in the design.

Table 2. Synthesis results for modulator with FPGA.

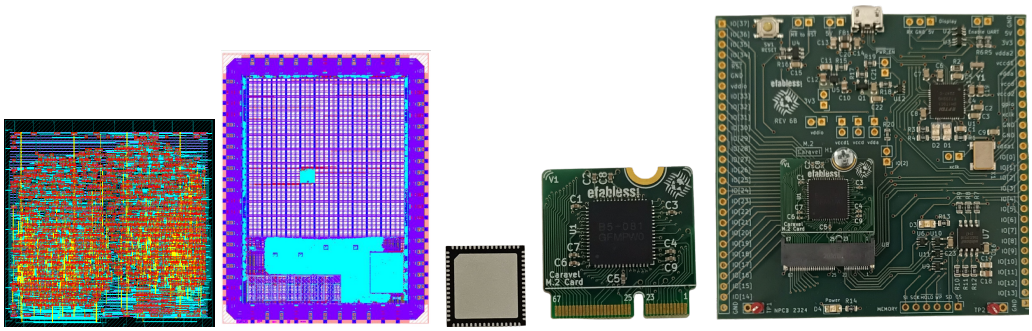
Parameter	EP4CE115F29C7
Total logic elements	780/114,480
Total registers	256/117,053
Total I/O pins	30/529
Total block memory bits	1664/3,981,312
Total logic array blocks	60/7155
Total power dissipation	144.25 mW
Core power dissipation	98.55 mW
I/O power dissipation	45.70 mW

6.2. IC Results

Following the *RTL* to *GDSII* flow, two *GDSII* files were generated. The first file encompassed a macro cell containing the *OQPSK* modulation system, as illustrated in Figure 8a. Meanwhile, the second file encapsulated the *core*, which was the *Caravel* template with the integrated *OQPSK* modulation, as depicted in Figure 8b. Table 3 shows the result obtained after the routing process.

Following the manufacturing process, we received 100 packaged parts, as seen in Figure 8c; 10 breakout boards, as seen in Figure 8d; and the assembly of two development boards, as seen in Figure 8e, each built according to predefined package and board designs.

In Appendix B, a table is included with performance metrics, energy consumption, and area for the *OQPSK* modulator previously developed using traditional tools provided by Cadence; it employs MOSIS 130 nm technology [29]. This implementation serves as a potential framework of reference for the readers, highlighting both the capabilities and limitations of the technology and methodologies used.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 8. View of the *GDSII* file depicting the *OQPSK* modulation (**a**), and the *Caravel SoC* (**b**), *OQPSK IC* (**c**), breakout board (**d**), *Caravel* M.2 developed board (**e**).

Table 3. Layout results for modulator using *GF180MCU PDK*.

Parameter	Quantity
Number of wires	1464
Number of logic cells	1051
Buffers	366
Clock buffers	49
Flip-Flops	24
Latches	4
Internal power	59 mW
Switching power	83.5 mW
Total power	142.5 mW
Worst slack (Setup)	17.85 ns
Worst slack (Hold)	0.54 ns

6.3. Physical Testing Results

Supplying a 5 V power voltage to the *IC*, physical measurements were carried out in *phase (I)*. This was due to the omission of both the *quadrature (Q)* output and the *MSB* of the phase during the manufacturing process. The omission of these aspects in the design was primarily due to restrictions related to the number of output pins available on the chip. Therefore, the decision was made to discard the previously mentioned bits to comply with these limitations.

The following laboratory equipment was used:

- D2-115 Board (Cyclone IV EPC11529C7 Altera).
- *Caravel* M.2 developed board.
- Agilent 16901A Logic Analyzer.
- Keithley 2200-3055 programmable power supply 30 V, 5 A.

Figure 9 shows the process flow, where the data generator, accompanied by its respective control signals, is loaded into the *FPGA*. Subsequently, the control signals are connected, establishing a direct link between the *FPGA* and the *IC*. The *IC* is powered with 5 V through the source Keithley 2200-3055. In the final stage, the *Data_Out* output signal from the *IC*, which corresponds to the *phase* of the modulated signal, is coupled to the logic analyzer, facilitating the observation of the waveform by graphing the obtained data.

In Figure 10a, the result obtained from the physical measurement in the phase stage is shown, illustrated through a graph. This graph specifically demonstrates the performance of the *raised cosine filter*, emphasizing the behavior observed in the time domain during the measurement. It is worth noting that, due to the omission of the *MSB*, an output of only positive values is observed. On the other hand, Figure 10b illustrates the frequency domain, emphasizing how the bandwidth of the signal filtered with the *raised cosine* is concentrated in the center, occupying less space than the signal processed with another filter.

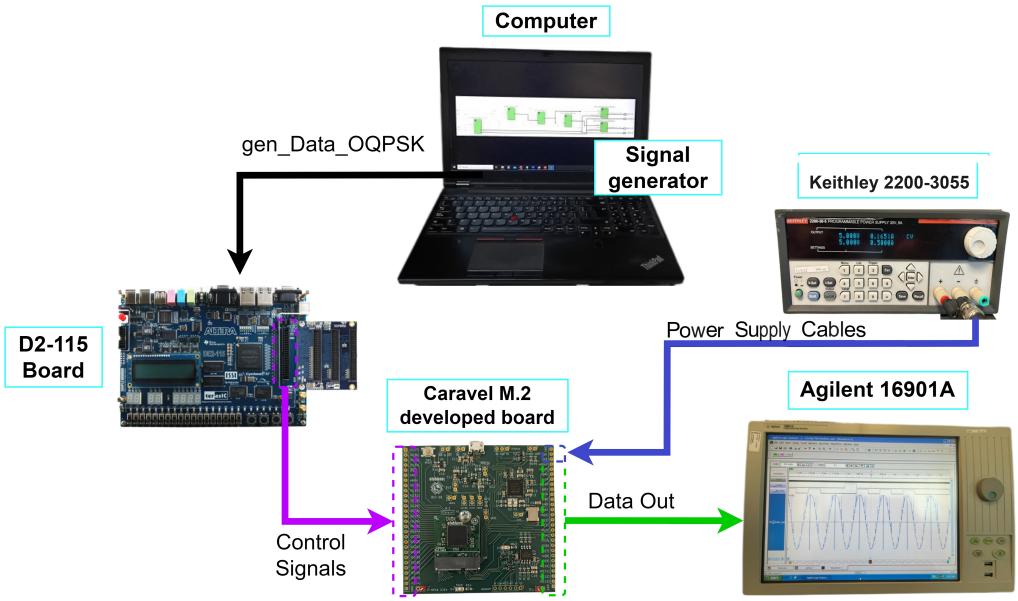
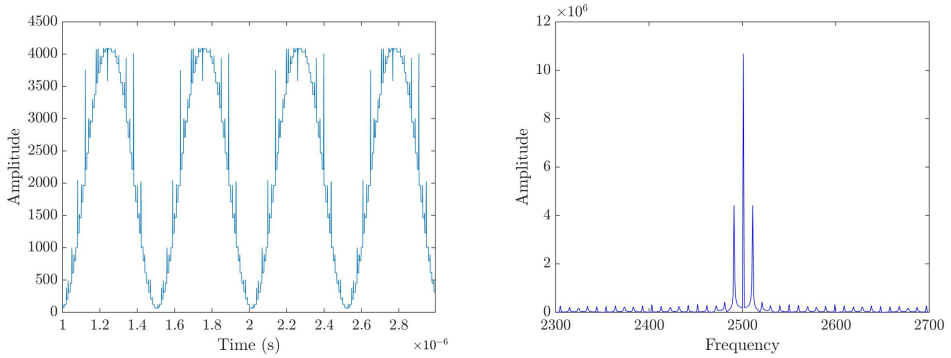


Figure 9. Physical connection for chip testing.



(**a**) (**b**)

Figure 10. Waveform obtained with the shaping filter in the time domain (**a**); waveform in the frequency domain of the raised cosine filter (**b**).

As previously observed, according to the output reports generated through *OpenLane*, the power consumption of the *OQPSK* modulator is expected to be 142.5 mW. The power consumption data obtained from physical measurements are detailed in Table 4. Here, the remaining power consumption can be attributed to *Caravel* and its components.

Table 4. Power consumption results.

Circuit Name	Voltage	Current	Power
Caravel developed board + Breakout board	5 V	165 mA	825 mW
Caravel developed board	5 V	9.9 mA	49.5 mW
Breakout board	5 V	155 mA	775 mW

7. Discussion

The omission of the *MSB* of the *phase*, particularly the sign bit, in an *IC*, has significant implications for the precision and functionality of the electronic system in which it is implemented. Firstly, the absence of the sign bit compromises the *IC's* ability to accurately represent negative values in *phase* measurements, resulting in a biased data representation. This can affect applications that rely on precise *phase* measurement, such as digital

communication systems and signal processing, where phase integrity is crucial for signal demodulation and decoding.

On the other hand, the lack of *quadrature* output limits the system's ability to perform complex signal processing operations, such as modulation and demodulation in advanced communication systems, which require both I and Q components for complex signal representation. The absence of one of these components forces the use of alternative methods for signal reconstruction, which can increase system complexity and reduce overall efficiency.

As previously mentioned, during the manufacture of the modulator, the *MSB* was omitted, resulting in an output of only positive values. To facilitate an adequate comparison between the results of the simulated and obtained measurements, a post-processing data algorithm was implemented with the objective of compensating for the omission of the sign bit and thus correct the affected signal.

The implemented process presents a sequence of actions we carried out, taking into account that the input bit rate was known:

- The first step was to divide the bit rate into two parts to simulate the operation of a serial-to-parallel converter. This operation effectively isolated a part of the signal that corresponded to the phase input bits.
- Knowing the duration of each input bit and the expected data from the signal (through simulation), it proceeded to identify if, for each time interval assigned to a bit, the signal was positive or negative.
- This process was carried out sequentially, facilitating the restoration of the signal to be equivalent to the simulated one. In this way, it was possible to recover the integrity of the phase representation that was compromised due to the initial omission of the sign bit.

This methodology allowed for compensating the loss of critical information and ensured a faithful representation of the desired signal, which is essential for the accuracy of applications that depend on precise phase measurements.

After post-processing the wave signal obtained during the measurement, a comparison was made with the expected signal in the time domain, with the results presented in Figure 11.

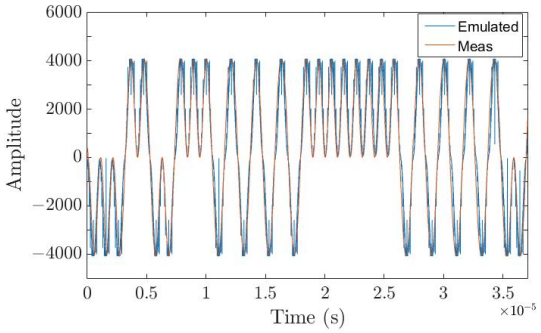


Figure 11. Comparison between the expected signal and the post-processed measured signal.

In the same way, the comparative graph of the signals in the frequency domain was developed, resulting in the graph presented in Figure 12.

Looking towards the future, it is anticipated that the use of these tools will expand, opening new possibilities for innovations in the field of electronics and circuit design. Consequently, an adjustment was made in the allocation of output pins for the *OQPSK* modulator detailed in this study, leaving the necessary output data without compromising the signal again. We are waiting for the next round of free manufacturing to begin to participate and take advantage of the benefits it offers. Similarly, we participated again in the round of free *IC* manufacturing, using open source tools in asynchronous circuits with the synchronous *OQPSK* modulator as a starting point, with the goal of more accurately evaluating the advantages and disadvantages inherent to synchronous versus asynchronous circuits. We emerged as winners [30] and are currently awaiting the delivery of the new

chip. This approach will not only allow us to leverage the benefits of this technology but will also contribute to the development and refinement of the design and manufacturing processes of digital *ICs*.

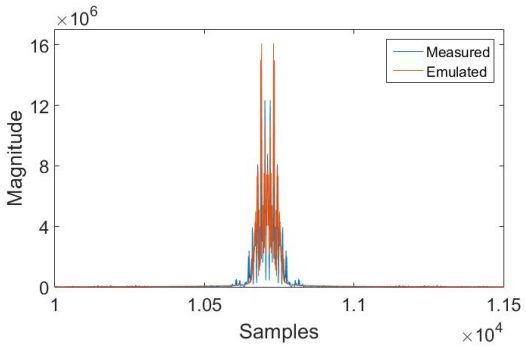


Figure 12. Comparison between the expected signal and the post-processed measured signal. Comparison between the expected signal and the post-processed measured signal in the frequency domain.

8. Conclusions

The design process for digital *ICs* was approached using *open source EDA* tools, focusing on the development of an *OQPSK* modulator with *Verilog HDL* and a *raised cosine filter* to optimize modulation. The *GF180MCU 180 nm PDK* from *Global Foundries* was employed, integrated with *OpenLane*, which significantly improved the design and implementation stages.

Caravel facilitated the manufacturing of the modulator through an open wafer service, marking a milestone in manufacturing with this *PDK*. After manufacturing, essential physical tests were conducted to confirm the design's objectives and parameters, thus demonstrating the effectiveness of *open source EDA* tools in advanced *IC* design.

Although they are free-to-use tools, their workflow is the same as that used by commercial tools. This makes them ideal for both educational and training environments, as they offer a valuable platform to learn and apply the fundamental principles of *IC* design without incurring the high costs associated with commercial software licenses. This approach not only facilitates access to *IC* design technology but also provides training for students and professors in the skills and practical knowledge necessary to work in the industry.

Author Contributions: Conceptualization, E.M.-G. (Emma Mascorro-Guardado), L.A.L.-R. and S.O.-C.; methodology, E.M.-G. (Emma Mascorro-Guardado), L.A.L.-R., E.I.B.-L., U.J.-T. and S.O.-C.; software, E.M.-G. (Emma Mascorro-Guardado), U.J.-T. and S.O.-C.; validation, E.M.-G. (Emma Mascorro-Guardado), E.I.B.-L., U.J.-T. and S.O.-C.; formal analysis, E.M.-G. (Emma Mascorro-Guardado), L.A.L.-R., E.I.B.-L. and U.J.-T.; investigation, E.M.-G. (Emma Mascorro-Guardado), E.I.B.-L., L.A.L.-R. and U.J.-T.; resources, S.O.-C.; data curation, E.M.-G. (Emma Mascorro-Guardado), E.I.B.-L. and L.A.L.-R.; writing—original draft preparation, E.M.-G. (Emma Mascorro-Guardado); writing—review and editing, E.M.-G. (Emma Mascorro-Guardado), L.A.L.-R., E.I.B.-L., U.J.-T., S.O.-C., E.M.-G. (Emanuel Murillo-García) and M.H.-A.; supervision, L.A.L.-R. and S.O.-C.; project administration, S.O.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. *OQPSK* Modulator at *RTL* Level

The *RTL* of the digital stage of the *OQPSK* modulator is shown in Figure A1. The block that performs the filtering is included within the *pulse shaping* module, which stores the quantization values of the shaping filter, as well as the sign change operations.

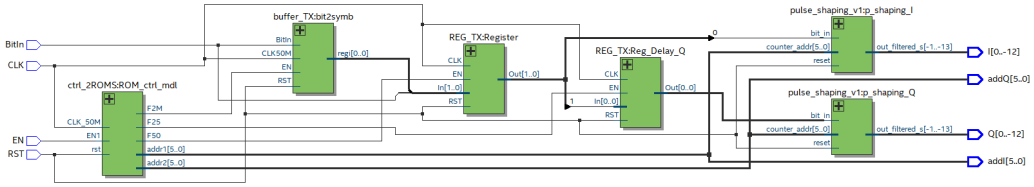


Figure A1. *RTL* diagram of *OQPSK* modulator with raised cosine shaping filter implemented.

Appendix B. Metrics of the *OQPSK* Modulator with Different Tools

Comparing ASIC designs implemented on different silicon nodes and with open source tools versus traditional solutions, based on specific technology metrics, often turns out to be an uneven and uninformative practice. The most recent manufacturing methods offer significant benefits in speed, area footprint, and energy efficiency compared to larger scale technologies. Table A1 shows the results of this implementation compared to an implementation with commercial tools.

Table A1. Performance metrics for *OQPSK* modulator.

Design	Year	Technology	Tools	Area (μm^2)	Power (mW)	Worst Slack (ns)
This work	2024	GF 180 nm	*OpenLane*	0.09	142.5	17.85
[29]	2019	MOSIS 130 nm	Cadence tools	-	0.052	1.94

References

1. Lojek, B. *History of Semiconductor Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
2. Trlica, C. Software Tools [Technology Analysis and Forecast]. *IEEE Spectr.* **1997**, *34*, 60–64. [CrossRef]
3. Ousterhout, J.K.; Hamachi, G.T.; Mayo, R.N.; Scott, W.S.; Taylor, G.S. The magic VLSI layout system. *IEEE Des. Test Comput.* **1985**, *2*, 19–30. [CrossRef]
4. Wolf, C.; Glaser, J.; Kepler, J. Yosys-a free Verilog synthesis suite. In Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip), Linz, Austria, 10 October 2013; Volume 97.
5. Ajayi, T.; Blaauw, D. OpenROAD: Toward a self-driving, open source digital layout implementation tool chain. In Proceedings of the Government Microcircuit Applications and Critical Technology Conference, Albuquerque, NM, USA, 25–28 March 2019.
6. Welcome to SkyWater SKY130 PDK's Documentation! Available online: https://skywater-pdk.readthedocs.io/en/main/ (accessed on 11 November 2023).
7. Welcome to GlobalFoundries 0.18UM 3.3V/(5V)6V MCU PDK's Documentation! Available online: https://gf180mcu-pdk.readthedocs.io/en/latest/ (accessed on 11 November 2023).
8. Ghazy, A.; Shalan, M. OpenLANE: The open source digital ASIC implementation flow. In Proceedings of the Workshop on Open Source EDA Technologies (WOSET), Online, 2–5 November 2020. Available online: https://woset-workshop.github.io/WOSET2020.html (accessed on 15 April 2024).
9. Genko, N.; Atienza, D.; Micheli, G.D.; Benini, L. Feature—NoC emulation: A tool and design flow for MPSoC. *IEEE Circuits Syst. Mag.* **2007**, *7*, 42–51. [CrossRef]
10. Kinsy, M.A.; Pellauer, M.; Devadas, S. Heracles: A tool for fast RTL-based design space exploration of multicore processors. In Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, USA, 11–13 February 2013; pp. 125–134.
11. Zulberti, L.; Di Matteo, S.; Nannipieri, P.; Saponara, S.; Fanucci, L. A script-based cycle-true verification framework to speed-up hardware and software co-design: Performance evaluation on ecc accelerator use-case. *Electronics* **2022**, *11*, 3704. [CrossRef]
12. Caravel Harness. Documentation. Available online: https://Caravel-harness.readthedocs.io/en/latest/ (accessed on 10 November 2023).
13. MOSIS. Available online: https://themosisservice.com/ (accessed on 29 February 2024).
14. CMC Microsystems—Home. Available online: https://www.cmc.ca/ (accessed on 29 February 2024).
15. EUROPRACTICE—IC Service. Available online: https://europractice-ic.com/ (accessed on 29 February 2024).
16. Efabless.com. Available online: https://efabless.com/ (accessed on 29 February 2024).
17. Proakis, J.G.; Salehi, M. *Digital Communications*, 5th ed.; McGraw-Hill: New York, NY, USA, 2007.
18. *Std 802.15.4-2003*; Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPAN's). IEEE Computer Society LAN/MAN Standards Committee: New York, NY, USA, 2003.
19. Guthaus, M.R.; Stine, J.E.; Ataei, S.; Chen, B.; Wu, B.; Sarwar, M. OpenRAM: An open-source memory compiler. In Proceedings of the 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 7–10 November 2016; pp. 1–6. [CrossRef]

20. Buhr, S.; Kreißig, M.; Ellinger, F. A Real-Time Fast Ethernet Transceiver achieving Sub-ns Time Synchronization. In Proceedings of the 2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC), Helsinki, Finland, 29–30 October 2019; pp. 1–7. [CrossRef]
21. Buhr, S.; Xu, X.; Kreißig, M.; Ellinger, F. A Low Power Fast Ethernet Physical Layer Transceiver. In Proceedings of the 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 27–29 November 2019; pp. 478–481. [CrossRef]
22. Zhang, Q.; Duan, W.; Edwards, T.; Ansell, T.; Blaauw, D.; Sylvester, D.; Saligane, M. An Open-Source and Autonomous Temperature Sensor Generator Verified With 64 Instances in SkyWater 130 nm for Comprehensive Design Space Exploration. *IEEE Solid State Circuits Lett.* **2022**, *5*, 174–177. [CrossRef]
23. Franck, L.D.; Ginja, G.A.; Carmo, J.P.; Afonso, J.A.; Luppe, M. Custom ASIC Design for SHA-256 Using Open-Source Tools. *Computers* **2024**, *13*, 9. [CrossRef]
24. Shah, M.; Ternes, J.; Koch, D. FABulous Demo: Open Source FPGA on Sky130. In Proceedings of the 33rd International Conference on Field-Programmable Logic and Applications (FPL), Gothenburg, Sweden, 4–8 September 2023; pp. 365–365. [CrossRef]
25. Xiong, F. *Digital Modulation Techniques*; Artech House: Norwood, MA, USA, 2000.
26. Kontorovich-Mazover, V.; Ramos-Alarcon, F.; Parra-Michel, R. *Fundamentos de Comunicaciones Digitales*, 1st ed.; Limusa: Mexico City, Mexico, 2009.
27. Luna-Rodriguez, L.A.; Ortega-Cisneros, S.; Mascorro-Guardado, E.; Rivera-Dominguez, J.; Parra-Michel, R. Implementation methodology for Self-Timed modulators with dual-rail structures based on FPGA. In Proceedings of the 2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 14–16 November 2018; pp. 1–6. [CrossRef]
28. Han, K.; Kahng, A.B.; Li, J. Optimal Generalized H-Tree Topology and Buffering for High-Performance and Low-Power Clock Distribution. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2020**, *39*, 478–491. [CrossRef]
29. Guardado, E.M.; Ortega-Cisneros, S. Metodología para la Implementación de CI Digitales a Nivel Layout. 2019. Available online: https://unidad.gdl.cinvestav.mx/plantillaTesis.php?Tesis=1405 (accessed on 15 April 2024).
30. Winners of the *ICs* Manufacturing Round GFMPW-1. Available online: https://docs.google.com/spreadsheets/d/1qxy1wK-fbegUjC_IC2k6f9NfbdSS9gevUzcF7rFQnfA/edit?usp=sharing (accessed on 15 April 2024).