UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

TESTBENCH DEVELOPMENT OF A SIGMA-DELTA ADC CONVERTER CHIP

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Izan García Cutillas

In partial fulfilment of the requirements for the

MASTER'S DEGREE IN ELECTRONIC ENGINEERING

Advisor: Daniel Fernández Martínez

Co-Advisor: Jordi Madrenas Boadas

Barcelona, May 2023

Para Mª José, Ángel, Cristina, Ricardo, Daniel y Adrián: mi madre, mi padre, mi novia y mis amigos. Por su paciencia y apoyo incondicional.

Acknowledgments

I would like to thank all the support that my advisor, Daniel, and co-advisor, Jordi, have given me during all these months. For giving me the opportunity to carry out a project as complete and interesting as this one. Without your guidance and help this would not have been possible. Thank you Daniel for your enormous patience and quick help at all times.

Abstract

Currently, there are several techniques to verify the operation of mixed-signal integrated circuits, however, most require that the chip itself contains the necessary hardware for it. If the verification circuitry is not included, one solution is to test its operation externally. This approach is the one followed in the project.

In this work has been developed a digital design to be embedded in a FPGA in order to verify the operation of the shift registers contained in a chip that was developed by the IS2 UPC research group. These registers are fundamental as they are intended to send data to all the blocks of the chip from off-chip. The created design is capable of generating data, transmitting it to the registers and controlling their operation by generating the appropriate signals. The design is very versatile since it allows adapting its operation according to the needs of the test by varying single parameters. In addition, the created design can be used not only as a test, but also to send and receive data to the internal blocks of the chip.

This work also includes the design and implementation of a PCB to act as a physical interface between the chip to be tested and the control FPGA. Furthermore, extra circuitry is added to the PCB in order to transmit or receive analog signals to or from one of the main blocks of the chip: the sigma-delta analog-to-digital converter.

Index

List of figures

List of tables

1. INTRODUCTION

It is essential to verify and validate the proper performance of any built prototype to ensure that the design provides the functionalities for which it was designed. Integrated circuits are no exception. In this case, chip simulation and verification is often much more vital than in other fields due to the complexity of design and manufacturing, usually, requiring the development of specific designs and the use of complex simulation tools. In addition, as in all verification and simulation, it is convenient to know in detail the operation and characteristics of the device under test in order to verify accurately the functionalities to be tested.

1.1. Project description

This project is aimed to verify the operation of some blocks contained into a CMOS-MEMS (Complementary Metal-Oxide Semiconductor - Micro Electronic Mechanical System) chip which was designed by a research group. Concretely, the operation of 15 SIPO (Serial-In, Parallel-Out) shift registers and 1 PISO (Parallel-In, Serial-Out) shift register. These registers are used to transfer data to other blocks of the chip or, as in case of the PISO, to output data from these blocks. The SIPOs and PISO blocks are very necessary because they greatly reduce the number of external inputs and outputs, therefore, reducing also the amount of pads and, consequently, the manufacturing cost. Serial data is converted in parallel data in case of the SIPOs and transmitted to the required block, and converted from parallel to serial data in the case of the PISO to be output outside the chip.

In this project, it will be created a digital design that allows these blocks to be controlled to verify their operation. Firstly, only the created design will be tested and, secondly, the design will be simulated together with the chip. This second step will allow verifying deeper the design created but, mainly and most important, it will allow testing the SIPOs and PISO operation in the simulation. Verifying their correct operation will allow knowing if it is possible to reliably access the rest of the blocks of the chip. At this point, it will be required to design a PCB (Printed Circuit Board) to use it as interface between the chip and a FPGA (Field Programmable Gate Array), where the created design will be implemented. Actually, this PCB will not only be used as an interface but should have more functionalities. For example, some additional circuitry will also be added to process signals to be transmitted to the internal ADC (Analog-to-Digital Converter) or to externally process externally signals coming from this ADC. Another example would be the incorporation of a temperature sensor. Finally, the complete system will be tested physically in the lab implementing the created design in a FPGA to send data to the shift registers and receive this data from them.

To sum up, the project consist of three main phases:

1) Design of the testing blocks, simulation and verification.
2) PCB design.
3) Physical testing.

1.2. Objectives

The objectives set to develop the project are broken down below:

- Design of the necessary digital blocks to control and test the operation of the SIPOs and PISO blocks.
- Simulate the behaviour of the designed testing blocks.
- Simulate the design created together with the chip simulation to verify the correct behaviour of the SIPOs and PISO.
- Design the schematics to be implemented in the PCB.

- Selection of components based on technical, physical and economic criteria.
- PCB design, implementation and verification.
- Design of a test bench to generate the input stimuli and read the data received from the shift registers.
- Physical verification of the complete system: Chip-PCB-FPGA.

1.3. Design requirements

For the development of the testing blocks, it should be met some design requirements:

- Operation frequency: It should be possible to configure the communication clock to operate on different frequencies so that the communication speed can be changed if desired.
- Chip-FPGA delay: Since there will be some elements placed between the chip and the FPGA, it is important to design in such a way that the FPGA-chip communication is not affected by the delay, whatever it may be.
- Conditions to trigger communication between the design made and the chip: It will be seen that the initiation or abort of communications must be governed either by the input data or by the value of an enable signal, depending on whether a SIPO or PISO block is being tested.
- Components selection for the PCB: The choice of components must be made considering the signals with which they will deal.

1.4. Chip content

One of the main components of the chip is the sigma-delta analog-to-digital converter. It is used to digitalise the signal produced by another important block of the chip: the accelerometer. In this way, the acceleration value can be processed in the FPGA. This converter can also be used as a standalone ADC to convert analog signals coming from off-chip because the chip has some pins for this purpose. A low-noise amplifier is also included to amplify the signal produced by the accelerometer in order to drive the ADC. This is needed because the output signal generated by the accelerometer has a small voltage range that should be increase.

These are the most significant blocks contained in the chip, but not the only ones. To generate all the voltage references for all the blocks of the chip, a bandgap voltage reference is used. Its fundamental characteristic is that it is temperature independent, generating constant voltage references regardless chip temperature variations. Due to this important advantage, it is a circuit widely used in integrated circuits.

To transmit data and configure these blocks is needed to use the already mentioned SIPO registers. The SIPOs are organised into 5 groups based on their purpose, these groups are: BG (Bandgap), DFT (Design For Test), ADC (Analog-to-Digital Converter), FE (Front-End) and FECK (Front-End Clock). The BG SIPOs are intended to configure the bandgap voltage reference circuit. The DFT SIPO is used to activate chip operation modes to monitor some internal signals of the chip. The ADC SIPOs are used to set the configuration of the ADC. The FE SIPOs manage the signals of the low-noise amplifier configuration. The FECK SIPO manages the clock signals for the signals used configure the amplifier, i.e. the FE signals.

Finally, it is important to mention the PISO. The data provided by this register comes from the ADC output and it is used to reconstruct this output off-chip (in the FPGA) and obtain the acceleration value.

1.5. Work plan

This section describes the tasks into which the project is broken down and the duration of each of them. The Gantt diagram shows the time dedicated to each task and the expected dedication to each of them throughout the duration of the project.

1.5.1. Work breakdown structure

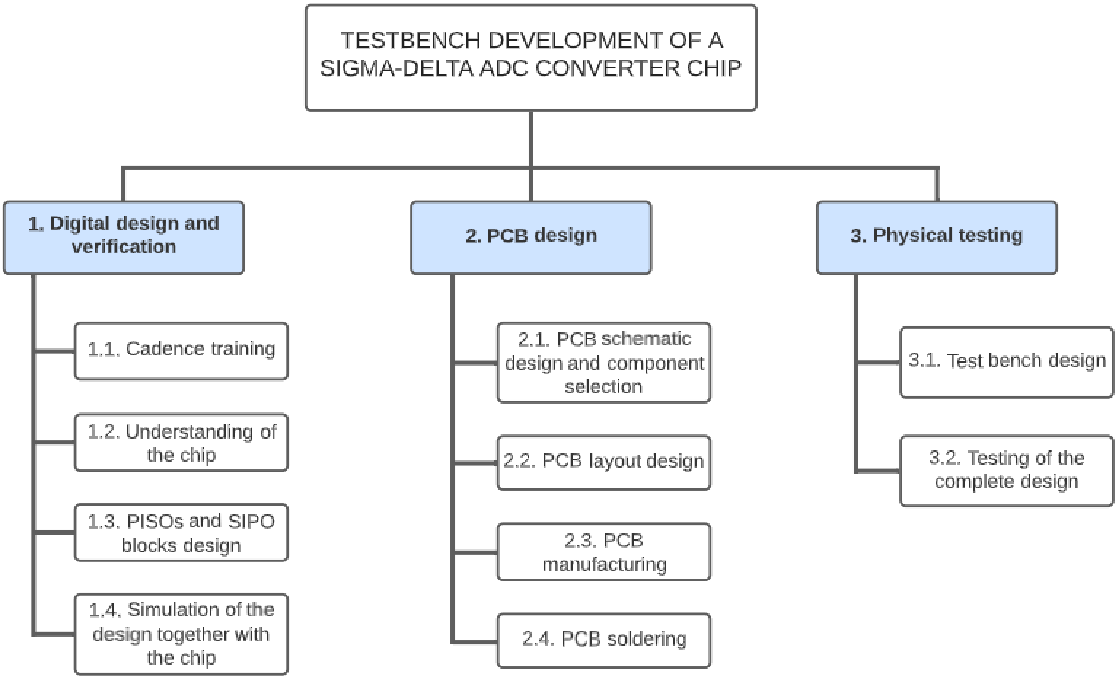

TESTBENCH DEVELOPMENT OF A SIGMA-DELTA ADC CONVERTER CHIP
1. Digital design and verification
2. PCB design
3. Physical testing
1.1. Cadence training
1.2. Understanding of the chip
1.3. PISOs and SIPO blocks design
1.4. Simulation of the design together with the chip
2.1. PCB schematic design and component selection
2.2. PCB layout design
2.3. PCB manufacturing
2.4. PCB soldering
3.1. Test bench design
3.2. Testing of the complete design

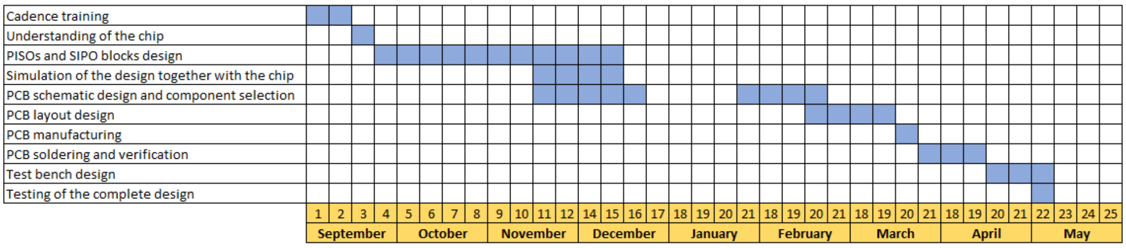
Figure 1. Work breakdown structure

1.5.2. Work packages description

Group	Task	Description	Duration (weeks)
Digital design and verification	Cadence training	A Cadence course was carried out to improve knowledge. This course contained videos on Virtuoso ADE Explorer and Assembler and on Virtuoso Schematic Editor.	2
	Understanding of the chip	The design of the chip and the circuits that comprised it were reviewed, and more specifically, the blocks that had to be tested in the project.	1
	PISOs and SIPO blocks design	The PISOs blocks and the SIPO block were designed to verify the operation of the SIPOs and the PISO of the chip, respectively.	11
	Simulation of the design together with the chip	Using analog-mixed signal simulations, the correct operation of the designed blocks and the blocks of the chip to be tested was simulated and verified.	4
PCB design	PCB schematic design and component selection	In addition to the connections between the FPGA and the chip, the design of the analog circuits that were implemented on the PCB was carried out. The selection of the components was also done.	9
	PCB layout design	PCB design was done.	4
	PCB manufacturing	The PCB was manufactured and the purchase of the components was made.	1
	PCB soldering and verification	The PCB was completed by soldering all its components and verifying their correct operation.	3
Physical testing	Test bench design	To send data to the SIPOs and receive this data from them to verify its operation	3
	Testing of the complete design together with the chip	The complete system was tested to verify the correct operation of the chip (Chip-PCB-FPGA).	1

Table 1. Description of project tasks

1.5.3. Gantt diagram



Cadence training	■	■																																		
Understanding of the chip			■																																	
PISOs and SIPO blocks design				■	■	■	■	■	■	■	■	■	■	■																						
Simulation of the design together with the chip											■	■	■	■																						
PCB schematic design and component selection											■	■	■	■	■					■	■	■	■													
PCB layout design																							■	■	■	■										
PCB manufacturing																											■									
PCB soldering and verification																												■	■	■						
Test bench design																															■	■	■			
Testing of the complete design																																	■			
	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	17	18	19	20	21	18	19	20	21	18	19	20	21	18	19	20	21	22	23	24	25
	September				October				November				December				January				February				March				April				May			

Figure 2. Gantt diagram

Note that for 4 weeks there was a break in the realization of the project due to reasons unrelated to the project.

2. STATE OF THE ART

In this section, several integrated circuit testing techniques are explored.

2.1. Boundary Scan [1]

The integrated circuit boundary scan test is a technology that consists in placing inside the integrated circuit 1 bit register for every pin, between internal logic and physical pin. With this register it is possible to control and observe any pin of the integrated circuit by serially shifted in and out in shift register manner. The standard from the Institute of Electrical and Electronics Engineers is IEEE1149.1 Boundary scan standard.

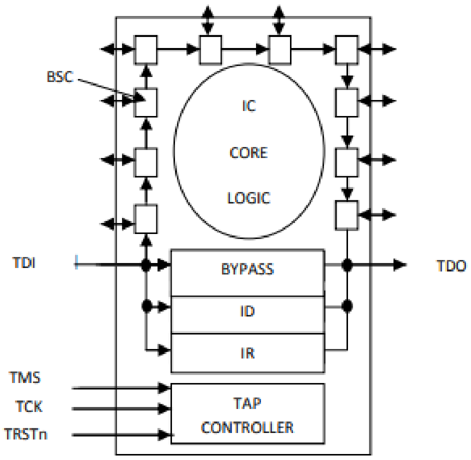


Figure 3. Boundary scan circuit [1]

In addition to the boundary scan cell per pin, a device that complies IEEE1149.1 also contains a bypass register, an identification register, an instruction register and a Test Access Port (TAP) controller. The TAP contains 4 or 5 pins Test-Data-In (TDI), Test-Data-Out (TDO), Test-Mode-Select (TMS), Test-Clock (TCK) and optional Test-Reset (TRSTn) [2]. TAP controller has the capability of serially shifting the data from TDI to TDO, updating/capturing the data to/from the pins/internal logic from/to the Boundary scan cell.

The advantage of this method is that it allows reducing the amount of test point, reducing the size of the PCB. Furthermore, it reduces the need to use probes for observe every pin.

2.2. Built-in self-test (BIST) [3]

This technique consists of designing additional hardware and software features into integrated circuits to allow them to perform self-testing [4]. BIST have been designed to meet requirements such as reliability, cost and reduced complexity. Both the test cycle duration and the number of input/output signals that must be driven and examined under test control are reduced. This two facts have a big impact in the reduction in complexity and cost of testing.

BIST is also the solution for testing critical circuits that do not have direct connections to external pins, such as embedded memories used internally by the devices. Currently, BIST is widely used to test memories. In the near future, even the most advanced tester may no longer be adequate for the fastest chip, then, self-testing may be the best solution for that [4].

For industrial purposes, the integrated circuits are tested both after manufacturing and periodically in the field. Testing verifies not only the correctness of the fabrication process. It guarantees for the correctness of the function and fault free design. In addition, testing can also be used during the latter stages of the product lifecycle, in order to detect errors due to aging, environment or other factors.

Now, it is presented the basic architecture of BIST:

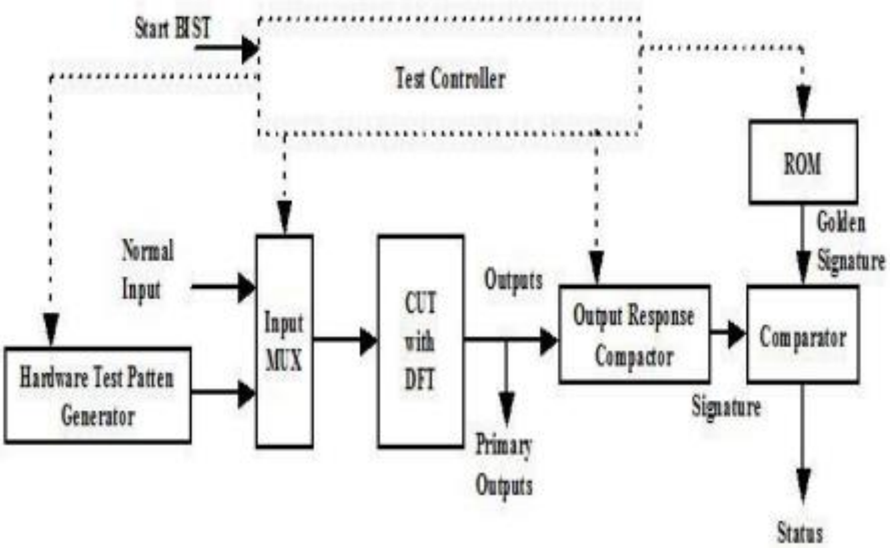


Figure 4. BIST architecture [3]

The test pattern generator is basically a type of register that synthesizes the pseudo-random patterns which behave as test patterns. As the area is limited, the storage and the generation of the test pattern is done by ATPG (Automatic Test Pattern Generation) algorithms on the CUT (Circuit Under Test).

The input multiplexer allows to select normal inputs into the circuit when it is functional and test inputs from the pattern generator when BIST is executed. This selection is controlled by the test controller.

The output response comparator compacts the output response of the CUT to the test patterns into a signature which is compared to the expected signature for the fault-free circuit, called golden signature. This golden signature is stored in the ROM (Read-Only Memory). Both signatures are compared in the comparator block and, if a fault is detected, the status line becomes high.

The test controller starts the BIST procedure to control the BIST once it receives the signal start BIST. Once the test is finished, the controller connects normal inputs to the CUT via the multiplexer, this making it ready for operation.

2.3. Scan chain

The use of scan design techniques to efficiently test and debug sequential circuits is widely used in industrial practice [5]. In scan design, some or all sequential elements or flip-flops in a circuit are linked into one or more scan chains. A scan chain is formed by a number of flip-flops connected back to back in a chain with the output of one flip-flop connected to another. The input of the first flip-flop is connected to the input pin of the chip (called scan-in) from where scan data is fed. The output of the last flip-flop is connected to the output pin of the chip (called scan-out) which is used to take the shifted data out [6]. This permits data to be serially shifted into and out of these flip-flops, greatly enhancing the controllability and observability of internal nodes in the design [5]. Scan chain structure accounts for 10% to 30% of die area [7], and nearly 50% of the chip failures are caused by scan chains [8].

The figure below shows how a scan chain is:

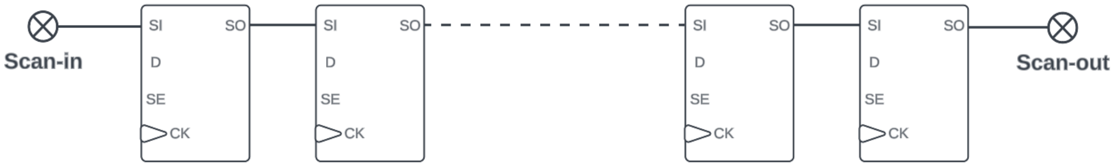


Figure 5. Scan chain circuit [6]

Note as the flip-flops in the design have been modified. The normal input (D) has to be multiplexed with the scan input. A signal called scan-enable is used to control which input is propagated to the output [6]:

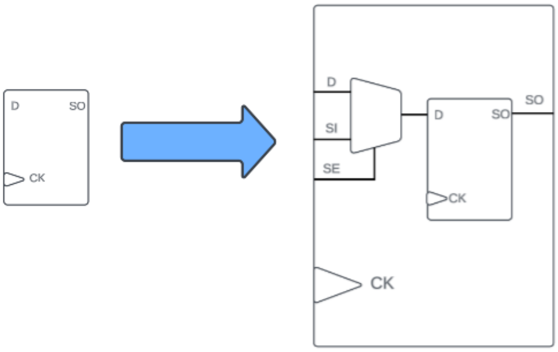


Figure 6. Flip-flop modification for scan chain technique [6]

Scan testing is carried out for several reasons, the two most important are [6]:

- To test stuck-at faults in manufactured devices
- To test the paths in the manufactured devices for delay

2.4. Ad-hoc design FPGA-based

The 3 previous techniques imply having the elements for testing physically implemented on the chip. The chip to be tested was already built and does not contain elements that allow testing the SIPOs and PISO, hence, it is not possible to use these techniques. That is why carrying out an ad-hoc design to test the specific elements of the chip becomes the only and best solution. Since the elements of the chip to be tested are digital blocks, using a FPGA is the best option since FPGAs allow creating digital circuits easily.

FPGAs are reprogrammable devices, based on configurable logic block arrays whose connections can be modified according to the designer's needs. Their reprogramming capacity, together with a less costly and lengthy development and manufacturing process than that of other devices, gives FPGAs a privileged position for use in a wide variety of applications compared to other integrated circuits [9]. An important factor in choosing FPGAs is the large number of programmable ports they contain. This fact makes them ideal devices for designs like the one developed in this project, in which there are a large number of inputs and outputs with specific functionalities. Unlike microcontrollers, whose hardware, including inputs and outputs, is already physically built by the manufacturer and cannot be modified.

In the laboratory there were 2 FPGAs available to implement the design: PYNQ-Z2 and ZedBoard Zynq-7000. The model chosen for the project was the Zedboard Zynq-7000 as it has a large number of usable GPIO (General Purpose Input/Output) pins compared to the PYNQ-Z2. Specifically, it has 5 PMOD (Peripheral Module) connectors in total, but only 4 of them can be used for programmable logic. Each PMOD has 8 usable pins, therefore, a total of 32 pins are available for programmable logic.

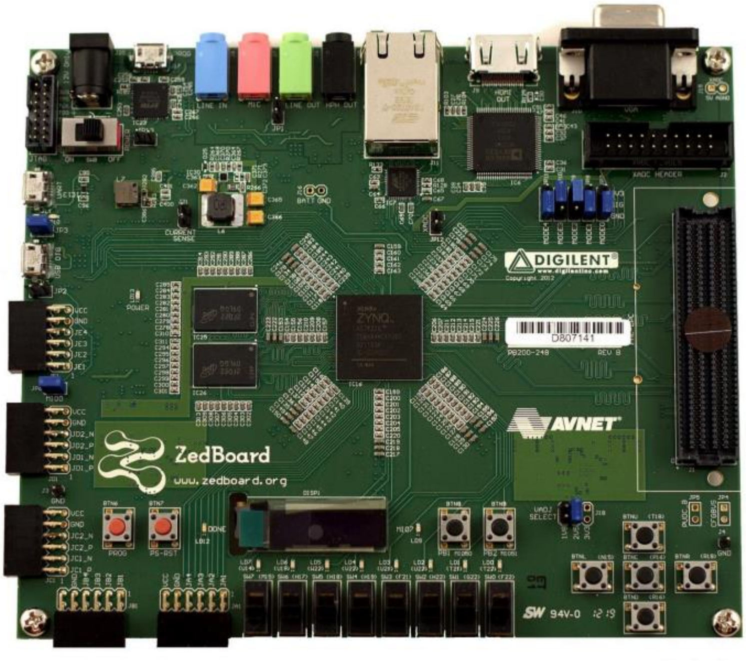


Figure 7. ZedBoard Zynq-7000 [10]

3. METHODOLOGY

This section contains in detail the entire design process carried out during the project. In addition to the designs created, the chip blocks that must be tested are presented to understand how the designs that verify them should be. The simulations that will be shown only verify the operation of the designs individually using the stimuli of test benches. On the other hand, the designs of the circuits implemented on the PCB and the choice of its main components will be also seen.

Below is a general block diagram to show an idea of the blocks to be designed and how they communicate with the chip:

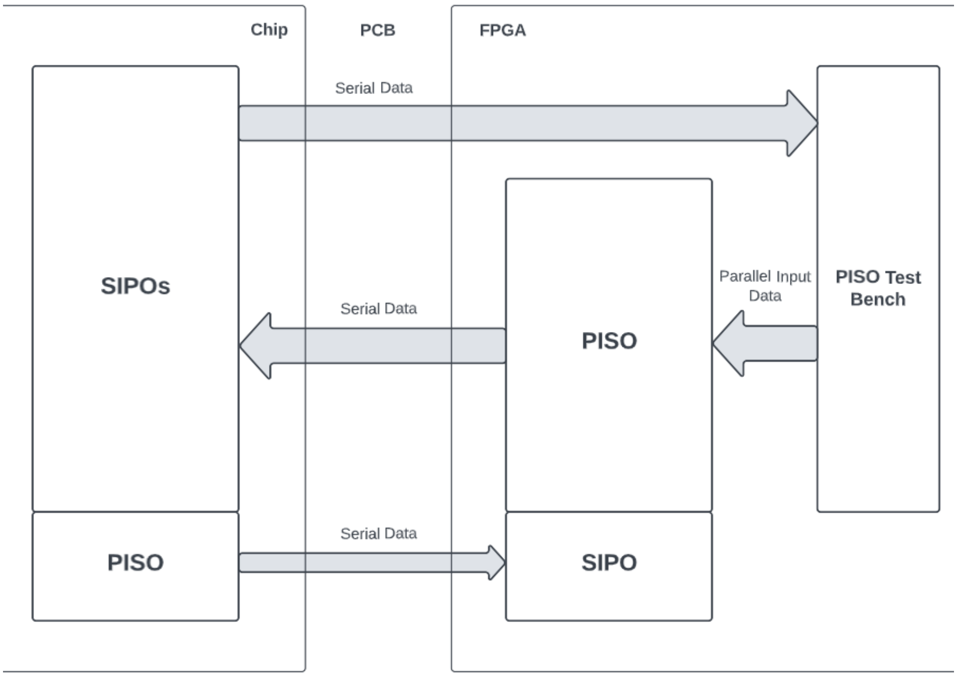


Figure 8. Block diagram of the complete system

3.1. Testing of the SIPO registers of the chip

The chip contains 15 SIPO registers that could be divided into 5 groups:

- BG SIPOs: 3 SIPO registers connected in series, where the output of one SIPO is the input of the follow one.
- DFT SIPO: It is a single SIPO register.
- FE SIPOs: 6 SIPO registers connected in series, where the output of one SIPO is the input of the follow one.
- FECK SIPOs: It is a single SIPO register.
- ADC SIPOs: 4 SIPO registers not connected among them but sharing some communication signals. Practically, they can be considered as single registers.

It is important to know how many SIPOs are contained in every group and how they are connected because the design will depend on these features as well as on the internal design of the single SIPO block that are used. A custom design for every group will be designed. The idea is to create complementary PISO registers to verify that the parallel input of the PISO matches with the SIPO parallel output of the chip. The PISOs will be specifically created for every SIPO group according to its structure. The following scheme shows the idea:

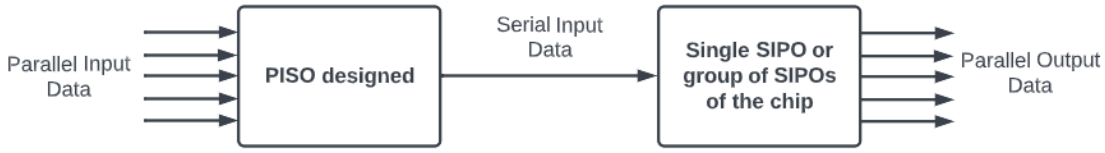


Figure 9. Block diagram of the circuit used to test the SIPOs of the chip

Below is presented the structure of a single SIPO register. All the SIPOs belonging to any group have the same structure. For this reason it is necessary to know its internal structure:

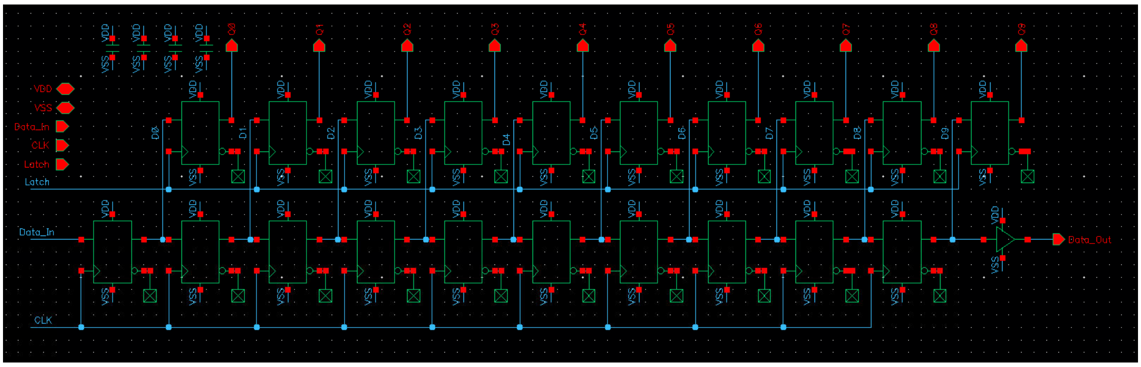
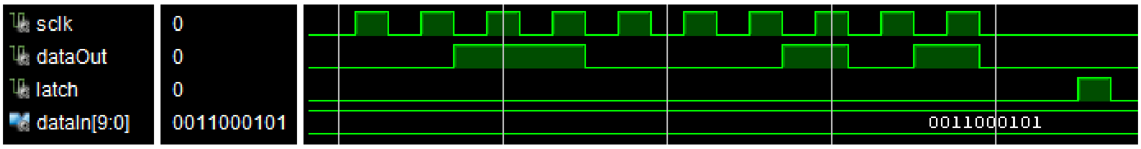


Figure 10. Internal circuit of a single SIPO register of the chip

It is a 10-bits register made up of 20 flip-flops sensitive to positive clock edges. By generating 10 clock pulses, the serial input data is shifted through the 10 flip-flops chain. Once these clock edges have been produced, every bit of the data is in the corresponding flip flop of the chain and ready to be serialised. However, the data is still not in the output. To finally do so, it is needed to generate a pulse in the clock input of the 10 flip-flops that do not belong to the indicated chain of 10 flip-flops (those at the top of the image). As can be seen in the figure, their clock inputs are connected to a signal called latch, then the pulse should be generated by this latch signal, which should be generated by the PISO to be designed.

3.1.1. Features of a single PISO designed to control a single SIPO

In the following plot is shown how a single PISO block should act in order to serialise its input data to be propagated in the SIPO and what signals must be generated for this purpose.



sclk 0
dataOut 0
latch 0
dataIn[9:0] 0011000101

Figure 11. 10-bit ("0011000101") serialisation of a single PISO

It can be seen as, besides the serial input data (Data_In in the Figure 10 and dataOut in the Figure 11) and the latch signal (latch), the PISO to be designed should be in charge of generating the clock signal (sclk) used in the communication. The three signals are output signals from the PISO, and consequently, inputs from the SIPO.

Note that as the flip-flops of the SIPO register are sensitive to positive clock edges, it is mandatory to put the data in the PISO serial output in every negative clock edge. In this way, it will be stable at the SIPO input when a positive clock edge is produced. It is also important to note that since there is no previous negative edge prior to the first positive clock edge, the design was done to put the first transferred bit in the negative edge of an "invisible" clock pulse before the first real clock pulse that can be seen in the image. So that, in the first positive clock edge the bit is already set (“0” in the case of the image), and the SIPO register can capture the first bit.

As stated before, a latch pulse is generated after the 10 bits have been serialised and transferred to the SIPO. As a design requirement, this signal must have the same width as a clock pulse.

Finally, for clarification, the dataOut signal of the plot (PISO serial output) would be connected to de Data_In signal of the SIPO circuit (SIPO serial input).

3.1.2. PISO block to test the BG SIPOs

In this section will be presented how the PISO block should be designed in order to control and test the BG SIPOs.

These are the BG SIPOs:

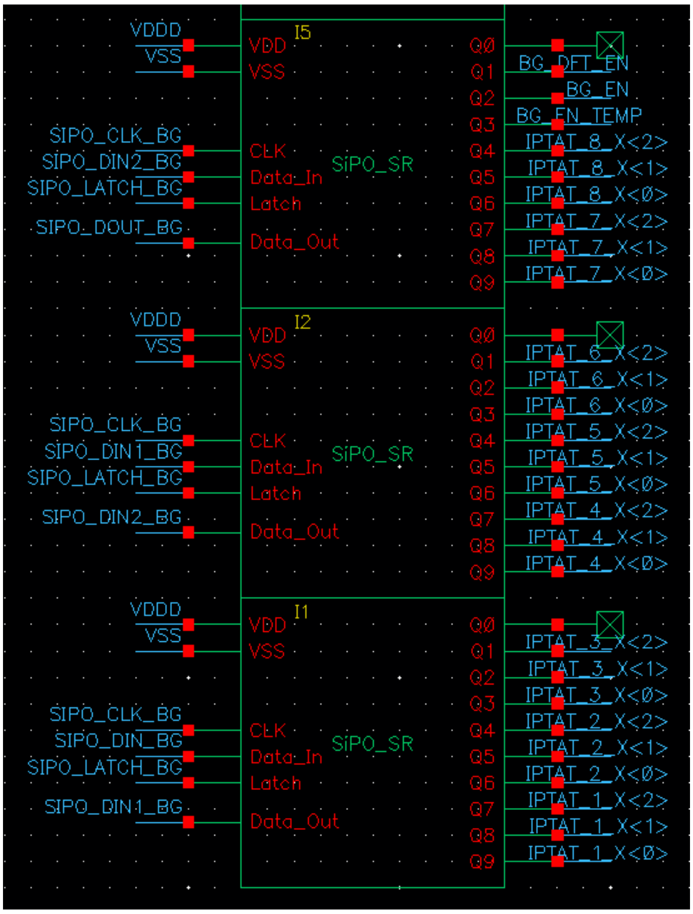
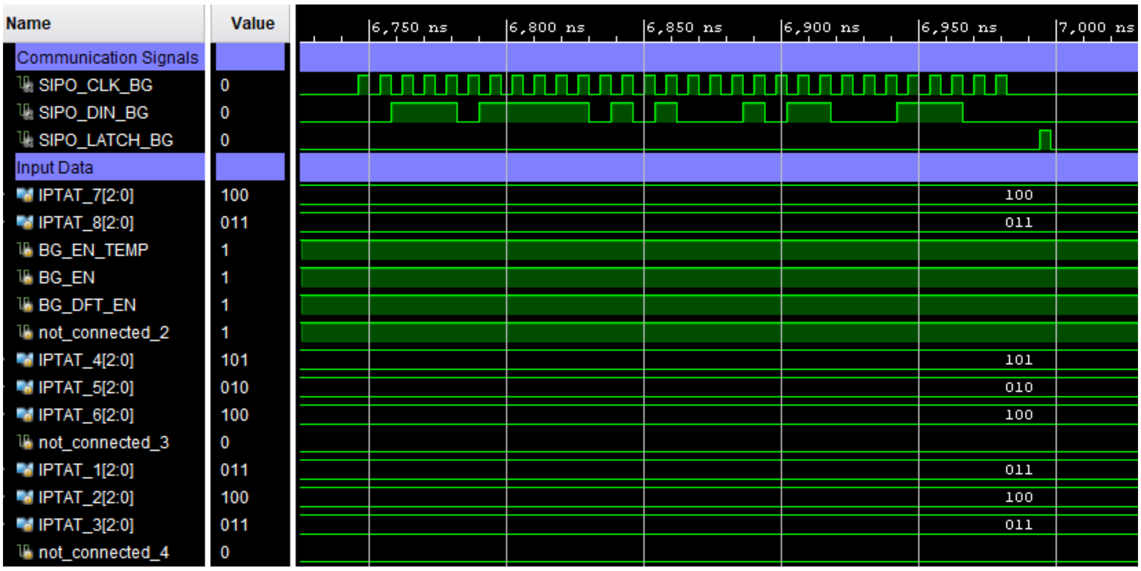


Figure 12. BG SIPOs of the chip

In the figure above can be seen as there are 3 SIPO registers connected in series. Every SIPO deserialises the received serial data, but it also has a serial data output connected to the next serial input of the SIPO. This serial data output simply outputs the input data to the next register. Furthermore, to design the PISO block, it is also important to take into account that the 3 registers share the same clock and latch signals. The internal structure of the SIPOs is exactly the same seen in the Figure 10.

From all the information stated above, it was designed a PISO block capable of serialising 30 bits which should transfer to the 3 SIPO registers of the chip. As the 3 SIPOs are connected serially, the first 10 bits sent by the PISO will be deserialised by the SIPO I5, the second 10 bits by I2 and, the last 10 bits by I1.

The designed PISO for the BG SIPOs behaves in this way:



Name	Value
Communication Signals	
SIPO_CLK_BG	0
SIPO_DIN_BG	0
SIPO_LATCH_BG	0
Input Data	
IPTAT_7[2:0]	100
IPTAT_8[2:0]	011
BG_EN_TEMP	1
BG_EN	1
BG_DFT_EN	1
not_connected_2	1
IPTAT_4[2:0]	101
IPTAT_5[2:0]	010
IPTAT_6[2:0]	100
not_connected_3	0
IPTAT_1[2:0]	011
IPTAT_2[2:0]	100
IPTAT_3[2:0]	011
not_connected_4	0

Figure 13. Simulation of the designed PISO to test the BG SIPOs

In this simulation is seen as the PISO generates an output clock signal of 30 pulses since there are 30 bits to transfer to the BG SIPOs. This clock has a frequency of 125MHz. Every new bit is put in the PISO output (SIPO_DIN_BG) in the negative clock edge, so that, the SIPOs will capture them in the positive clock edge when the data is already stable.

After the 30 clock pulses have already been produced and before the latch pulse is generated, all the 30 bits are distributed in the corresponding flip-Flop of the 3 SIPOs. Then, the SIPO_LATCH_BG pulse can be generated by the PISO in order to deserialise the data in the SIPOs output.

3.1.3. PISO block to test the DFT SIPO

In this case, there is only a single DFT SIPO, thus, the PISO is designed to control a single SIPO as in the explanation of the Section 3.1.1.

This is the DFT SIPO to be tested:

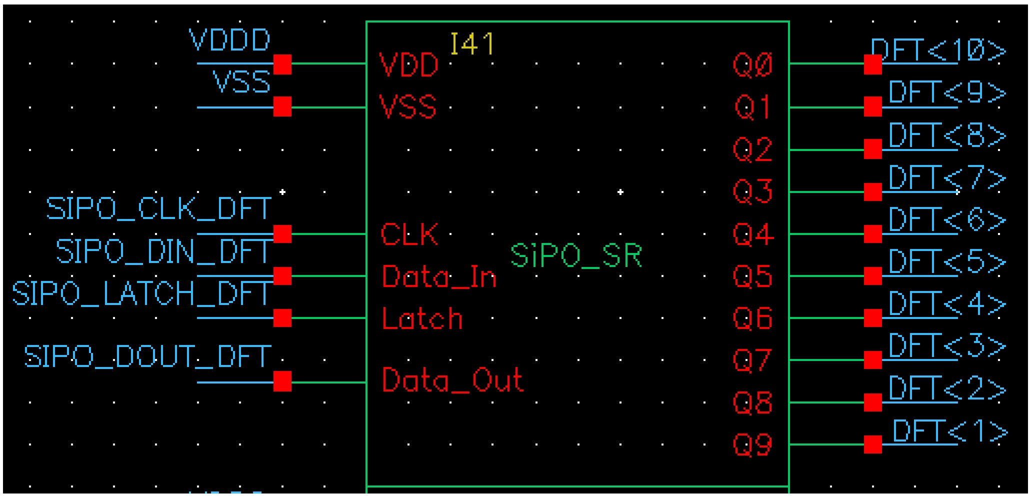


Figure 14. DFT SIPO of the chip

Now, it is presented the behaviour of the designed PISO by means of a simulation:

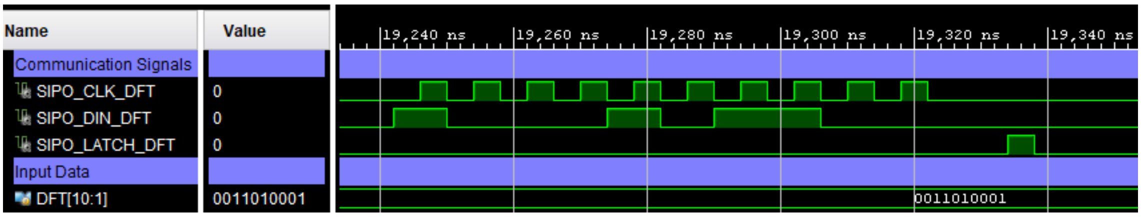


Figure 15. Simulation of the designed PISO to test the DFT SIPO

Since a single SIPO only has 10 flip-flops (not counting the ones controlled by the latch signal in order to output the data), it is only needed to generate a clock signal (SIPO_CLK_DFT) of 10 pulses. This clock signal has a frequency of 125MHz.

On the other hand, it can be seen as the parallel input data of the DFT[10:1] bus is serialised in every negative clock edge, as expected. Note that the most significant bit in the DFT SIPO corresponds to DFT[1]. Therefore, the DFT[1] data is firstly serialised, as it must go through all the flip-flops chain of the SIPO until it reaches the tenth flip-flop, which corresponds to the most significant bit of the SIPO (Q9 in the Figure 14). When DFT[1] reaches this last flip-flop, once the tenth clock pulse is produced, the rest of the bits are in their correct flip-flop too. Thus, the 10 bits are ready to be outputted in parallel and the latch pulse is generated.

3.1.4. PISO block to test the FE SIPOs

In this case, a PISO block is designed for 6 SIPO registers connected in series. Here is presented the structure of the 6 FE SIPOs:

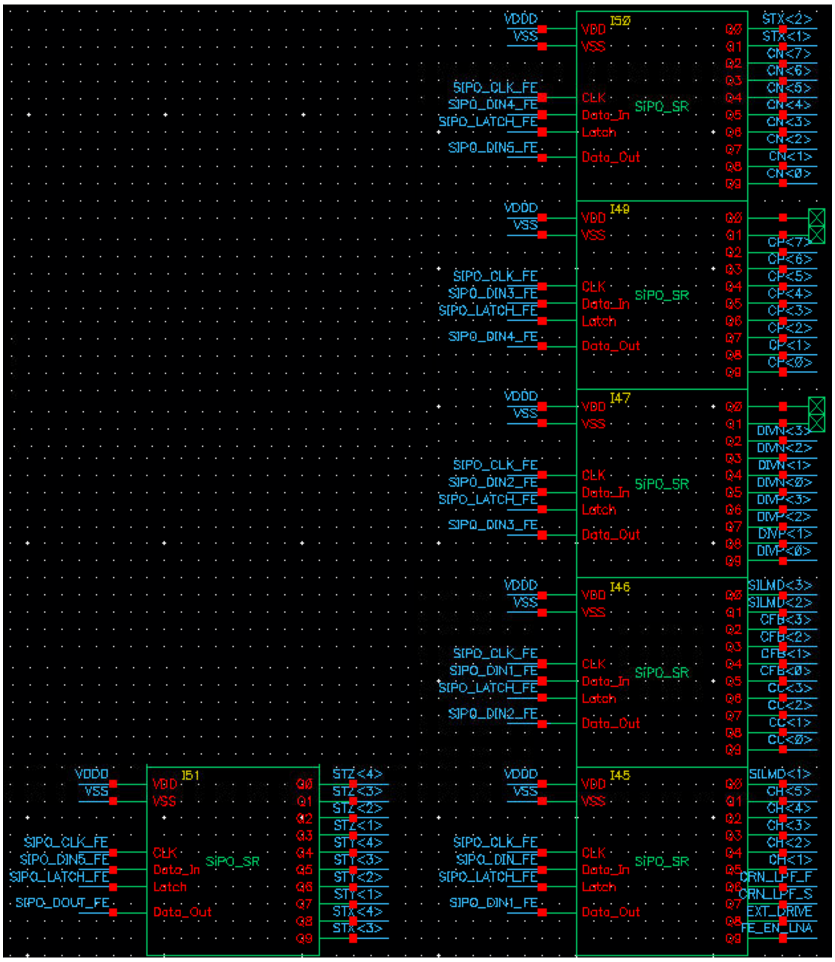
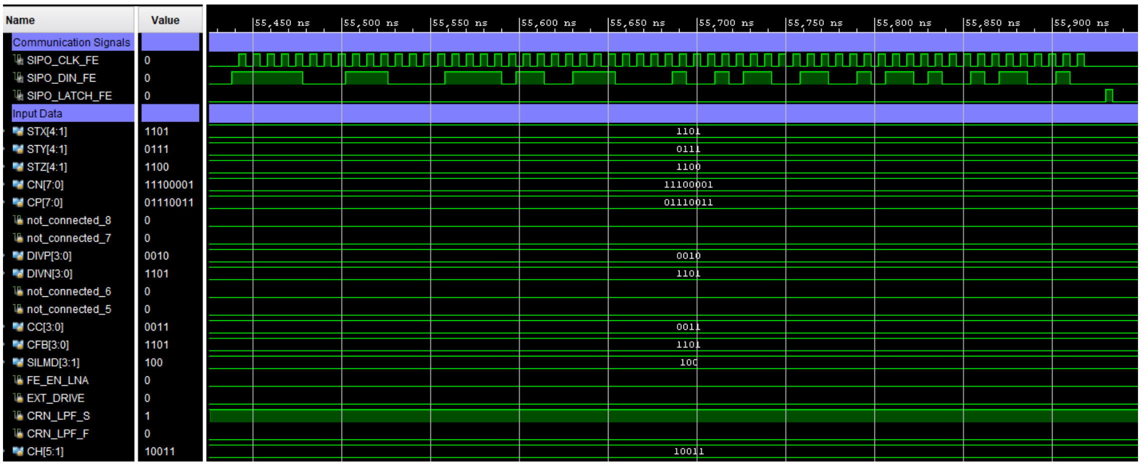


Figure 16. FE SIPOs of the chip

The connection is the same as the BG SIPOs seen in the Section 3.1.2, but in this case there are 3 extra shift registers. Which means that the designed PISO block, instead of generating a clock of 30 pulses, it generates a 60 pulses clock since there are 60 bits to be transmitted to the SIPOs.

Then it was created a PISO block capable of serialising 60 parallel input bits to be transferred to the first SIPO register, I45. The output of the PISO block is connected to the Data_in input of this register I45 in order to propagate all the bits through the rest of the SIPOs in every positive clock edge generated by the PISO block.

See the behaviour of the PISO block that has been created:



Name	Value
Communication Signals	
SIPO_CLK_FE	0
SIPO_DIN_FE	0
SIPO_LATCH_FE	0
Input Data	
STX[4:1]	1101
STY[4:1]	0111
STZ[4:1]	1100
CN[7:0]	11100001
CP[7:0]	01110011
not_connected_8	0
not_connected_7	0
DIVP[3:0]	0010
DIVN[3:0]	1101
not_connected_6	0
not_connected_5	0
CC[3:0]	0011
CFB[3:0]	1101
SILMD[3:1]	100
FE_EN_LNA	0
EXT_DRIVE	0
CRN_LPF_S	1
CRN_LPF_F	0
CH[5:1]	10011

Figure 17. Simulation of the designed PISO to test the FE SIPOs

It can be appreciated as the designed PISO generates the 60 clock pulses necessary to serially transfer the 60 input bits to the SIPOs, in addition to the latch signal. The clock signal of this simulation has a frequency of 125MHz as in the previous subsections and in the next subsections of the Section 3.

3.1.5. PISO block to test the FECK SIPO

The PISO that has been designed to test the FECK SIPO should have the same structure as the PISO designed to test the DFT SIPO as both are a single register that deserialises 10 bits. Thus, the PISO only has to generate 10 clock pulses.

Below is shown the FECK SIPO to be tested:

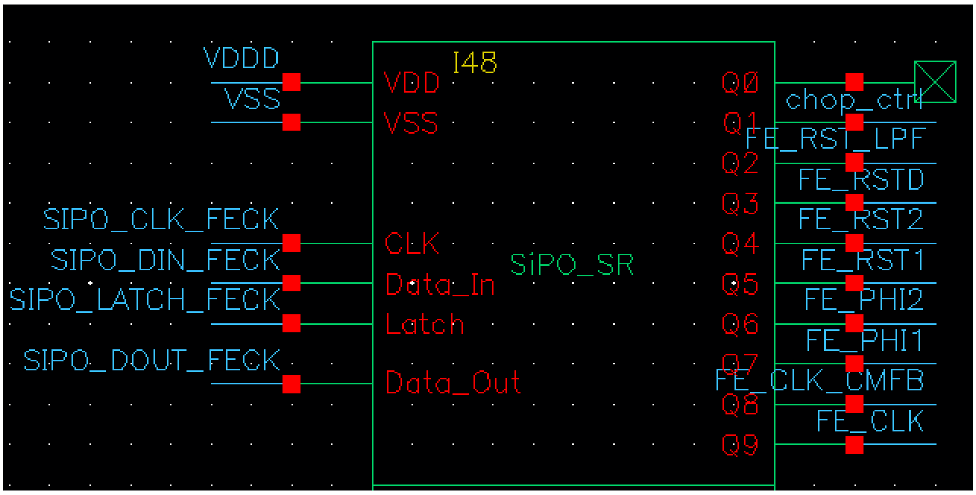
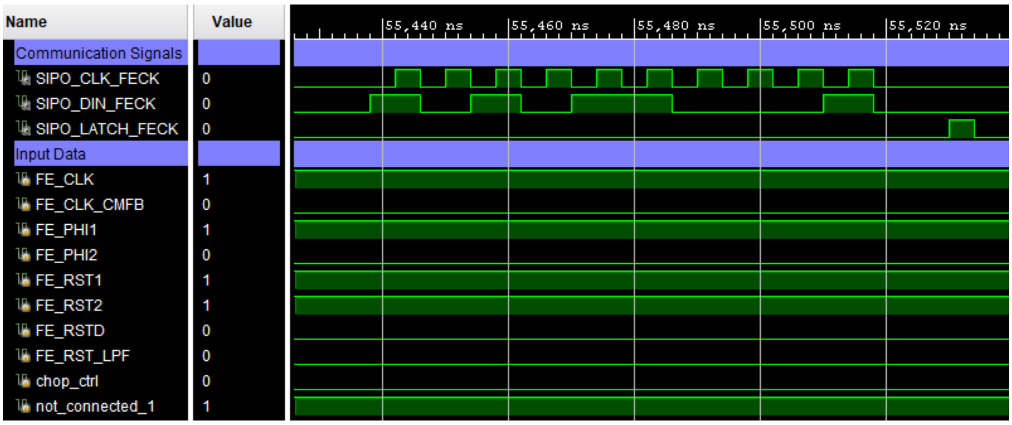


Figure 18. FECK SIPO of the chip

Now, the behaviour of the designed PISO:



Name	Value
Communication Signals	
SIPO_CLK_FECK	0
SIPO_DIN_FECK	0
SIPO_LATCH_FECK	0
Input Data	
FE_CLK	1
FE_CLK_CMFB	0
FE_PHI1	1
FE_PHI2	0
FE_RST1	1
FE_RST2	1
FE_RSTD	0
FE_RST_LPF	0
chop_ctrl	0
not_connected_1	1

Figure 19. Simulation of the designed PISO to test the FECK SIPO

3.1.6. PISO block to test the ADC SIPOs

To test the 4 ADC SIPOs, the PISO block cannot be designed by considering that the 4 registers are connected serially as seen in previous cases, this is a different structure. Actually, they can be considered as 4 single SIPOs because they are not connected in series. The input to each register comes from off-chip individually and the outputs are also sent off-chip individually. However, they all share the same clock signal and the same latch signal.

These are the ADC SIPOs to be tested:

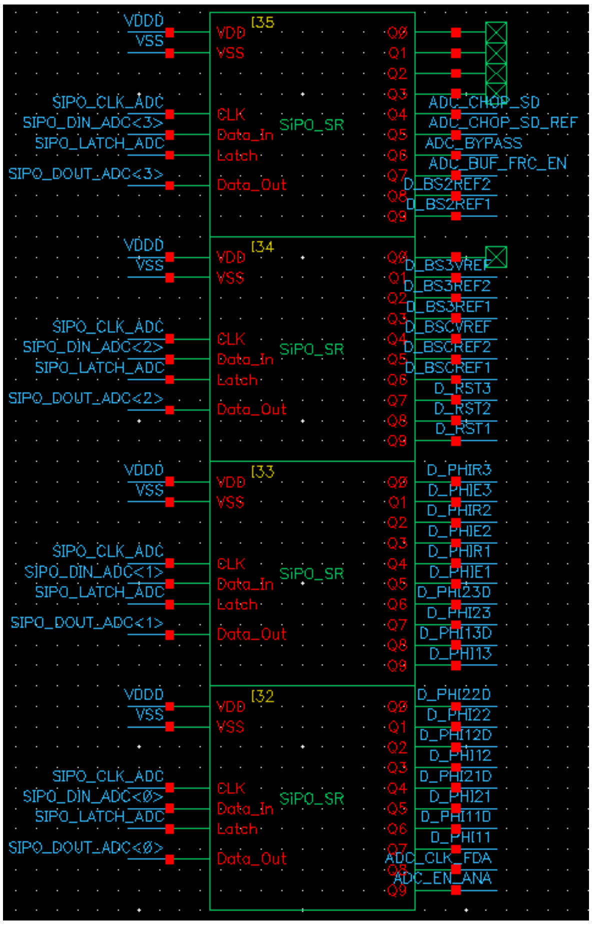
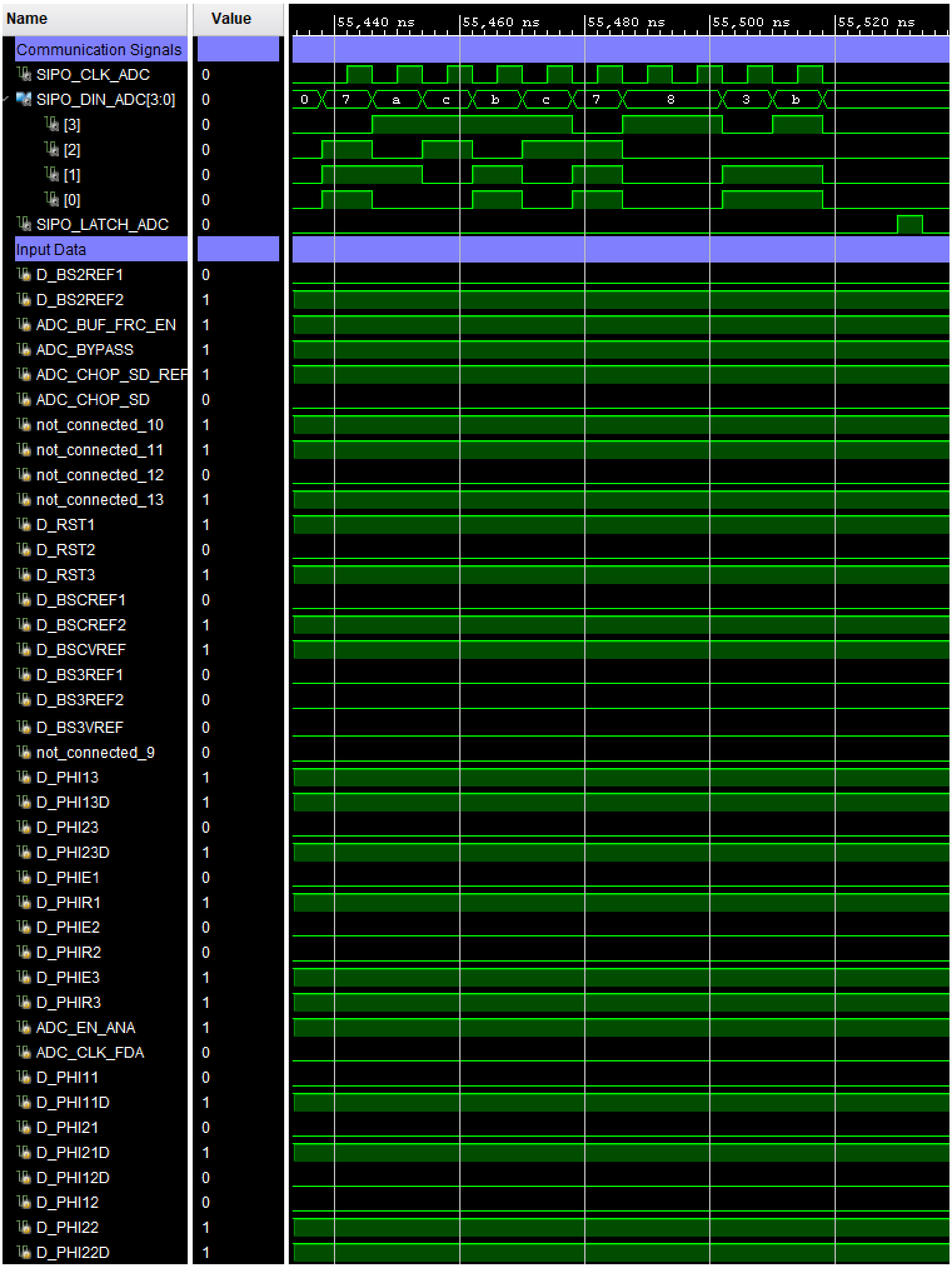


Figure 20. ADC SIPOs of the chip

Now, it is presented the behaviour of the designed PISO block to control the 4 ADC SIPOs:



Name	Value
Communication Signals	
SIPO_CLK_ADC	0
SIPO_DIN_ADC[3:0]	0
[3]	0
[2]	0
[1]	0
[0]	0
SIPO_LATCH_ADC	0
Input Data	
D_BS2REF1	0
D_BS2REF2	1
ADC_BUF_FRC_EN	1
ADC_BYPASS	1
ADC_CHOP_SD_REF	1
ADC_CHOP_SD	0
not_connected_10	1
not_connected_11	1
not_connected_12	0
not_connected_13	1
D_RST1	1
D_RST2	0
D_RST3	1
D_BSCREF1	0
D_BSCREF2	1
D_BSCVREF	1
D_BS3REF1	0
D_BS3REF2	0
D_BS3VREF	0
not_connected_9	0
D_PHI13	1
D_PHI13D	1
D_PHI23	0
D_PHI23D	1
D_PHIE1	0
D_PHIR1	1
D_PHIE2	0
D_PHIR2	0
D_PHIE3	1
D_PHIR3	1
ADC_EN_ANA	1
ADC_CLK_FDA	0
D_PHI11	0
D_PHI11D	1
D_PHI21	0
D_PHI21D	1
D_PHI12D	0
D_PHI12	0
D_PHI22	1
D_PHI22D	1

Figure 21. Simulation of the designed PISO to test the ADC SIPOs

As the 4 SIPO registers are driven by the same clock signals, it is not needed to generate 40 clock pulses but 10. These clock pulses are used by every SIPO to deserialise the received serial data simultaneously. Therefore, in this case, the PISO block should have 4 output data signals and every one of them should be connected to one SIPO input. This can be seen in the former simulation. These 4 PISO output data signals are: SIPO_DIN_ADC[3], SIPO_DIN_ADC[2], SIPO_DIN_ADC[1] and SIPO_DIN_ADC[0].

Once, the 10 bits have been transferred to every register, a common latch pulse for the 4 registers is generated in order to deserialise the data in every SIPO.

3.1.7. Additional PISO requirements

All the sub blocks explained in former sections were integrated in a single PISO top design. This top block has some extra characteristics that are important to highlight: adjustable clock frequency for data transmissions, input change detection and manual transmission.

In addition to these 3 features, it has a reset pin to stop any transmission that is being produced and to reset all the internal variables of the block.

3.1.7.1. Adjustable clock frequency

The clock frequency at which the transmissions are performed in all the PISOs has been parameterised and can be varied by only changing a single top block parameter. Thus, all the PISOs generate the clock signal with the same frequency. This is a divider parameter and the system clock is divided by this factor to get the frequency of the clock used in the communications.

3.1.7.2. Input change detection

This feature is present in all the sub PISOs contained in the top block. When the input data is changed in a PISO register during a data transmission, the transmission must be aborted and restarted. It is only needed that a single bit changes to abort and restart the transmission. For sure, the latch signal is not generated until the complete data has been transferred.

This feature is shown in the simulation below:

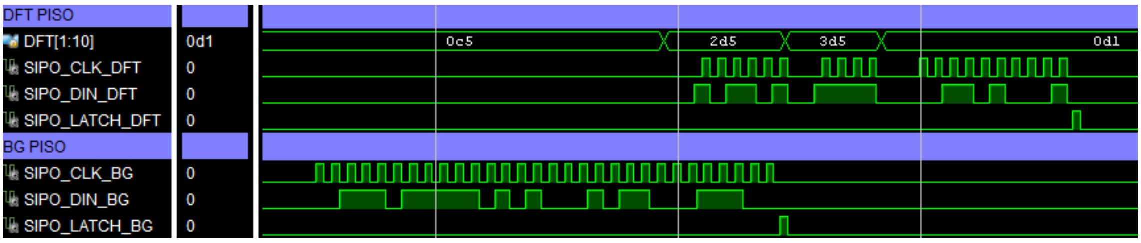


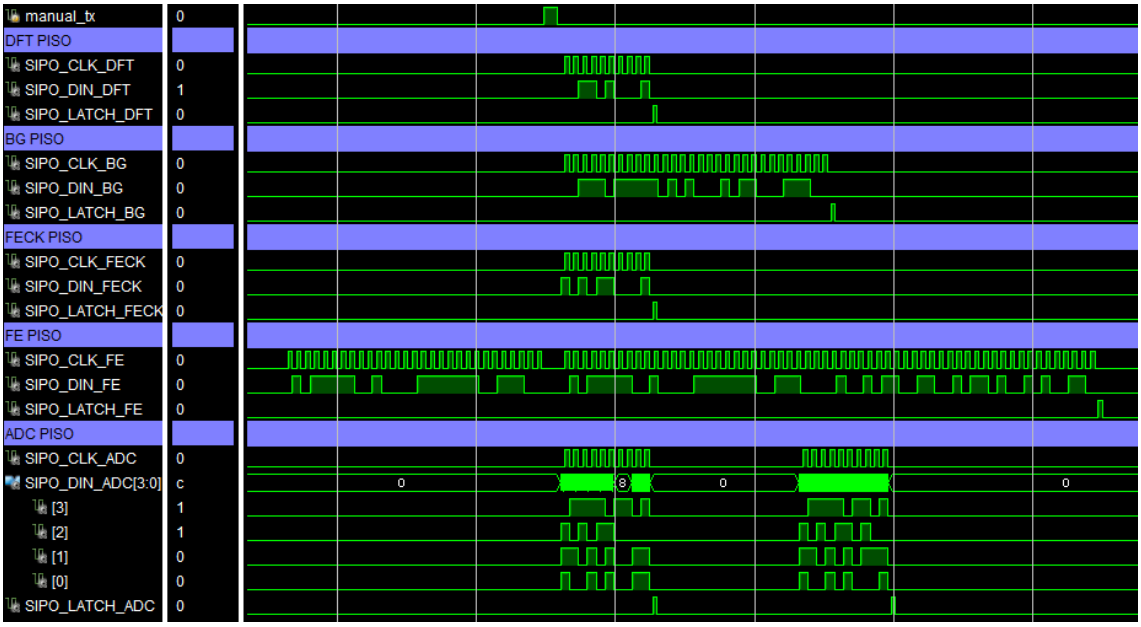
Figure 22. Input change detection simulation

Every time the DFT input data changes, even if it is just a bit, the transmission is aborted and restarted. Only if the whole data has been serialised, the latch pulse is generated. Note that the DFT input data is changed during a simultaneous transmission in the DFT and BG PISOs. However, the only transfer that is aborted is the one of the DFT. The BG PISO transfer is not aborted as its input did not change. Therefore, the input change detection is independent for each PISO.

3.1.7.3. Manual transmission

The top design has an input to transmit manually the input data of all the PISOs at the same time, regardless of whether it changed or not. This is only a single data transfer. This input signal should be in low state by default and when a single transmission is desired, it is only needed to generate a pulse in this input. No matter how long the pulse is, the transmission is done when a positive signal edge is produced. In case, during a transmission, a pulse in the manual transmission input is generated, the transmission will be aborted and restarted as if the input data of that PISO would have changed.

The behaviour is shown in the next simulation:



manual_tx	0
DFT PISO	
SIPO_CLK_DFT	0
SIPO_DIN_DFT	1
SIPO_LATCH_DFT	0
BG PISO	
SIPO_CLK_BG	0
SIPO_DIN_BG	0
SIPO_LATCH_BG	0
FECK PISO	
SIPO_CLK_FECK	0
SIPO_DIN_FECK	0
SIPO_LATCH_FECK	0
FE PISO	
SIPO_CLK_FE	0
SIPO_DIN_FE	0
SIPO_LATCH_FE	0
ADC PISO	
SIPO_CLK_ADC	0
SIPO_DIN_ADC[3:0]	c
[3]	1
[2]	1
[1]	0
[0]	0
SIPO_LATCH_ADC	0

Figure 23. Manual transmission simulation

3.2. Testing of the PISO register of the chip

In this section, it is explained how it was tested the only PISO register contained in the chip and how its internal structure is. For the former, it was necessary to design a complementary block: a SIPO shift register. The main idea is the same as in previous sections to test the SIPOs: to serially connect the PISO and the designed SIPO to verify that the parallel input data matches with the parallel output data. However, in this case, this input data should be inserted directly in the device under test contained in the chip (PISO), and not in the designed block to then propagate it to the design under test as was seen previously.

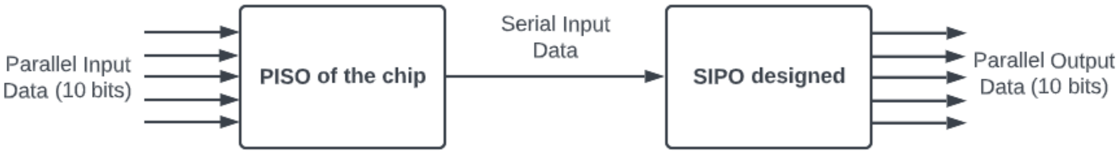


Figure 24. Block diagram of the circuit used to test the PISO of the chip

Below is shown how the PISO block is:

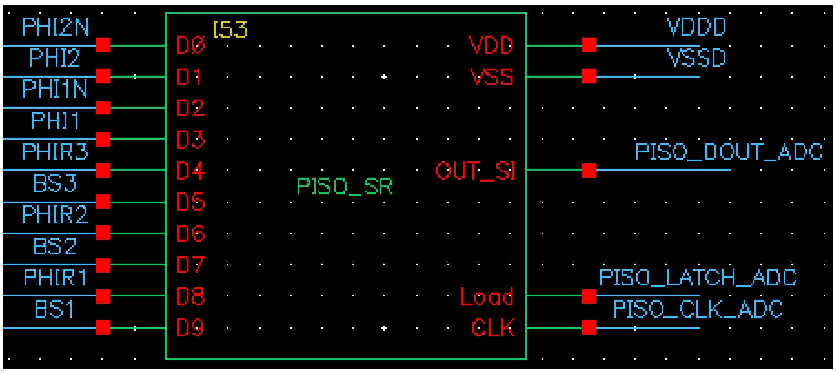


Figure 25. PISO block of the chip

Now, it is analysed the internal structure of the PISO to be tested in order to design the SIPO register accordingly:

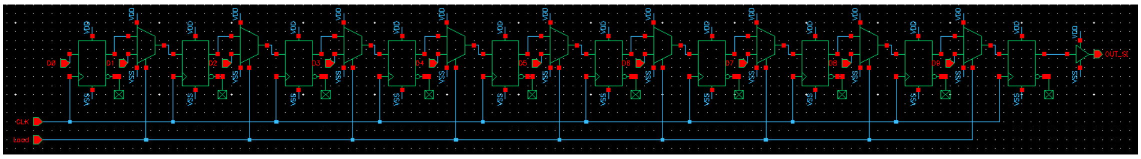


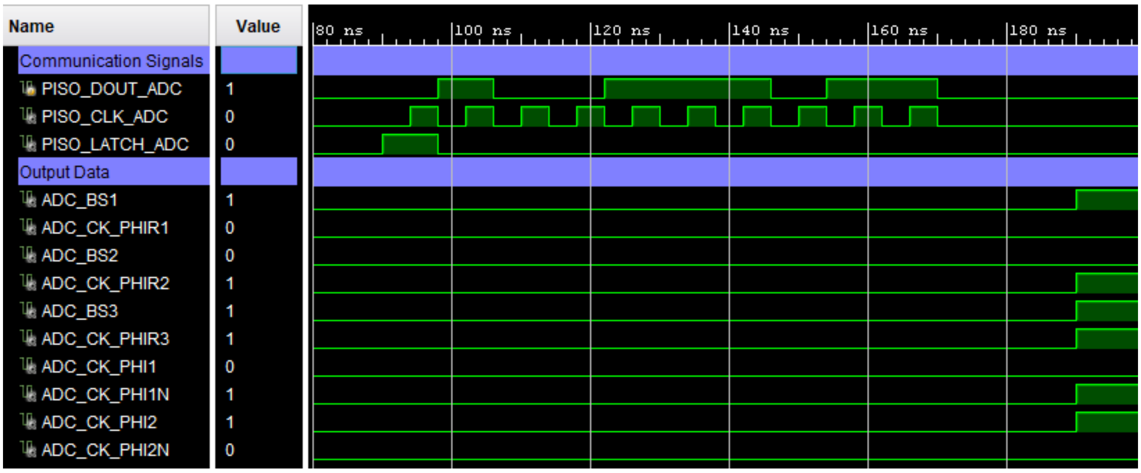
Figure 26. Internal circuit of the PISO of the chip

Since it is a 10-bit shift register, it has 10 flip-flops. Unlike the SIPOs, there are not 10 extra flip-flops but 9 multiplexers among them. These multiplexers are used to capture the data from the input and transfer every bit to the next flip-flop.

The load/latch signal controls the multiplexers in order to load the parallel data placed in the PISO input into the flip-flops. For this reason, it is fundamental that only during the first positive clock edge, the load signal is in high state. This signal must not be in high state until the next data transfer. In order to ensure that the load signal is already in high state when the first positive clock edge is produced, the pulse is generated with a width of a clock period. The first positive clock edge always coincides with the middle of the load pulse. Even if the clock frequency is changed, thus varying the period, the pulse width of the load signal will also change and adapt to the duration of the new clock period.

3.2.1. Design of the SIPO register

In this section it is shown how the designed SIPO should act in order to read the PISO output data:



Name	Value
Communication Signals	
PISO_DOUT_ADC	1
PISO_CLK_ADC	0
PISO_LATCH_ADC	0
Output Data	
ADC_BS1	1
ADC_CK_PHIR1	0
ADC_BS2	0
ADC_CK_PHIR2	1
ADC_BS3	1
ADC_CK_PHIR3	1
ADC_CK_PHI1	0
ADC_CK_PHI1N	1
ADC_CK_PHI2	1
ADC_CK_PHI2N	0

Figure 27. Deserialisation of the sequence “1001110110”

In the simulation above can be seen how the SIPO generates the serial clock at 125MHz and the load signal (PISO_LATCH_ADC) so that the PISO can transmit serially its parallel input data to the SIPO. Despite the simulation is showing like the PISO of the chip puts every bit in the negative clock edge, it is not actually, the flip-flops of the PISO are sensitive to the positive clock edge. This only was designed in this way in the test bench, in order to verify that the designed SIPO captured the bits in the positive clock edges.

An important characteristic of the design is that, by default, the first bit (MSB) is captured in the second positive clock edge. The last bit to be deserialised (LSB), is captured after the tenth clock pulse, in an eleventh “invisible” clock pulse.

The data capture must start in the second positive clock edge because when the SIPO was simulated together with the chip using a realistic simulation, it was found a certain delay between the FPGA and the chip. This fact can be seen in the following image:

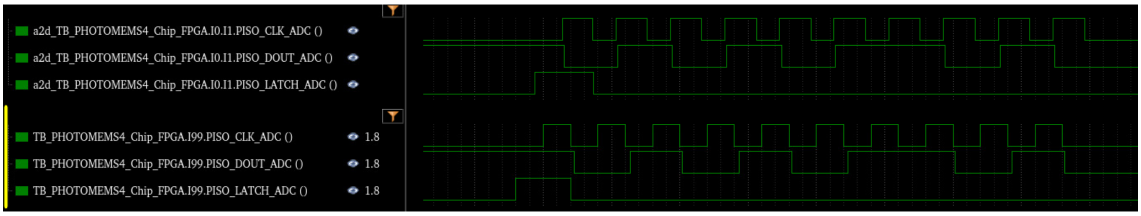

a2d_TB_PHOTOMEMS4_Chip_FPGA.I0.I1.PISO_CLK_ADC ()
a2d_TB_PHOTOMEMS4_Chip_FPGA.I0.I1.PISO_DOUT_ADC ()
a2d_TB_PHOTOMEMS4_Chip_FPGA.I0.I1.PISO_LATCH_ADC ()
TB_PHOTOMEMS4_Chip_FPGA.I99.PISO_CLK_ADC () 1.8
TB_PHOTOMEMS4_Chip_FPGA.I99.PISO_DOUT_ADC () 1.8
TB_PHOTOMEMS4_Chip_FPGA.I99.PISO_LATCH_ADC () 1.8

Figure 28. FPGA-chip delay. Signals of the chip (above) and signals of the FPGA (below)

There is no problem with the clock and latch signals because both are generated by the FPGA and are synchronised between them. However, as both PISO (chip) and SIPO (FPGA) interacts with the PISO_DOUT_ADC signal, the former to put the data and the later to capture it, it is very important to consider this delay.

Initially, it was considered an ideal case: since the PISO transmits the bits in every positive clock edge, the SIPO could be designed to capture the data in the negative clock edges. Nevertheless, in the Figure 28 is seen that there is a considerable delay that does not allow this option. In addition, when other elements, such as the PCB, are added, the delay will even get worse. In the Figure 28 is appreciated as the firs bit transmitted by the PISO is a “0”. However, if the SIPO captures this bit in the negative clock edge, the captured data would be a logic “1”, because the “0” is still not received. If, on the contrary, the data capture starts in the second positive clock edge, all the bits are captured correctly. Then, thanks to this delay, there is no conflict because the PISO transmits the data on a positive edge and the SIPO reads on the next positive edge.

3.2.2. Additional SIPO requirements

As was done with the PISO section, now, it is presented some functionalities that make the design much more complete and versatile. It is important to mention that the SIPO also has a reset pin that stops the data transfer and initialises the SIPO register.

3.2.2.1. Adjustable clock frequency

The frequency of the clock used for the transmissions can be modified by means of a single divider parameter, in the same way as in the PISO top block.

3.2.2.2. Enable signal

This enable signal is an input of the SIPO and it is used to start or finish the reading data from the PISO of the chip. In the case of the PISO, it was seen that a change in the input data starts the transmission. In the same way, the SIPO block needs some condition to start the transmission, this condition is given by the enable signal state.

As long as the enable signal has a value of “1”, the SIPO is in a loop reading the data provided by the PISO. This means that the SIPO generates a load pulse and 10 clock pulses constantly (with a certain delay between every transmission). This functionality allows to have a faithful image of the current output value, or almost current. The delay in updating the data will be equal to the delay between the start of one transmission and the next. This is because if there is a change in the input data of the SIPO contained in the chip during a transmission, this change will not be reflected in the PISO output until the next data reading is completed.

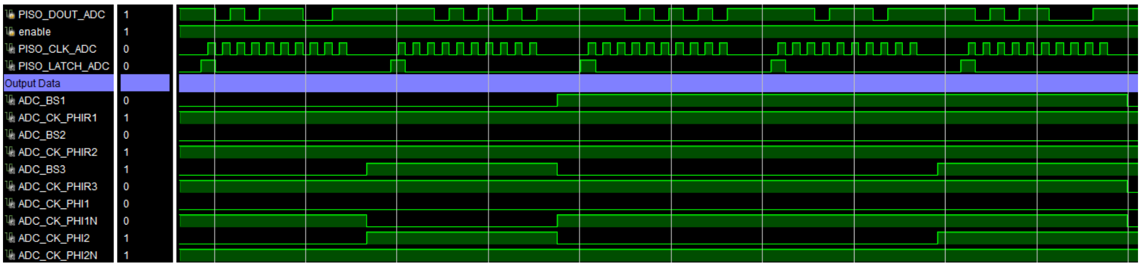


Figure 29. Data transmissions loop from the PISO (chip) to the SIPO (FPGA) when the enable signal is in high state

On the other hand, when the enable signal is equal to "0", the transmission loop is finished and there are no more readings while the signal remains in this state. In case the value of enable signal changes from "1" to "0" during a data transmission, the PISO will keep generating the load and clock signals and, the transmission will be completed. After that, there will be no more data readings.

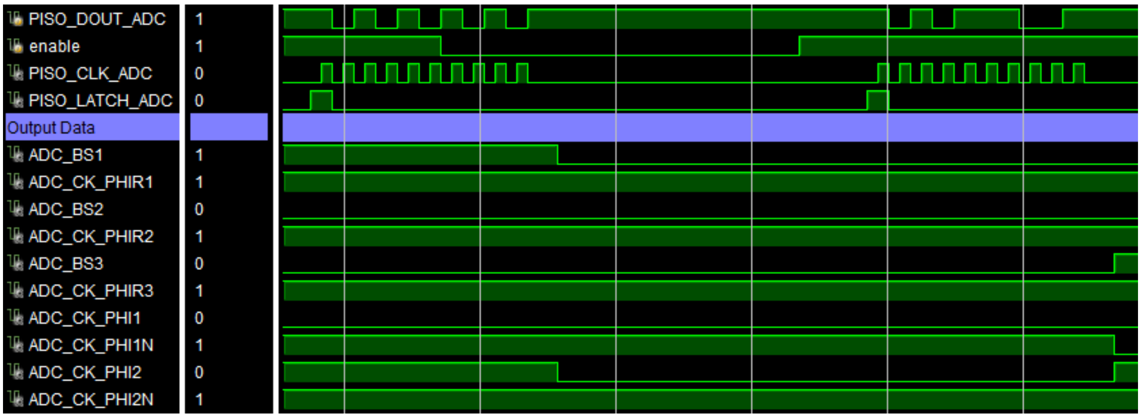


Figure 30. End of the reading loop when the enable signal is "0" during a data transmission and resumption of the loop when the enable signal is"1"

3.2.2.3. Data capture modulation (Delay parameter)

As was mentioned before, there is a significant delay between the chip and the FPGA. In the design to test the SIPOs of the chip, the delay does not affect in the communication as the designed PISOs are just writing data in the SIPOs. Therefore, as all the signals (clock, latch and serial data) are generated in the transmitter, they all are synchronised among them. However, this is not this case, as could be clearly seen in the Figure 28. Now, the clock and the load (latch) signals are generated by the data receptor (the designed SIPO block) and the data is sent by the transmitter (the PISO under test).

In the Figure 28, it was found that starting to read the data in the second positive clock edge was enough to prevent the delay between the chip and the FPGA from affecting the communication. Nevertheless, it was also mentioned that there are more elements that could affect this delay, mainly the PCB which acts as interface, but also the PCB of the FPGA. These delays cannot be estimated by simulation and it is probable that, once these elements are physically added to the system, starting to capture data in the second positive clock edge is not enough, as the delay will increase.

For this reason, the design incorporates the possibility of varying the instant in which the received data is captured by the SIPO. This is done by means of a single parameter of the top design code. By default, the design is made to capture the values on each positive edge of the communications clock, doing the first bit capture in the positive clock edge of the second pulse. This is the default design operation and occurs when the parameter is equal to 0. Each time this design parameter is incremented by one unit, the data capture will be done one system clock later than the value of default capture. It is important to note that it is a delay of one period of the internal clock of the FPGA (system clock), not the clock used for communications, PISO_CLK_ADC.

For example, if it is configured a delay parameter equal to 2, the first data bit will be captured two FPGA clock periods after the positive edge of the second communications clock pulse occurred. The second bit will be captured two FPGA clock periods after the positive edge of the third communications clock pulse occurred. The third bit will be captured two FPGA clock periods after the positive clock edge of the fourth communications clock pulse occurred.

The following simulations show how the data capture can be set by varying the delay parameter value:

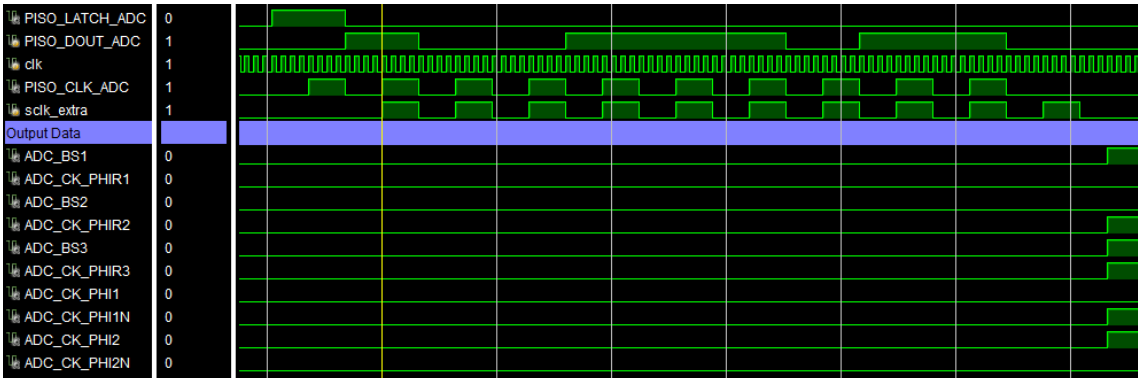


Figure 31. Data capture with Delay = 0 (default value)

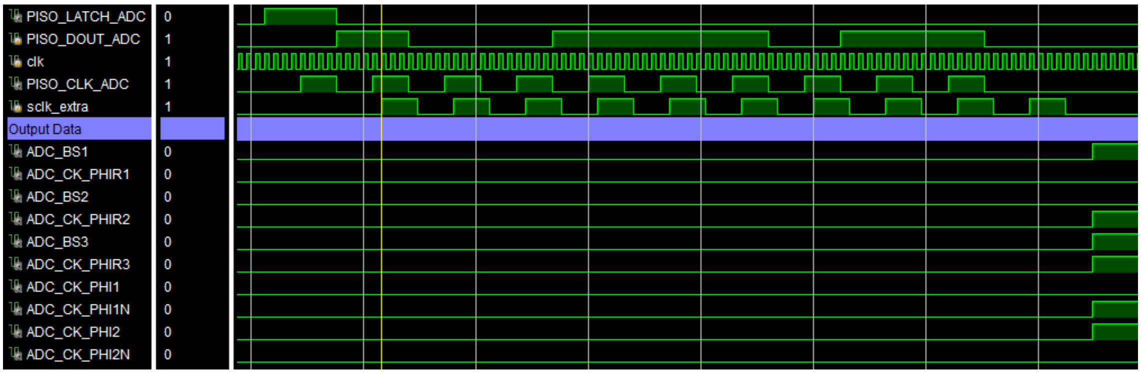


Figure 32. Data capture with Delay = 1

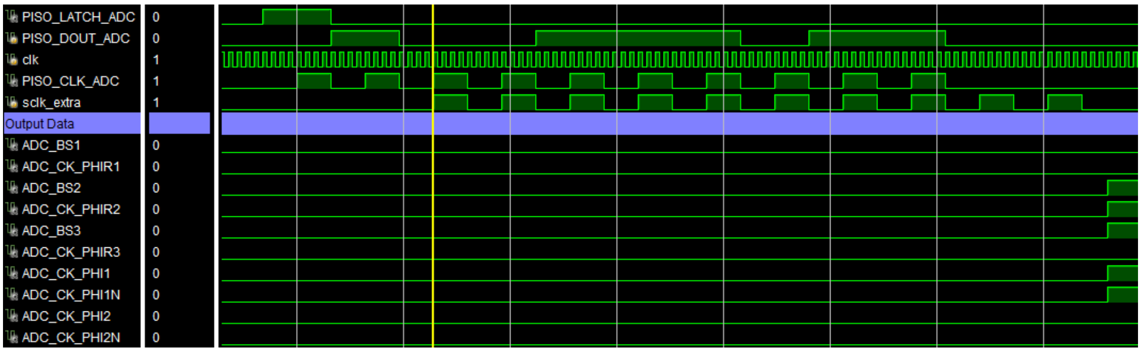


Figure 33. Data capture with Delay = 8

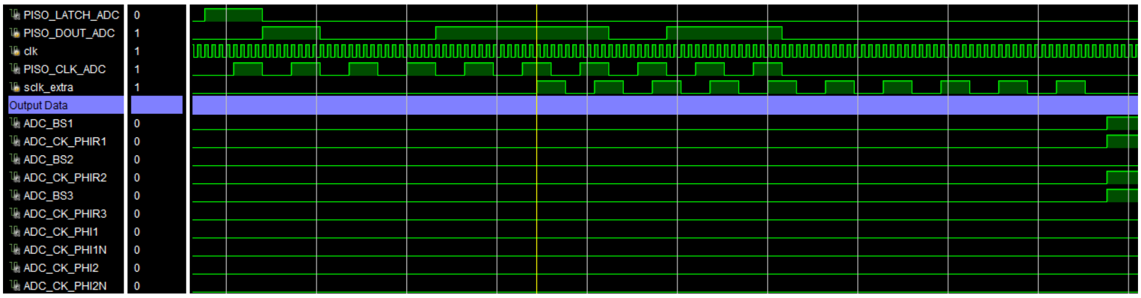


Figure 34. Data capture with Delay = 34

The data capture starts in the positive edge of the internal signal sclk_extra (yellow marked). A new bit is captured in every positive sclk_extra edge. It is important to remark that the sclk_extra signal is an internal signal and only appears in these simulations to show when the data is captured based on the value of the Delay parameter. The clock used for communications is always PISO_CLK_ADC.

As in the former simulations, the PISO_CLK_ADC clock has been set to be 8 times slower than the system clock (clk signal), it can be seen as Delay = 8 implies a delay of one period in the data capture regarding the default capture. Note that the start of data capture can be delayed as many system clock periods as desired. The idea is to modify this parameter once the whole system is physically integrated to set the proper Delay value that provides a correct data lecture.

3.3. Input data generation blocks for simulation

In the Sections 3.1 and 3.2, it was seen how the test blocks have been designed. It has also been seen that in order to check the correct operation of the chip blocks, the idea is to verify that the input data of the PISO-SIPO system coincides with the output data. Whether we are testing the SIPOs or the PISO of the chip, the idea is the same. Until now, this only has been verified by means of a test bench without taking into account the chip. However, in order to perform these tests with the chip by simulation, one more block still needs to be created: a block that provides the input stimuli of the PISO-SIPO system. Two different blocks were created: a block that creates input data for the PISO of the chip and another block that creates input data for the PISOs that test each group of SIPOs.

3.3.1. Input data generator block for the designed PISOs

This block generates the input data provided to the PISOs in order to test the SIPOs contained in the chip. It generates input data to test the 5 groups at the same time, so it has as many outputs as parallel outputs have the SIPOs. This block will have 150 data outputs, since the chip contains 15 SIPOs to test.

On the other hand, this data generator block has 2 operation modes controlled by the state of an input pin. If this pin is in low state, the block sends a single data transfer to each PISO. If, on the other hand, the pin is in high state, the block will carry out 3 consecutive data transfers. The data generated is different in each of the 3 transfers. The behaviour of the block can be clearly understood from the following flow diagram:

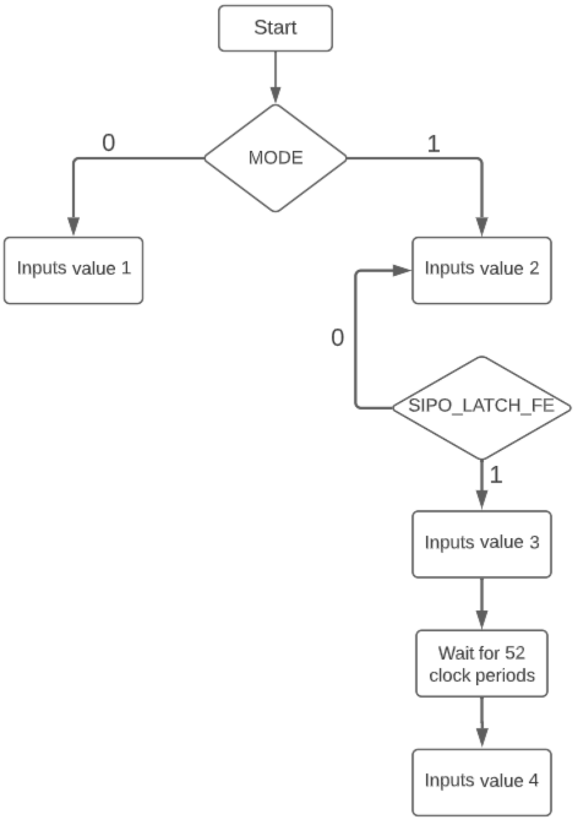


Figure 35. Block diagram of the input data generator block for the designed PISOs

The 2 operation modes are selected with the MODE signal. The SIPO_LATCH_FE generated by the PISO to test the FE SIPOs is used as a flag to change the input values from one value to other (from input value 2 to input value 3).

It is used the signal SIPO_LATCH_FE because when a pulse of this signal is produced, it is sure that all the transfers have been completed since the data transfer to the FE SIPOs is the longer one because there are 60 bits of data. Regarding the 52 clock periods wait, waiting this amount of clock cycles also ensures that all the transfers were completed.

3.3.2. Input data generator block for the PISO contained in the chip

The block generates the input data for the PISO of the chip. Unlike the block of the former section, this one only has to generate 10 bits of input data, as there is only a PISO of 10 input bits. This block provides 3 different output data in a loop. Its operation is shown in the flow diagram below:

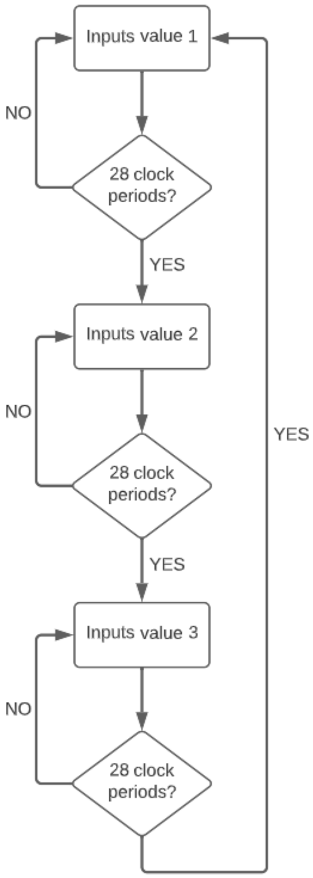


Figure 36. Block diagram of the input data generator block for the PISO of the chip

To change the PISO input data, the generator block waits 28 clock periods in order to ensure that the PISO has serialized the 10 bits that must be transferred to the SIPO. This is because in a whole transfer are produced 10 PISO_CLK_ADC pulses (20 system clock pulses, when working at half the frequency of the system clock). Then, waiting 28 clock cycles is enough to be sure that the transfer was completed.

3.4. Test bench for physical verification of the SIPOs

The input data generator blocks seen previously only send input data to the designed PISOs and SIPO but the output of the blocks of the chip should be verified by simulation. This input data generator blocks are good to verify in simulation that the signals arrive correctly at each block or, in case of the PISO, to stimulate it directly with input data and check that the same data is correctly received. However, it is not valid to know if the shift registers are working as expected with the chip in the real world since it is not possible to access internally to every input of the block. To verify that the registers work correctly, the data out signal of each of them has been used. This output allows to serially obtain all the serial bits that have been entered in the registers. Thus, knowing the values of the signals transmitted to the chip, it can be verified that they coincide with the serial data received from the chip.

The test bench, as well as sending the input data to be verified at the output, should be able to receive, store and show the output data in the LEDs (Light-Emitting Diode) of the FPGA. The strategy is to show the most significant bits or the least significant bits in 5 LEDs, depending on the state of one switch.

Now, it is presented the block diagram of this test bench:

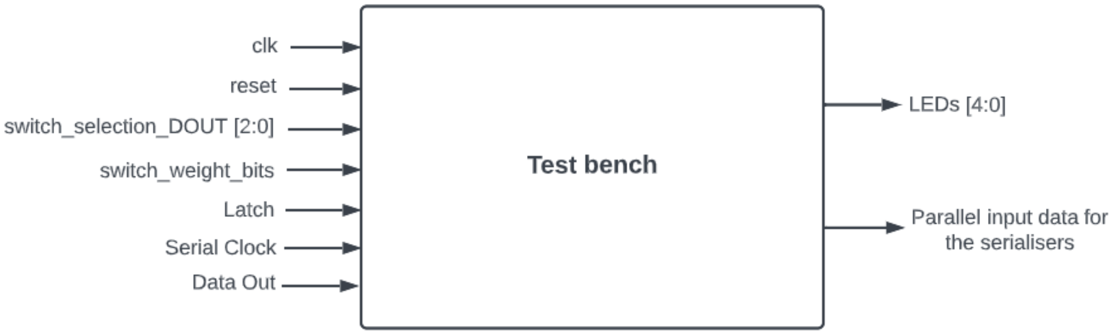


Figure 37. Block diagram of the SIPOs test bench

The data to be displayed in the LEDs is received serially in the Data Out input. To know when every bit should be captured, the test bench has another input, the Serial Clock. This clock is used as a reference to capture the 10 bits, since it is known that the SIPOs send every bit in the positive clock edge of the Serial Clock. As was seen in the section of the designed SIPO (Section 3.2), ideally, every bit could be captured in the negative serial clock edge, however the unknown delay in the chip, the PCB and the FPGA, provokes that a parameter to adjust the data capture is necessary. This is exactly the same approach as in the SIPO created to test the PISO of the chip since in that and in this case, both designs receives data from the chip, and the serial clock is generated in the FPGA. Then, depending on the frequency used for the communications, it will be appropriated to capture the bits in different instants. As in the SIPO designed, incrementing in one unit this parameter, the data capture of every bit is delayed 1 FPGA clock period from the point at which it was done with the previous value of the parameter. If this parameter is equal to 1, it means to capture every bit 1 FPGA clock period after every bit is transmitted by the SIPOs of the chip. The operation of the data capture parameter will be seen deeper in the Section 3.4.1.

The latch signal is used to know when all the bits have been received in Data Out and a new parallel input data can be sent from the test bench to the PISOs. It is needed to assure when a data transfer has been completed because if the input data is changed during a transfer, the current transfer is aborted and a new one starts with the new input data. It was already seen that this is a specification of the PISOs designed (Section 3.1).

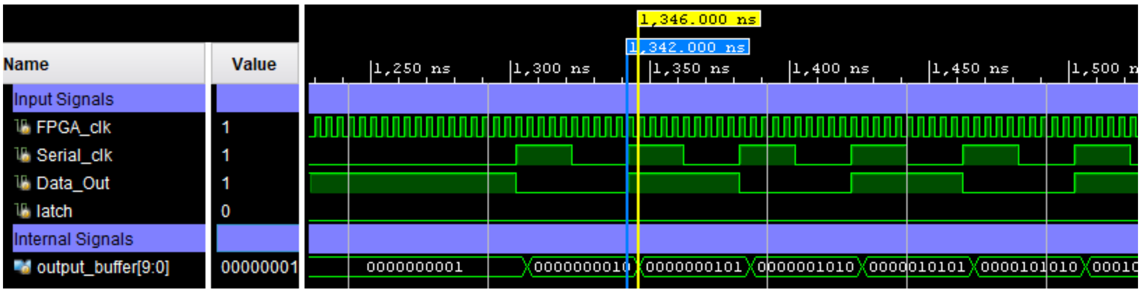
The switch_selection_DOUT signal is used to select which output of all the SIPOs to be tested is desired to show in the LEDs and, the switch_weight_bits signal selects if in the 5 LEDs is desired to display the five most significant bits or the five least significant bits of the received data. Displaying the 10 bits in the FPGA LEDs, it is possible to check visually if the received data matches the one sent by the test bench. Note that in case of the BG group and the FE group, there are several SIPOs connected serially, which means that the whole data received is more than 10 bits. However, as there are not many LEDs available in the FPGA to display the data, the 10 most significant bits are only displayed because this bits go through all the SIPOs of the chain. Therefore, if this 10 bits are correctly received, it means that all the SIPOs operate properly.

Regarding the input data sent to the PISOs, it should be considered that in order to receive in the Data Out input the 10 bits transmitted, it is needed to send 2 different transfers. This is because the 10 bits sent during a clock cycle (10 clock pulses) are received in the next clock cycle. This can be clearly seen by checking the internal structure of the SIPOs of the chip, Figure 10.

It is important to note that to verify the PISO of the chip it is not necessary to create any test bench. This is because the input data for that PISO comes from the chip and cannot be accessed from outside the chip. Therefore, it is only needed to capture and deserialise the data sent by the PISO and, for example, display it on the FPGA LEDs. As seen previously, this reception and deserialisation processes are already carried out by the SIPO designed to verify the PISO, thus, a test bench would be unnecessary.

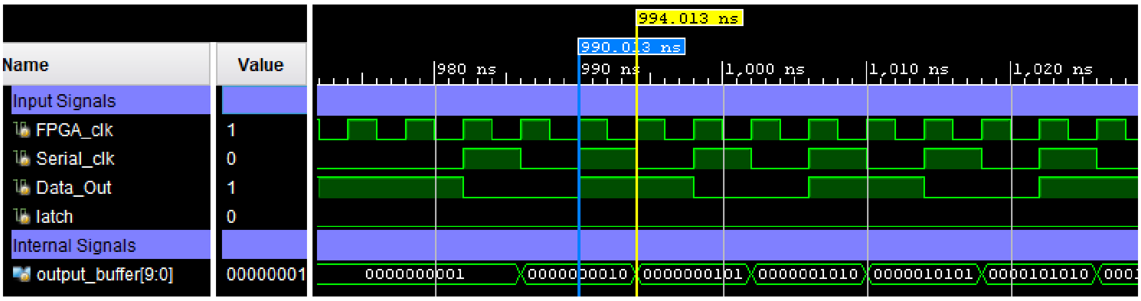
3.4.1. Test bench data capture

The objective of this subsection is to clarify visually the operation of the data capture parameter of the test bench. Its minimum value is one. When the parameter is equal to 1, the data is captured one FPGA clock cycle after the serial data was sent by the transmitter (on each positive serial clock edge), the SIPOs of the chip. This is reflected in the images below:



Name	Value
Input Signals	
FPGA_clk	1
Serial_clk	1
Data_Out	1
latch	0
Internal Signals	
output_buffer[9:0]	00000001

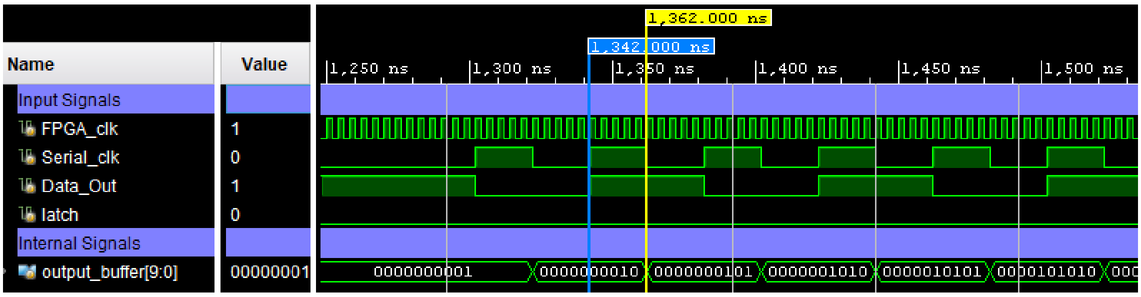
Figure 38. 1 FPGA clock period delay in the data capture (parameter = 1) at 25MHz of serial clock frequency



Name	Value
Input Signals	
FPGA_clk	1
Serial_clk	0
Data_Out	1
latch	0
Internal Signals	
output_buffer[9:0]	00000001

Figure 39. 1 FPGA clock period delay in the data capture (parameter = 1) at 125MHz of serial clock frequency

It is important to note that depending on the frequency operation, the same parameter delay implies capture the data at different points of the serial clock. In the first simulation, as the frequency is very low regarding the FPGA clock, each bit is captured right after the positive serial clock edge has been produced. However, in the second simulation, as the frequency is higher than before, each bit is captured in the negative serial clock edge. Therefore, if it is desired to capture every bit in the negative serial clock edge at 25MHz, the parameter value must be 5, in order to delay 5 FPGA clock periods the data capture from the positive serial clock edge. This is shown in the simulation below:



Name	Value
Input Signals	
FPGA_clk	1
Serial_clk	0
Data_Out	1
latch	0
Internal Signals	
output_buffer[9:0]	00000001

Figure 40. 5 FPGA clock periods delay in the data capture (parameter = 5) at 25MHz of serial clock frequency

3.5. PCB design

As was explained in the project description, the PCB is the physical interface between the chip and the FPGA where the designed blocks are implemented. In this PCB will be placed the chip to be tested. Besides a simple interface, the PCB should also incorporate some circuitry that will be explained in the next sections.

3.5.1. Level shifters

The FPGA operation voltage is 3.3V and the chip operates at 1.8V maximum. For this reason is necessary to incorporate level shifters in the PCB to adapt the signals coming from the FPGA to 1.8V and the signals coming from the chip to 3.3V. To choose the proper component there is a main requirement to be meet: the clock frequency. According to simulations of the chip operation, the ideal system clock frequency is 250 MHz with a divider parameter of 2. This means that the clock used in the communications should have a frequency of 125MHz. This frequency limits the choice of the component.

The clock used in the communications is always generated by the FPGA. Therefore, the component must be able to drop the voltage from 3.3V to 1.8V for a minimum pulse duration of 4ns:

$$T_w = \frac{1}{2 * 125\text{MHz}} = 4\text{ns}$$

The chosen voltage level shifter is the SN74AVC4T774. This component provides a data rate of 200Mbps when dropping voltages down to 1.8V. Which means that the minimum pulse duration is 5ns. This is higher than the 4ns pulse duration for the ideal communication clock frequency of 125MHz. However, since this is the fastest level shifter found and it is also not mandatory to work with this frequency exactly, the frequency can be reduced down to 100MHz (200MHz system clock) in order to get the minimum pulse width of 5ns and be able to use this component. This frequency reduction is not critical at all.

When this level shifter increases the voltage from 1.8V to 3.3V (signals from the chip to the PCB), the component baud rate is 380 Mbps. This is fast enough for the signals transmitted from the chip since the generated clock from the FPGA which determines the communications speed will be much slower.

The schematic below extracted from the datasheet shows how to make the connections for the level shifters:

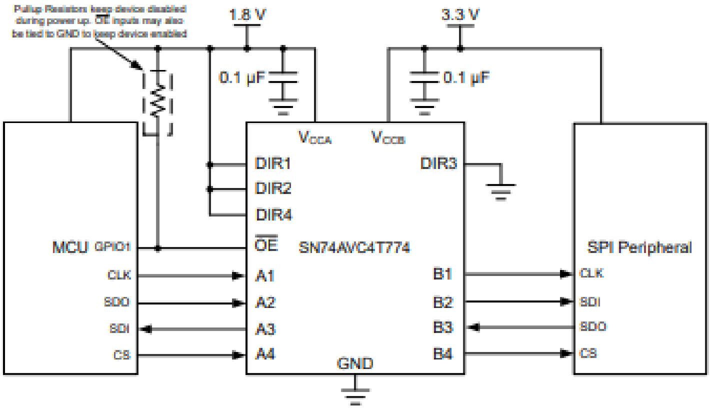


Figure 41. Level shifter connections according to its datasheet [11]

All the level shifters were used unidirectionally. Since in all the level shifters, the direction of the signal is from Ax to Bx, all the DIRx pins are connected to the VCCA voltage. The pin OE was connected to VSS permanently to keep the device always working.

Here is presented an example of one of the level shifters used in the project:

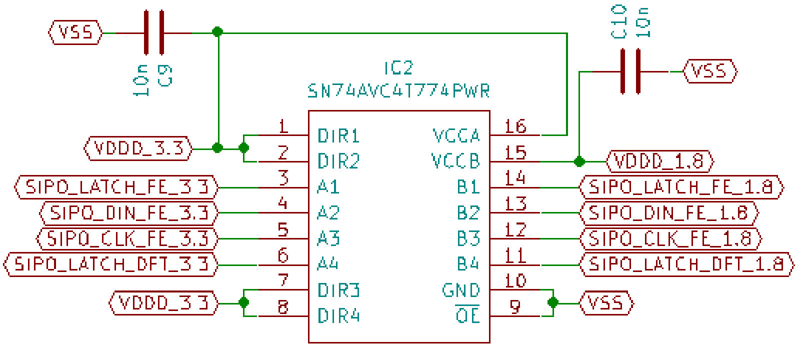


Figure 42. Level shifter schematic

3.5.2. Voltage regulators

As the chip and all the elements present in the PCB will be powered with the voltage provided by the FPGA, it is needed to add voltage regulators to generate 1.8V from the 3.3V supplied by the FPGA. The FPGA has 3 PMOD connectors and each connector has 2 VDD pins, then 3 different voltage regulators will be placed in the PCB, one for each PMOD connector.

This is the schematic of one of three voltage regulators:

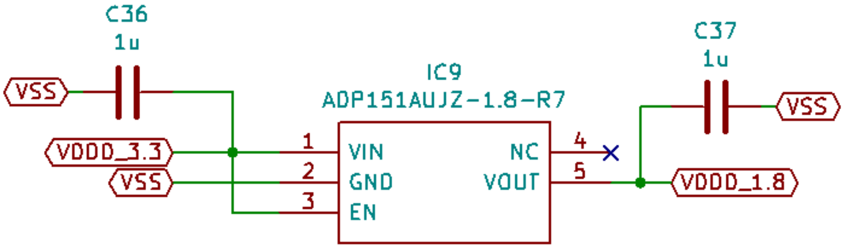


Figure 43. Voltage regulator schematic

3.5.3. Potentiometer

The voltage of the FE_VST pin of the chip should be controlled by a potentiometer. This is a pin of the ADC and its voltage must be between 0V and 1.8V. A rail-to-rail amplifier is used in order to match the impedances of the potentiometer and the FE_VST path since the impedance of this path inside the chip is unknown.

The main requirement for the rail-to-tail amplifier is its voltage operation. This voltage must be 1.8V as its output is an input of the chip. Then, in order to avoid any voltage glitch, it is better to not use a power voltage higher than the chip operation voltage. Furthermore, the amplifier should support single-supply operation as it is not possible to generate negative voltages nor they are supported by the chip. The chosen amplifier is the model TLV9041SIDBVR. This operational amplifier has a rail-to-rail input/output and a minimum supply voltage of 1.2V.

Below is shown the circuit implemented in the PCB:

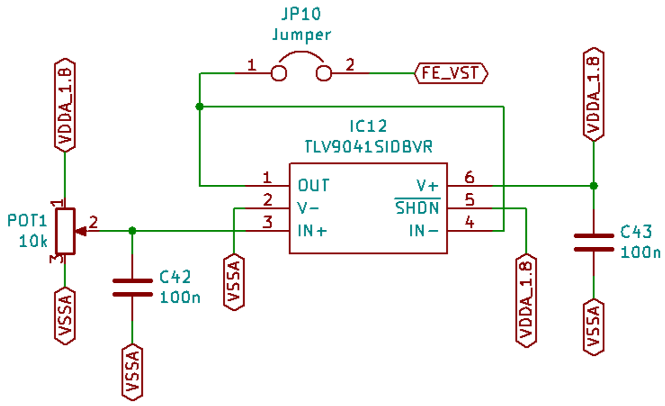


Figure 44. Potentiometer schematic

3.5.4. Single-ended to differential circuit

This circuit is used to drive two ADC input pins ADFT_INP and ADFT_INN with the same analog input signal. Instead of using a single-ended-to-differential amplifier of the market, it was decided to design it using two operational amplifiers. This is due to the circuit requirements. The amplifier must support-supply operation at 1.8V, which was not easy to find. Furthermore, the amplifiers that met these requirements did not accept a good range of input signal amplitudes and the output signals got saturated from input amplitudes above of 0.4V.

It was chosen the operational amplifier AD8515. It is a rail-to-rail amplifier that can be supplied with a minimum voltage of 1.8V and accepts single-supply operation. The designed single-ended to differential amplifier accepts a good range of input amplitudes. It can be used with input signals with a DC voltage of 0.9V and amplitudes up to 0.85V without output signals saturation. The reference voltage of the amplifier must be VDDA/2, i.e. 0.9V. This reference has been generated with a voltage divider.

Here is presented the schematic of the circuit:

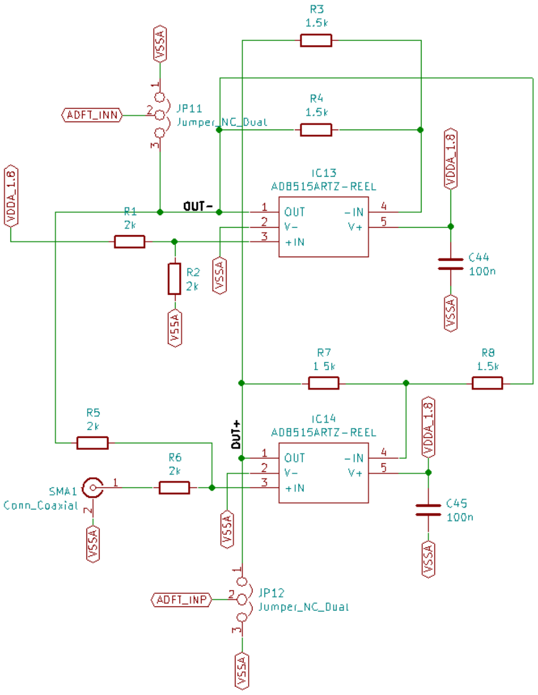


Figure 45. Single-ended to differential schematic

3.5.5. Differential to single-ended circuit

This circuit has 3 outputs. Two of them allow reading the signals from the ADFT_OUTP and ADFT_OUTN pins. The other output is differential since it is the difference between both signals. This last output signal is provided by a single-ended to differential amplifier.

The circuit contains two operational amplifiers as voltage followers in order to match the impedance between the chip pins and the output resistance of both non-differential outputs. As explained in Section 3.5.3, this is because the internal resistance of both pins is unknown.

The differential to single-ended amplifier does not need to be supplied by 1.8V since its output does not drive the chip. Then, there is no risk of glitches for the chip. This fact made it easier than in the former section to find a commercial differential to single-ended amplifier implemented in a single integrated circuit. As in the rest of the shown circuits, the amplifiers must support single-supply operation. Considering these requirements, the chosen model was THS4121. The voltage reference for the amplifier was also VDDA/2, but in this case, is 1.65V as the voltage supply of the amplifier is 3.3V. It was also generated using a voltage divider.

The schematic is shown below:

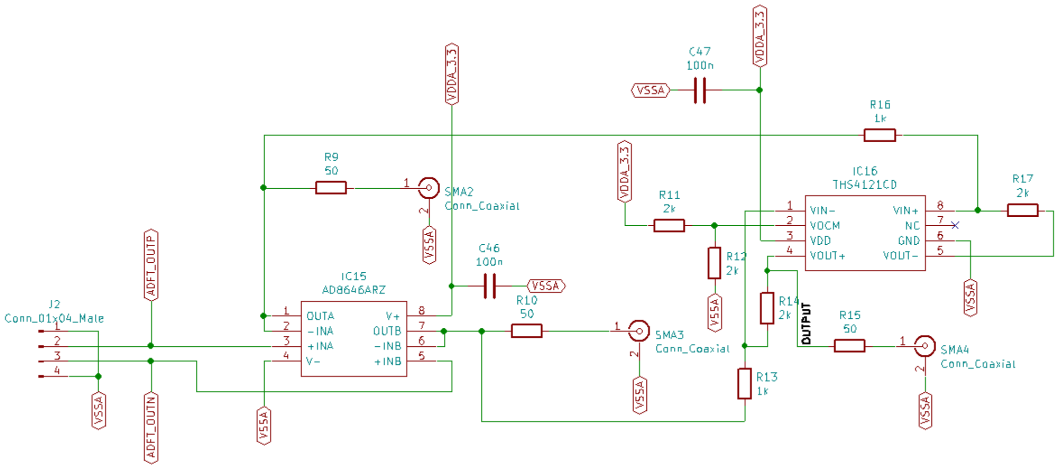


Figure 46. Differential to single-ended schematic

3.5.6. Temperature sensor

A temperature sensor is incorporated in the PCB to obtain data from the ambient temperature in order to calibrate the internal thermometer of the chip. Despite the created design in this project does not include the functionality to process data from the sensor, it could be added as an extra functionality in the future.

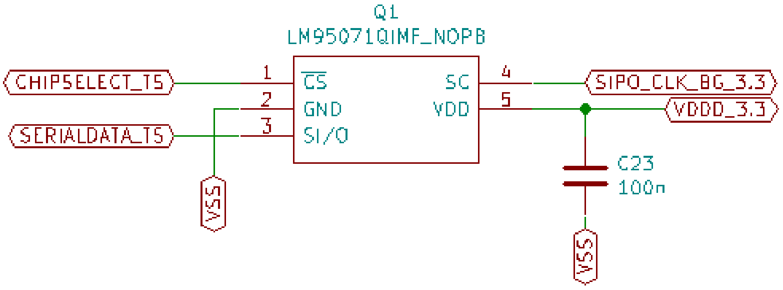


Figure 47. Temperature sensor schematic

The FPGA should generate the chip select and the clock signal needed for the communication. The serial data signal is connected to a GPIO pin of the FPGA in order to read the provided temperature data. Since there were no more GPIOs available to dedicate a pin exclusively to generate the clock signal of the sensor, the same clock signal used to transfer data to the BG SIPOs had to be used.

3.5.7. Footprint for capacitive sensor

Taking advantage of the fact that the ADC of the chip can be used as a standalone ADC, a footprint has been included on the PCB to be able to use a capacitive sensor. The idea is, in the future, to wire-bond the footprint with this sensor. In this way, the output data of the sensor could be digitised and processed.

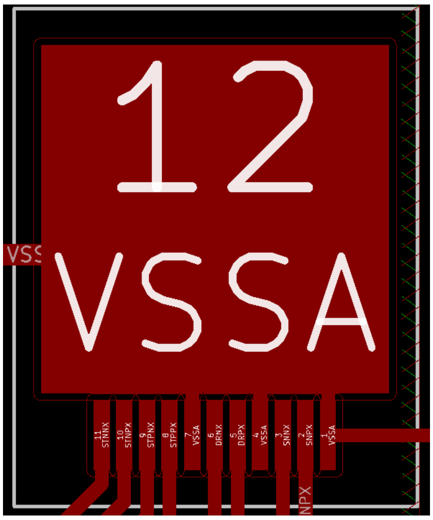


Figure 48. Footprint for a capacitive sensor

4. RESULTS

4.1. Simulation of the designs together with the chip

Once the designs to test the SIPOs and the PISO of the chip were individually tested and verified, it will be now tested both that the designs work correctly together with the block of the chip they are intended to test and that the blocks under test of the chip work correctly. The latter is the main objective of the project. This is done using a tool from the software Cadence called Virtuoso. In this tool, several tests were generated to verify how the designed testing blocks and the blocks to be tested work. The simulation used to launch every test is a mixed-signal simulation since despite the designed blocks are purely digital, the chip has both analog and digital parts.

4.1.1. Simulation of the BG PISO-SIPOs structure

Now, the designed PISO and the BG SIPOs of the chip are connected to verify if the data transmitted by the PISO is outputted by the SIPO blocks of the chip. If this is the case, it could be confirmed that the designed PISO and the SIPO of the chip works correctly.

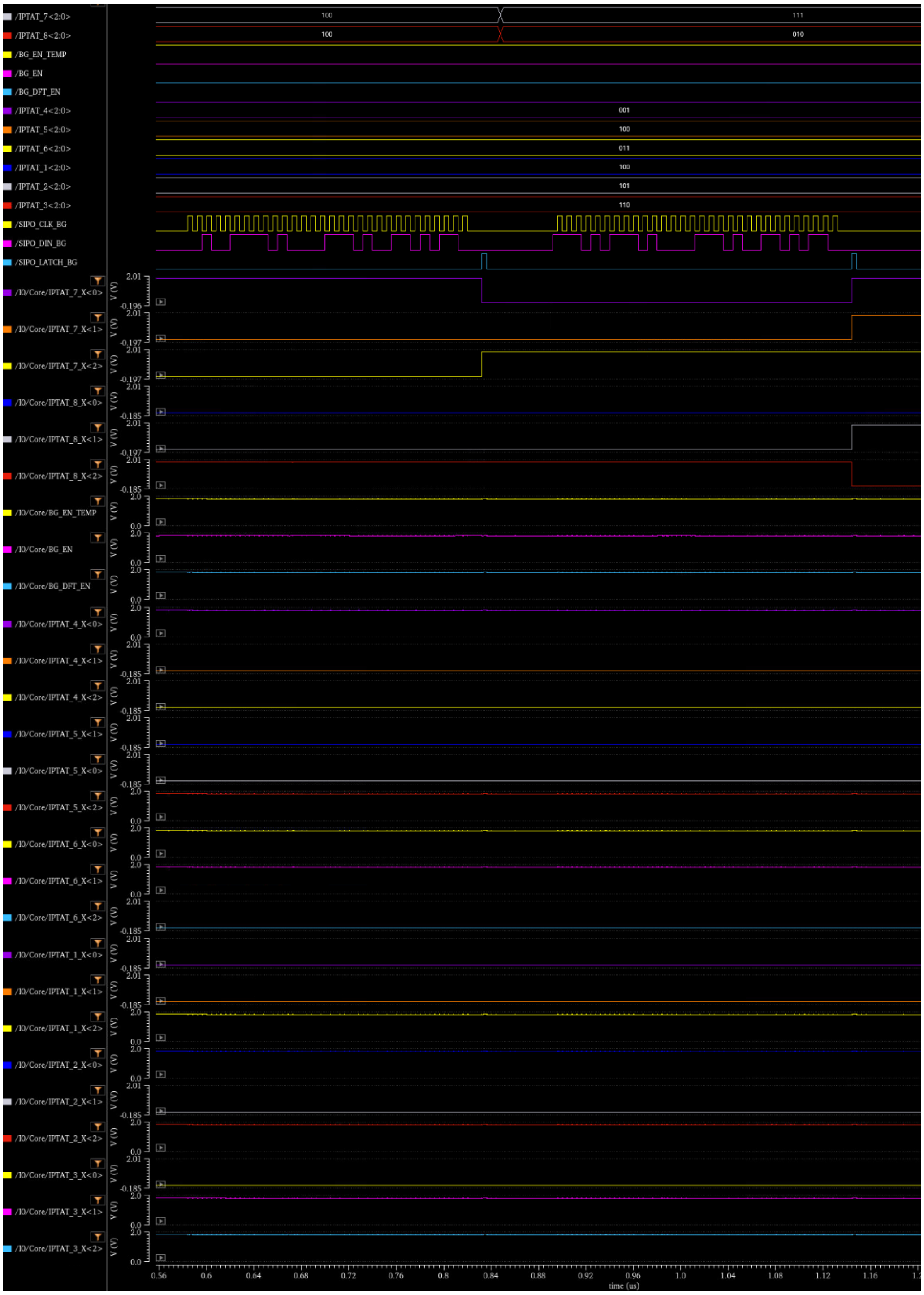


Figure 49. Simulation of the BG SIPOs test

From the simulation, all the signals above the communication signals (SIPO_LATCH_BG, SIPO_CLK_BG and SIPO_DIN_BG) are the input signals to be transmitted by the PISO. The signals below the communication signals are the output of the SIPOs. As can be seen, the input signals are perfectly received at the SIPOs outputs.

In the former simulation it is shown that all the parallel data is correct. Then, this data would arrive correctly at every block of the chip where they are connected. Now, it is presented the simulation of the BG PISO-SIPOs structure together with the test bench. This simulation is the procedure followed in the real world to verify the SIPOs.

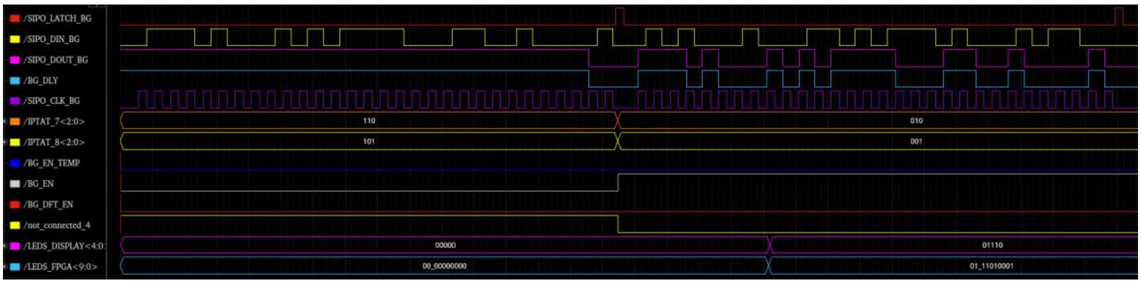


Figure 50. Reception of the sequence "0111010001" with no delay

It can be seen that the 5 most significant bits of the 30 bits sent to the BG SIPOs would be displayed in the FPGA LEDs through the LEDS_DISPLAY variable. The 10 most significant bits can also be found in the LEDS_FPGA internal variable.

Therefore, with the first simulation of this section is demonstrated that all the parallel input signals of the designed PISO are propagated in the SIPOs parallel outputs and, with the second simulation is also checked the proper operation of the SIPOs but also, this can be noted by the user in the FPGA.

Now is verified, how it would be the reception of data simulating a delay of 1 serial clock period (SIPO_CLK_BG) in the signal with which the data is received (SIPO_DOUT_BG):

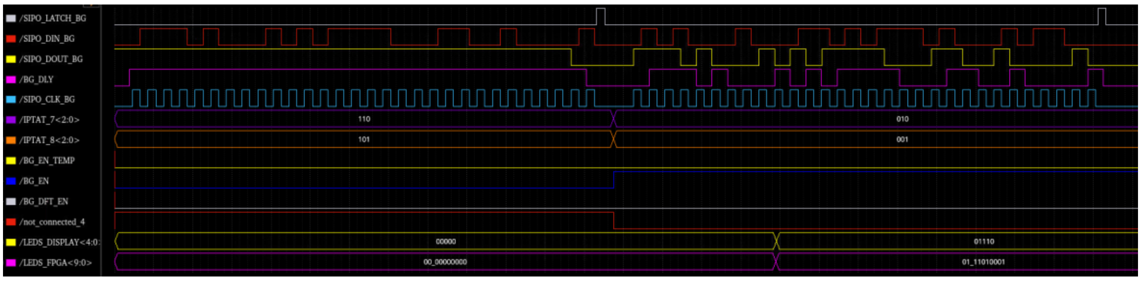


Figure 51. Reception of the sequence "0111010001" with 1 serial clock period delay

Note that the BG_DLY signal is the same signal as SIPO_DOUT_BG but with the delay applied. As can be seen, despite having a certain delay between the data sent by the chip (SIPO_DOUT_BG) and the signal received in the FPGA (BG_DLY), the obtained data sequence is exactly the same as in the previous simulation with no delay. This is thanks to the delay parameter of the test bench, which allows select the data capture instant.

4.1.2. Simulation of the DFT PISO-SIPO structure

Now, the designed PISO and the SIPO of the chip are connected to verify if the data transmitted by the PISO is outputted by the SIPO block of the chip. If this is the case, it could be confirmed that the designed PISO works correctly, as was found in the simulations with the test bench, and, more important, the SIPO of the chip works correctly.

The simulation below is the result of connecting the designed PISO with the DFT SIPO. It is verified that the parallel input data transmitted serially from the PISO to the SIPO is obtained at the parallel outputs of the SIPOs. Then, the 3 SIPOs work as expected.

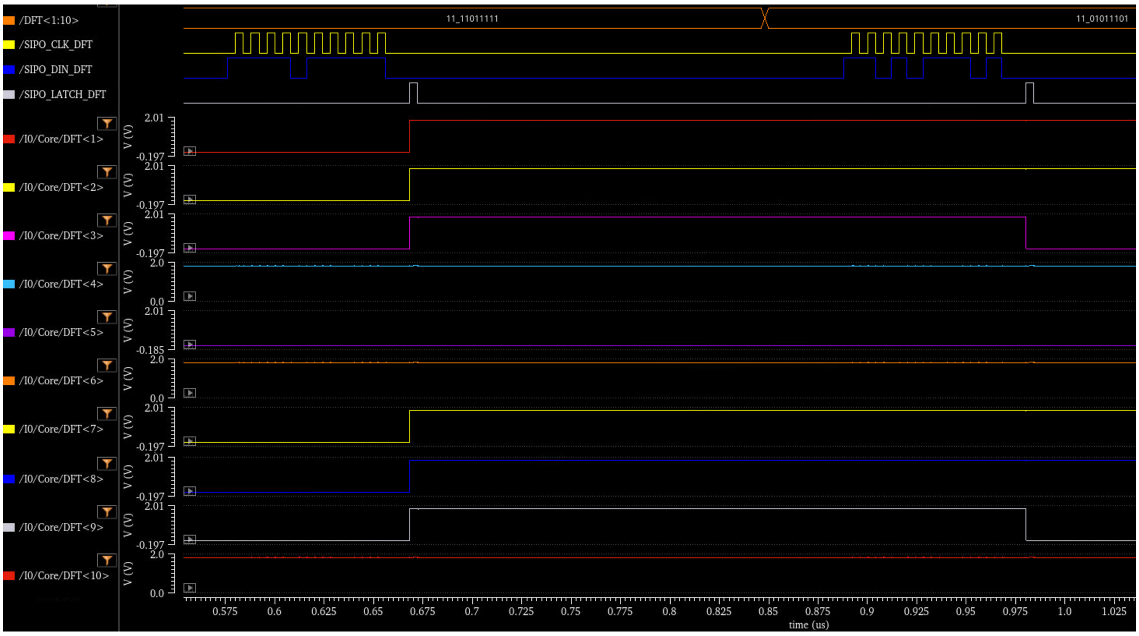


Figure 52. Simulation of the DFT SIPO test

In this simulation it is shown the transmission to the SIPO of two different data: “1111011111” and “1101011101”. It can be seen as the PISO serialises the input data DFT<1:10> and the same input data is present at the SIPO output. See the value of the signals from /I0/Core/DFT<1> to /I0/Core/DFT<10>. Once a pulse is generated in SIPO_LATCH_DFT, the SIPO outputs of the chip acquires the transmitted value. Therefore, both devices works correctly.

Below is verified the reception of the 10 bits of the DFT SIPO in the FPGA:

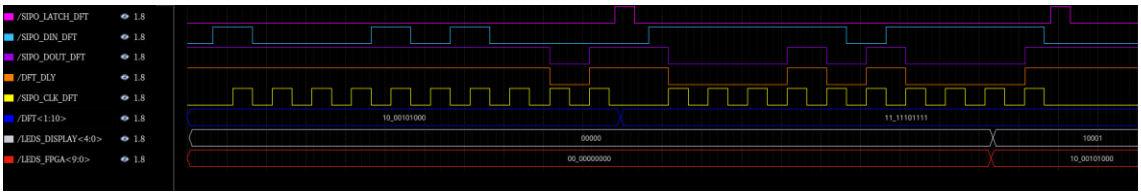


Figure 53. Reception of the sequence “1000101000” with no delay

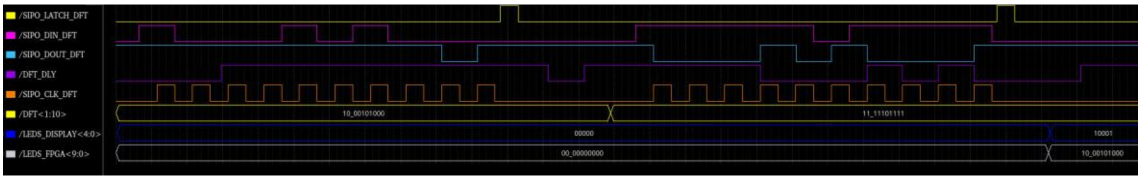


Figure 54. Reception of the sequence “1000101000” with 3 serial clock period delay

DFT<1:10> are the transmitted signals by the test bench to the PISOs to be transmitted by the PISOs to the DFT SIPO of the chip. LEDS_FPGA<9:0> are the received signals. It can be seen as despite having an important delay of 3 serial clock periods, the captured data is the same as in the Figure 53 with no delay.

4.1.3. Simulation of the FE PISO-SIPOs structure

Now, 2 different data transfers were simulated. First data starts transferring serially to the SIPOs (SIPO_DIN_FE), however, during this transfer some of the parallel input data was changed. Consequently, the transfer is aborted without producing a final pulse of SIPO_LATCH_FE, and a new transfer with the new data starts. Once this second transfer is completed, the SIPO_LATCH_FE pulse is produced in order to output all the data in the SIPOs.

On the other hand, it could be also verified that all the input data is correctly transferred to the SIPOs outputs. The simulation is split into 2 different figures (Figure 55 and Figure 56) as it did not fit on a single page.

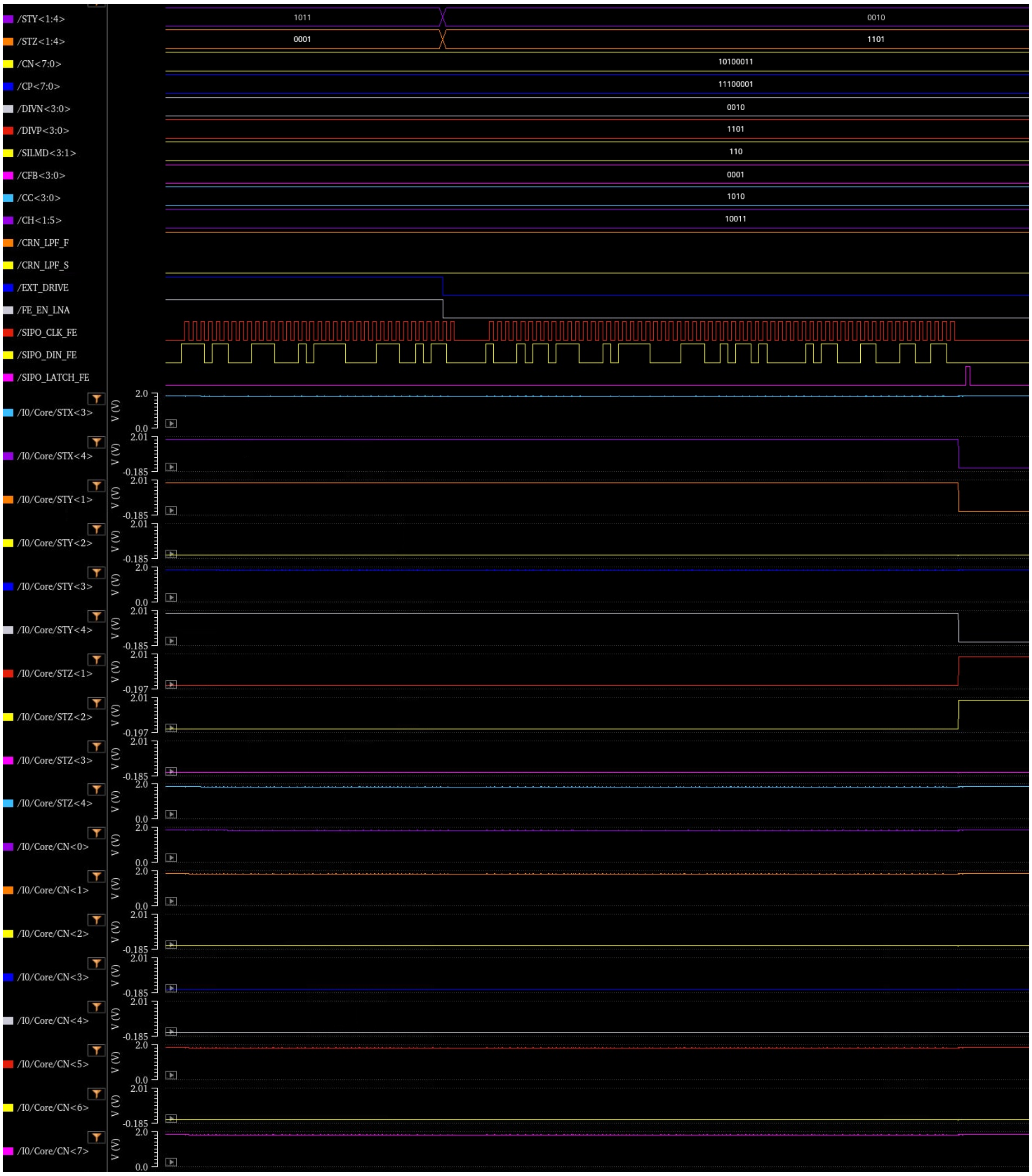


Figure 55. Simulation of the FE SIPOs test (part 1)

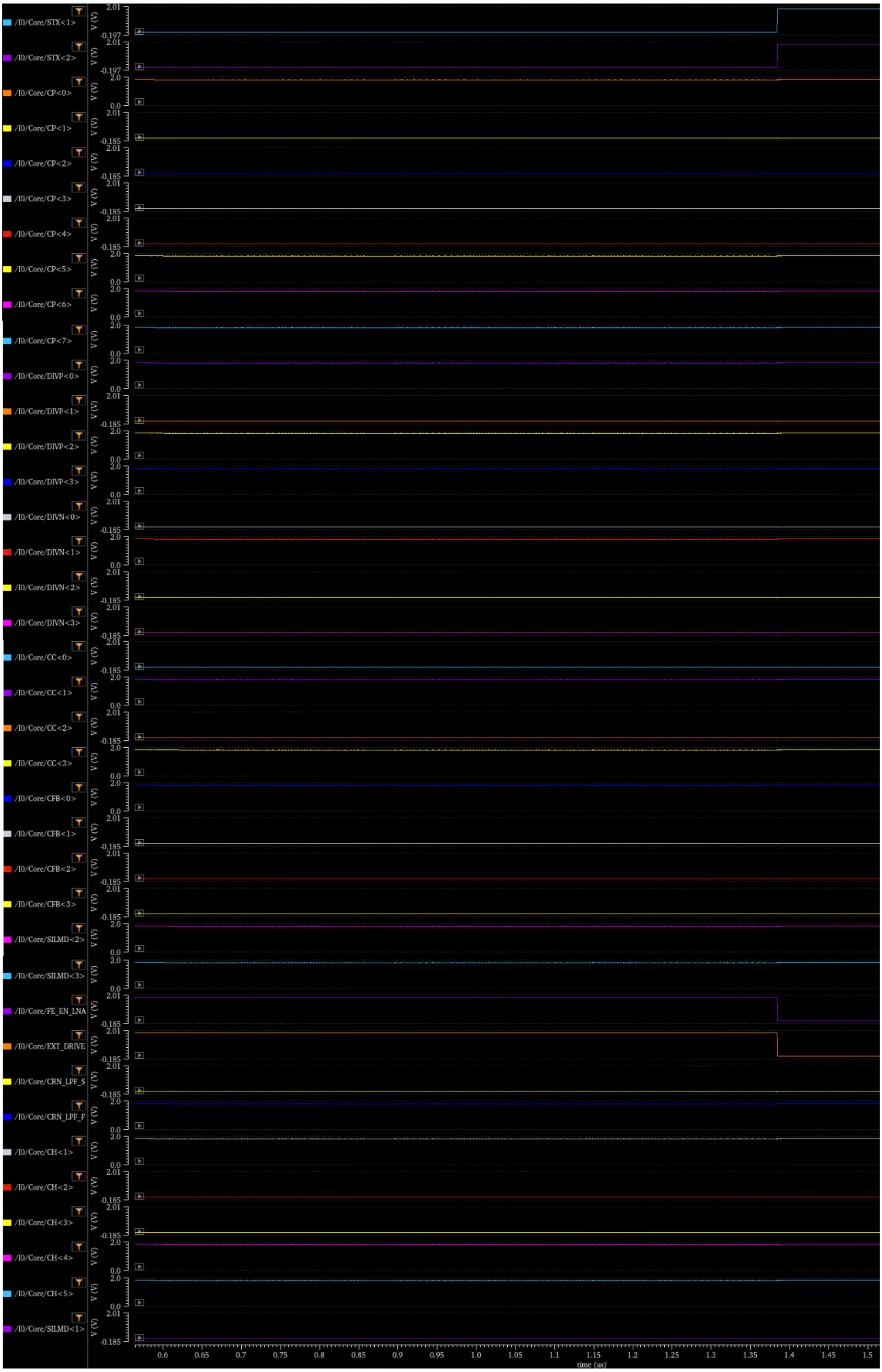


Figure 56. Simulation of the FE SIPOs test (part 2)

The Figure 56 is simply the continuation of the Figure 55. It only shows the rest of the output signals of the FE SIPOs so that they can be compared with the input signals of the PISO in the Figure 55.

Below it is verified that the 10 most significant bits from the 60 bits transmitted to the FE SIPOs are properly received:

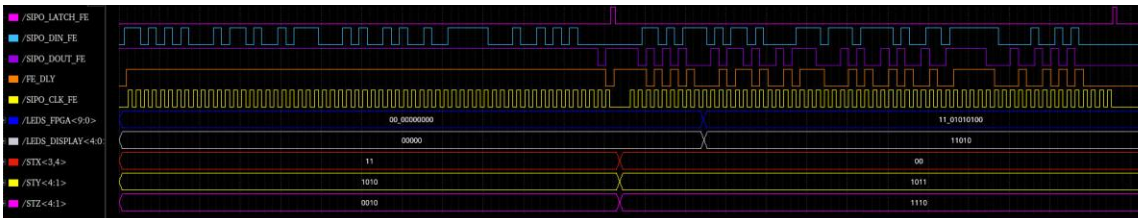


Figure 57. Reception of the sequence “1101010100” with no delay

It can be seen as the 10 most significant bits go through the 6 SIPOs and are output off-chip once the 10 first clock pulses of the next data transfer are produced. It is verified that the input data transmitted serially to the chip is received in the PCB. Therefore, the SIPOs operate as expected.

4.1.4. Simulation of the FECK PISO-SIPO structure

In this simulation it can be seen as all the input data transferred by the designed PISO is deserialised by the SIPO.

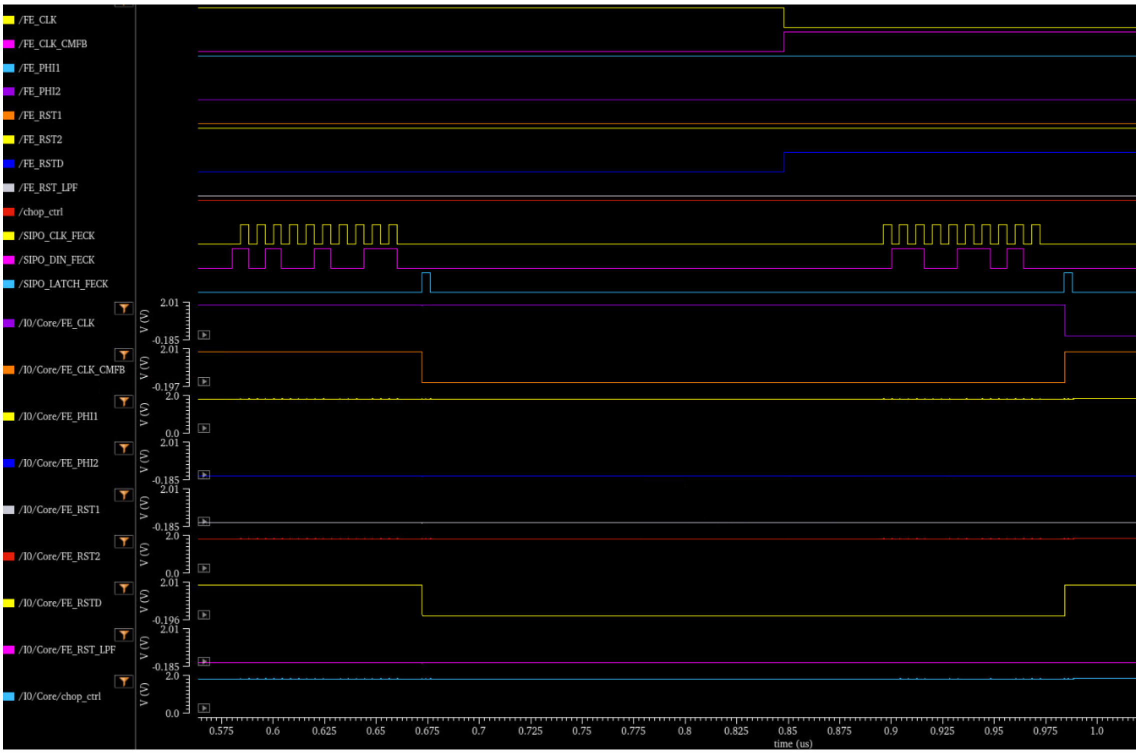


Figure 58. Simulation of the FECK SIPO test

Now it is shown the verification of the correct capture of 10 bits:

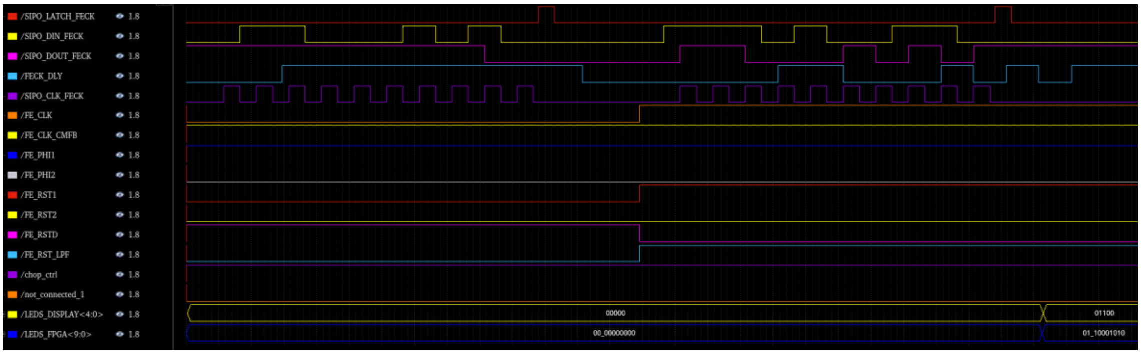


Figure 59. Reception of the sequence “0110001010” with 3 serial clock period delay (FECK SIPO)

4.1.5. Simulation of the ADC PISO-SIPO structure

The ADC SIPOs of the chip are not connected serially. As was explained in the Section 3.1.6, actually, they can be considered single SIPOs and, consequently, simulated individually. The output of every SIPO is not connected to another register but transmitted off-chip.

Now, it is shown the simulation of every ADC PISO-SIPO structure. Note that the clock and the latch signal is shared by the four SIPOs. This means that they deserialise the data at the same instant, once they receive the latch signal.

In order to simplify and facilitate the understanding of the simulations. A different simulation has been carried out for each PISO-SIPO structure.

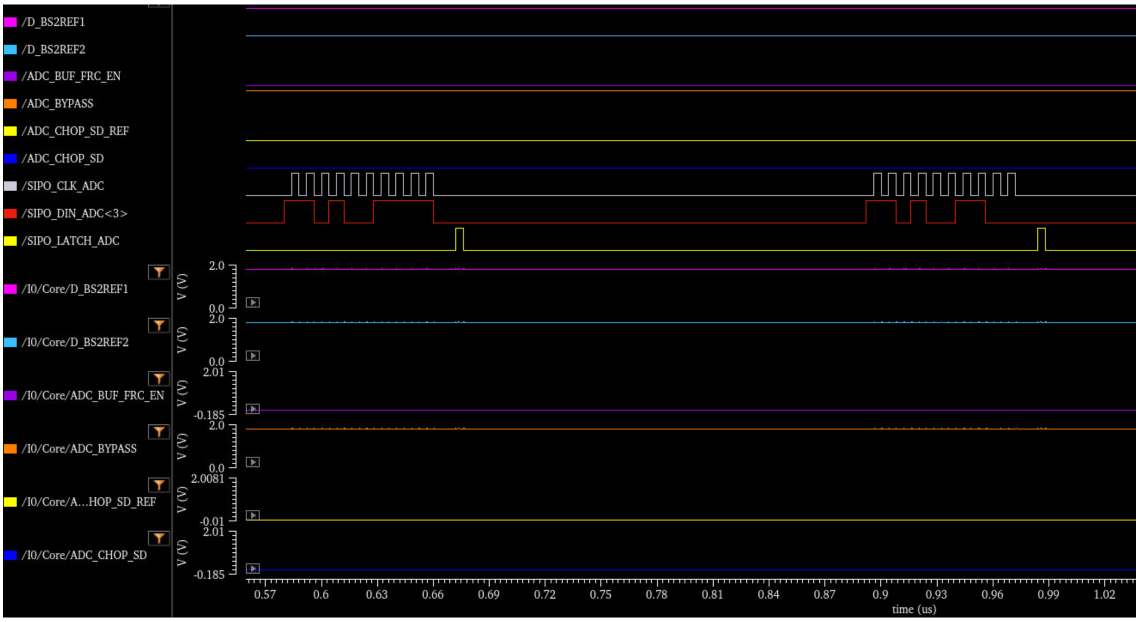


Figure 60. Simulation of the test of the ADC SIPO block 3

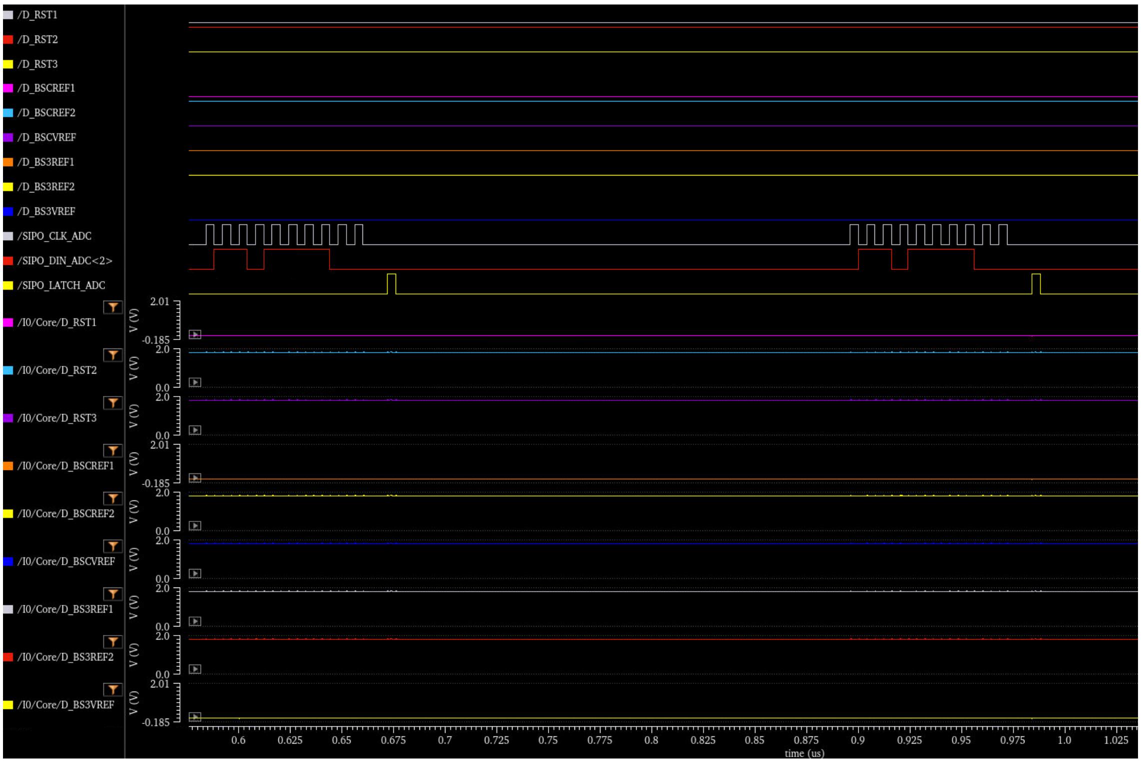


Figure 61. Simulation of the test of the ADC SIPO block 2

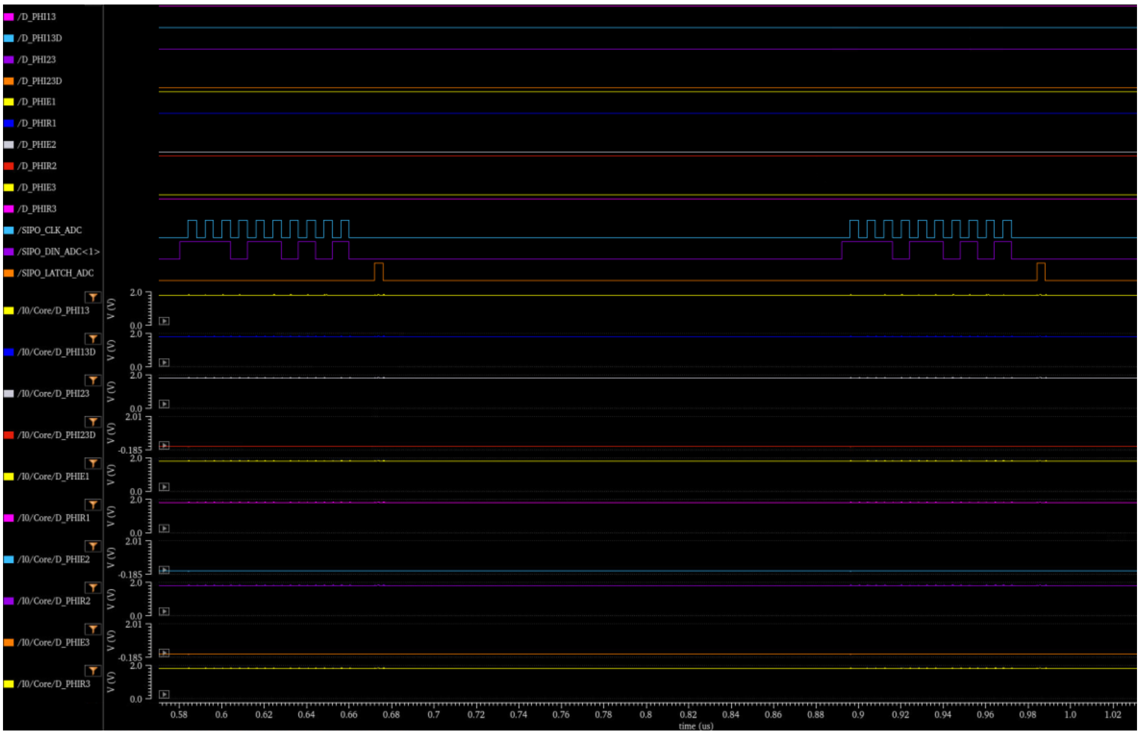


Figure 62. Simulation of the test of the ADC SIPO block 1

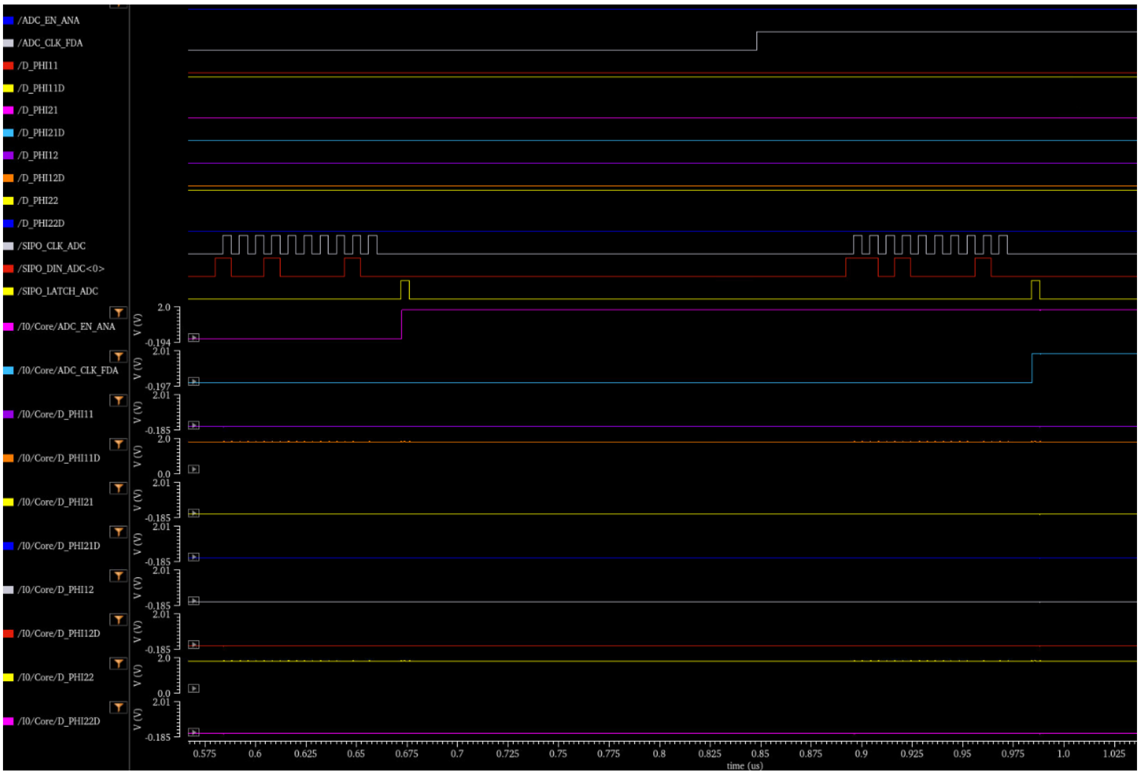


Figure 63. Simulation of the test of the ADC SIPO block 0

It is important to realize that if an input of a single ADC SIPO changes and the rest of the inputs of the rest of the SIPOs remain the same, a new transfer will start to all the SIPOs, including those SIPOs whose input data has not changed.

In the case of the simulation, it can be seen the only input data that has changed is the ADC_CLK_FDA signal from the SIPO that receives the serial input data from the SIPO_DIN_ADC<0> signal (see Figure 63). The 4 least significant bits of the SIPO are not shown in the simulation since, by the design of the chip, the least important outputs of this register remain unconnected.

Below are presented the simulations of the 4 ADC SIPOs to show how the 10 bits would be received in the FPGA:

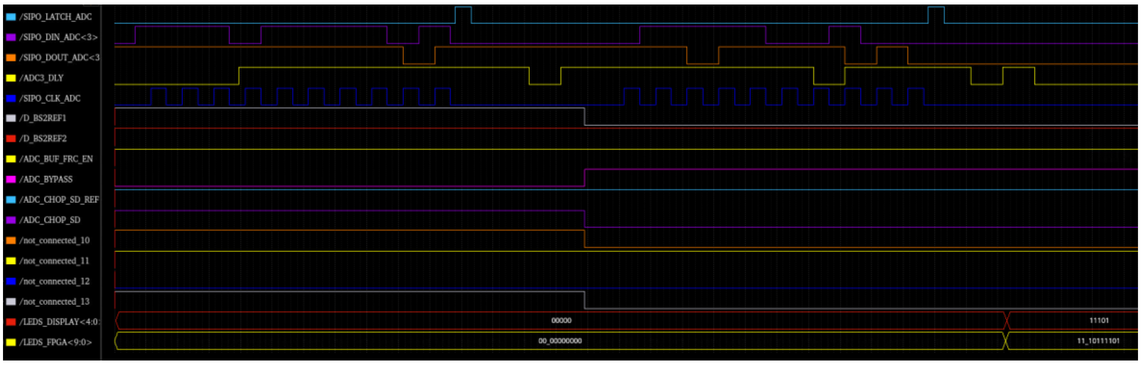


Figure 64. Reception of the sequence “1110111101” with 4 serial clock period delay (ADC SIPO block 3)

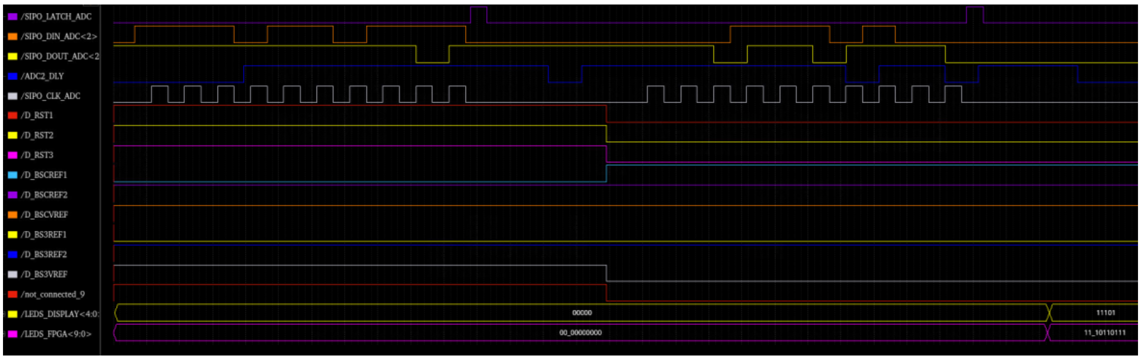


Figure 65. Reception of the sequence “1110110111” with 4 serial clock period delay (ADC SIPO block 2)

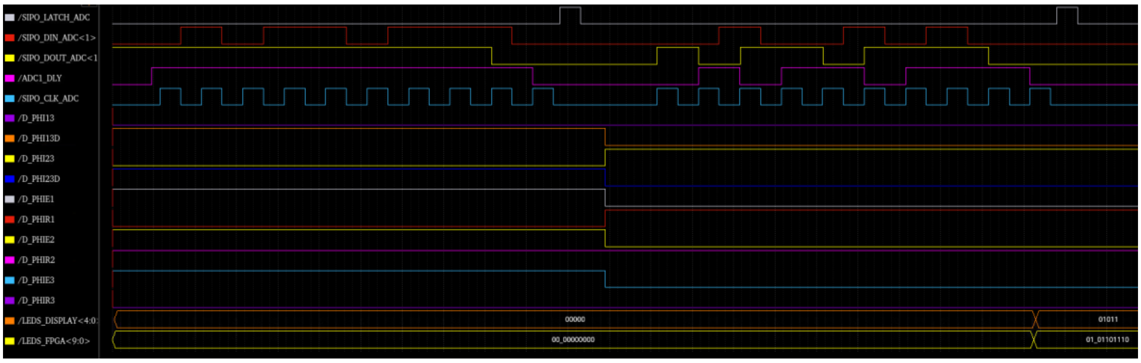


Figure 66. Reception of the sequence “0101101110” with 1 serial clock period delay (ADC SIPO block 1)

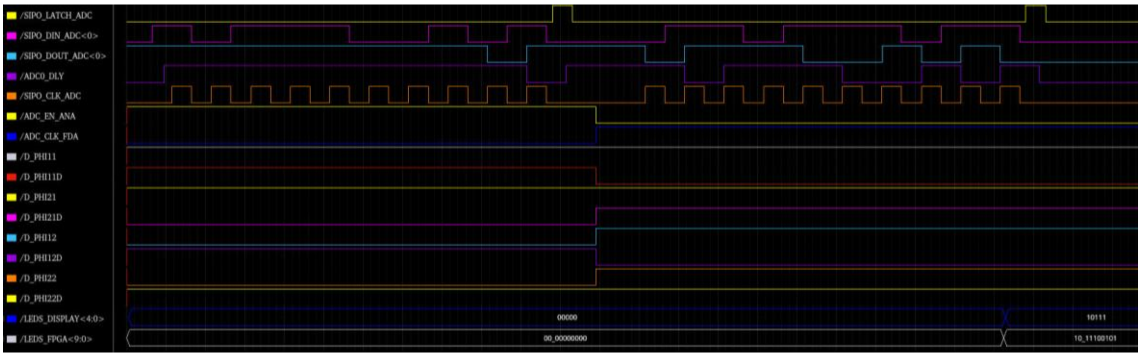


Figure 67. Reception of the sequence “1011100101” with 1 serial clock period delay (ADC SIPO block 0)

4.1.6. Simulation of the SIPO-PISO structure

In this section, it is tested the PISO contained in the chip together with the designed SIPO. To do it, parallel data was transmitted to the PISO contained in the chip thanks to the input data generator block shown in the Section 3.3.2. This is the set of blocks that was simulated:

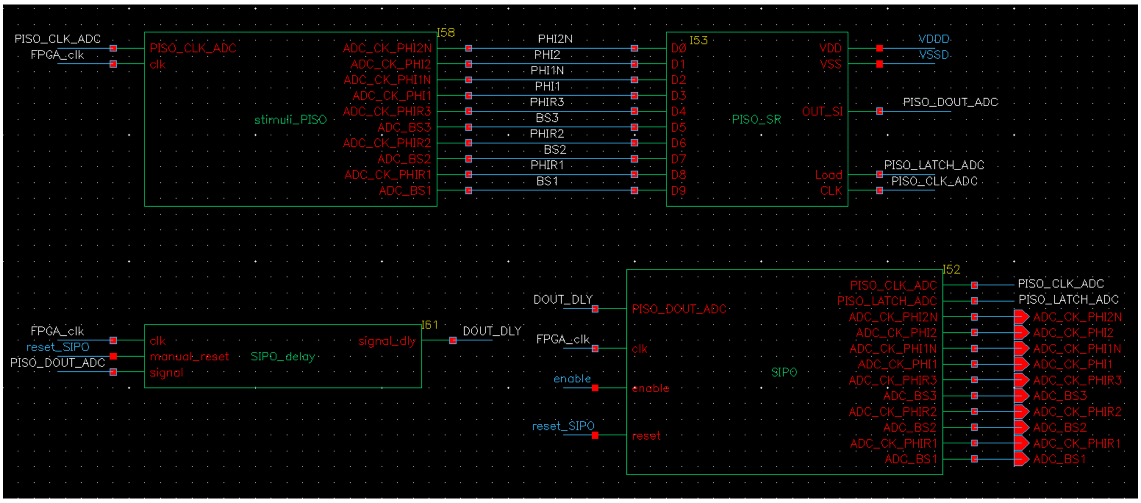


Figure 68. Schematic for PISO testing

It is interesting to note that in order to simulate the delay between the clock and the serial data signal due to the physical elements between chip and FPGA, a delay block was added. This delay block delays 5 system clock cycles the serial data signal transmitted from the PISO to the SIPO (PISO_DOUT_ADC). Then, there is a delay in this signal between transmitter and receptor, but more important, the signal is delayed regarding the PISO_CLK_ADC generated by the SIPO.

In the following simulation, it will be shown how the designed SIPO captures and deserialises the serial data received thanks to the delay parameter adjustment and despite having added a considerable delay between the data and the clock:

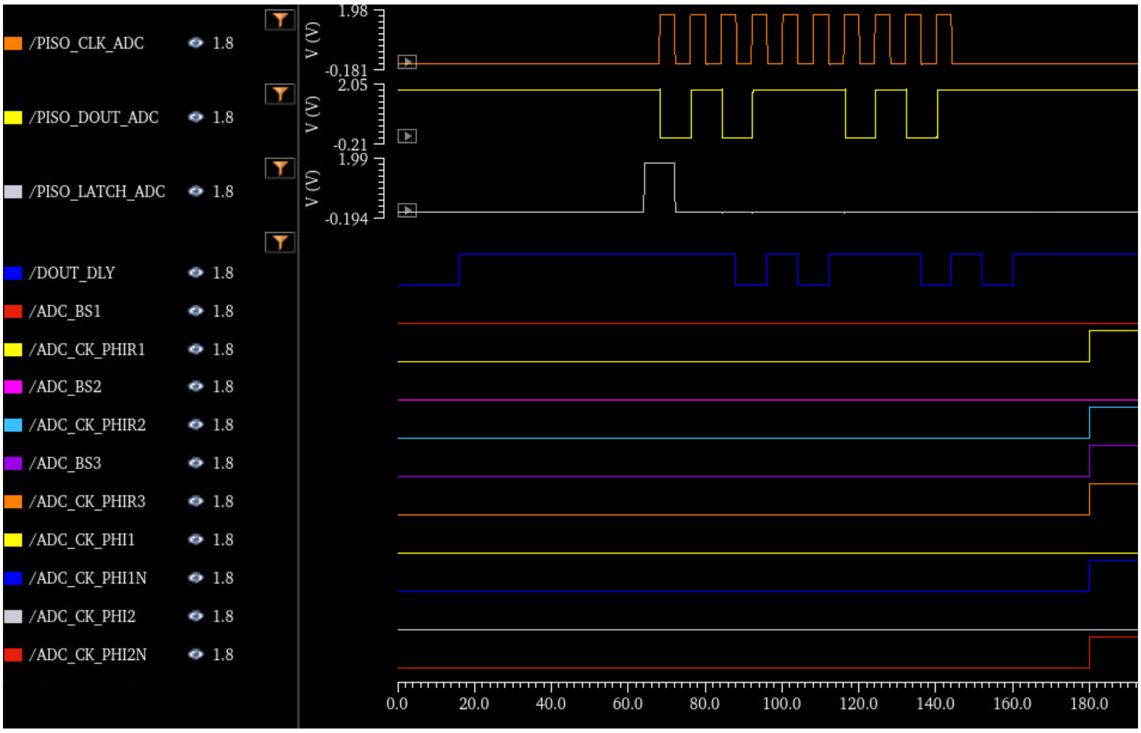


Figure 69. Capture and deserialisation of data done by the designed SIPO

As a clarification to be able to interpret the simulation, the signals are displayed from top to bottom according to the weight of each bit. The signal of the top (ADC_BS1) corresponds to the most significant bit of the serial data signal. DOUT_DLY is the signal used by the SIPO to capture the data.

The PISO of the chip transmits every bit in the positive edge of the clock pulses and, by default, the SIPO captures the data also in the positive clock edge but starting from the second clock pulse. However, this default case only considers a minimum delay between the clock and data. The simulation shows a higher delay between PISO_CLK_ADC and DOUT_DLY. There is a delay of two and a half cycles of the PISO_CLK_ADC clock. According to the simulation below, the solution to capture the correct data from the DOUT_DLY signal is to start capturing the data in the positive clock edge of the fourth clock pulse, instead of in the positive clock edge of the second clock pulse. To do it, the delay parameter of the SIPO was set to 4.

This is clearly seen in the image below:

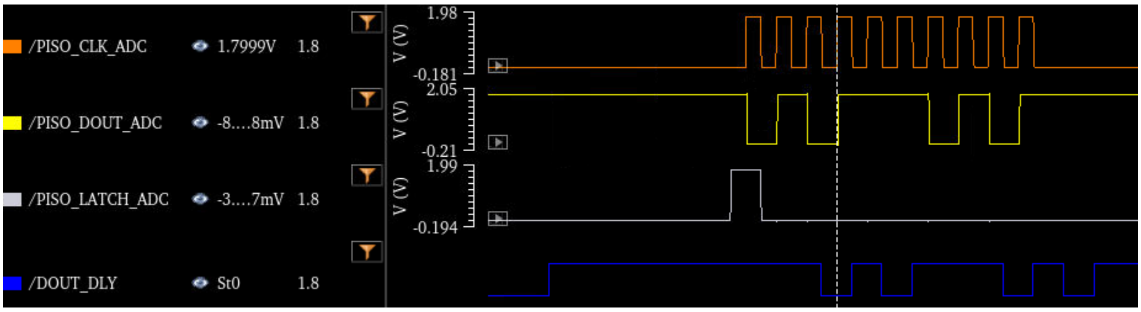


Figure 70. Data capture start in the fourth clock pulse

In the simulation below are shown two complete data transfers and it can be seen as the input data inserted in the PISO and the output of the designed SIPO match:

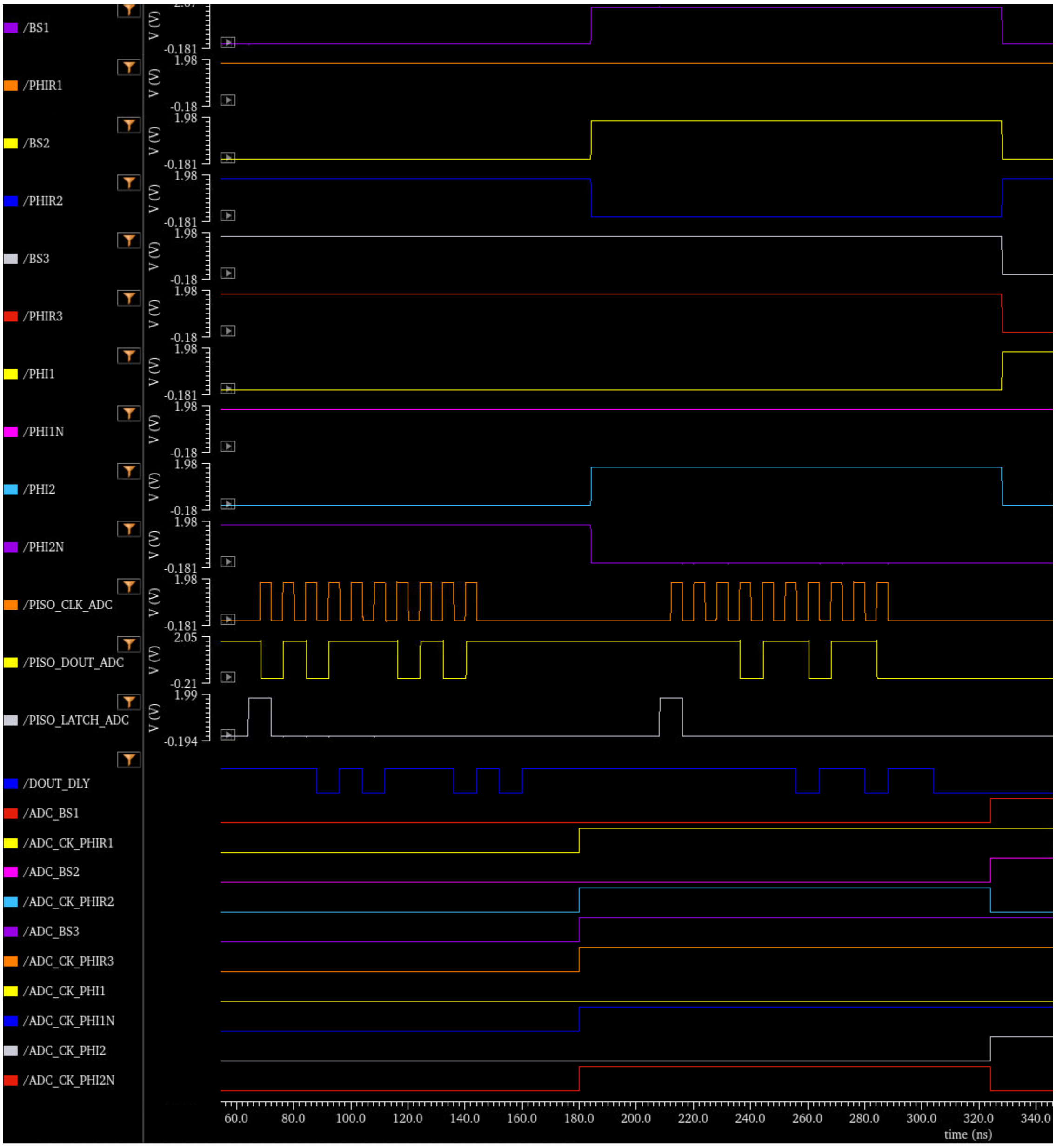


Figure 71. Simulation of the PISO test

The signals above the communication signals (PISO_CLK_ADC, PISO_DOUT_ADC and PISO_LATCH_ADC) are the input signals inserted by the input data generator block in the PISO and the signals below the communication signals correspond to the output signals of the designed SIPO. Note how the output signals are updated with the values of the inputs once the 10-pulse clock cycle ends, which means that the 10 bits of data have been serially transmitted. Thus, the values of the outputs at the end of a serial transfer take on the same value that the inputs had before the same serial transfer began. Then it is verified that both input and output match.

4.2. PCB circuit simulations

In this section it will be simulated the more complex circuits of the PCB. Since they are more complex as the rest of the circuits, it is needed to verify that the designs meet the requirements. These circuits are the single-ended to differential circuit and the differential to single-ended circuit.

4.2.1. Single-ended to differential circuit simulation

The single-ended to differential amplifier designed with operational amplifiers was simulated in LTspice in order to verify its operation. Usually, the input signal will have a maximum amplitude of 500mV and a DC voltage of 900mV.

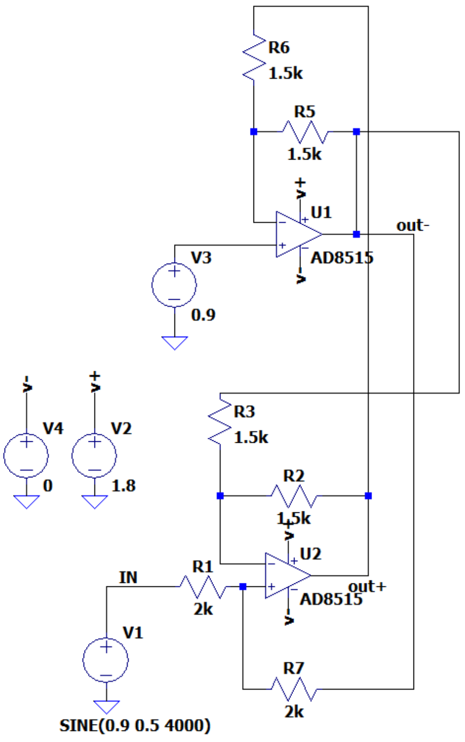


Figure 72. Single-ended to differential circuit used for the simulation

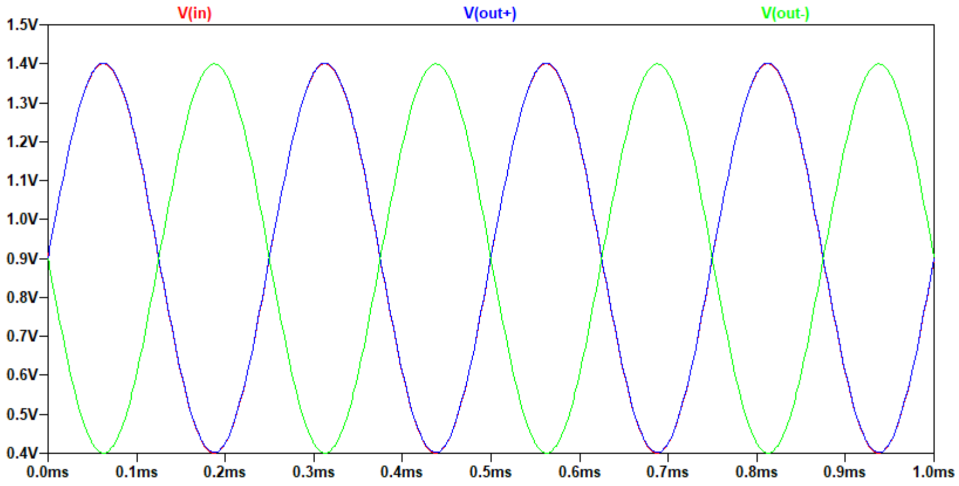


Figure 73. Single-ended to differential circuit simulation

It is verified that the design works as expected. The input signal is at both outputs which are connected to the chip inputs ADFT_INP and ADFT_INN.

4.2.2. Differential to single-ended circuit simulation

The ADFT_OUTP and ADFT_OUTN signals on the chip are the input signals of the circuit, V1 and V2. For the simulation, two signals with a common voltage of 900mV and amplitudes of 800mV and 300mV have been used.

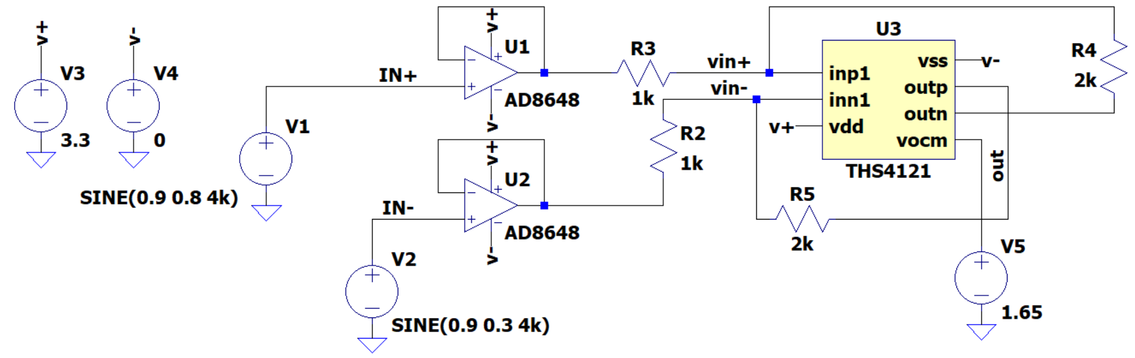


Figure 74. Differential to single-ended circuit used for the simulation

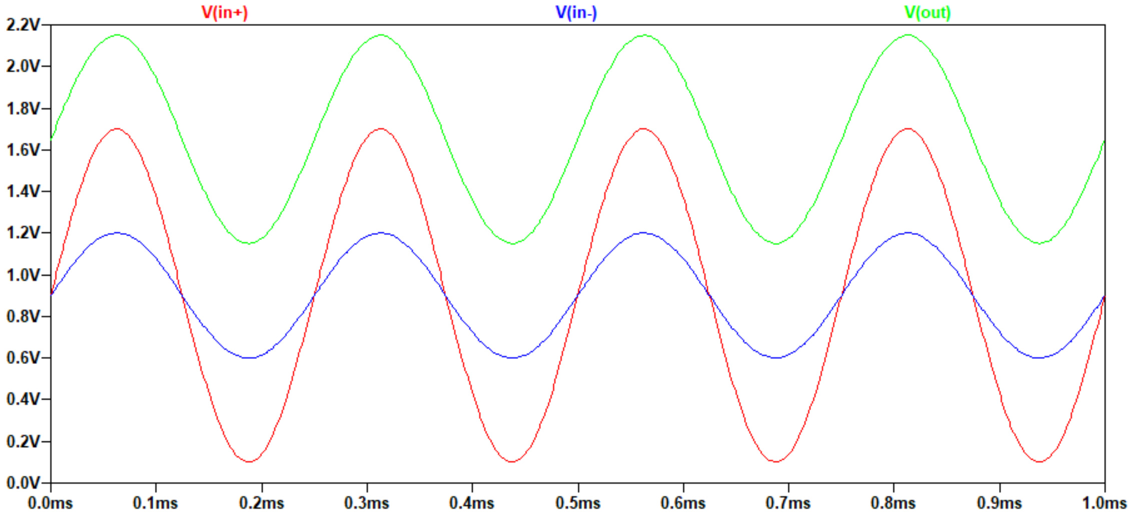


Figure 75. Differential to single-ended circuit simulation

4.3. Laboratory verification

In this section are shown the results of the design verification process performed in the laboratory.

4.3.1. PCB final result

It is presented the final version of the PCB:

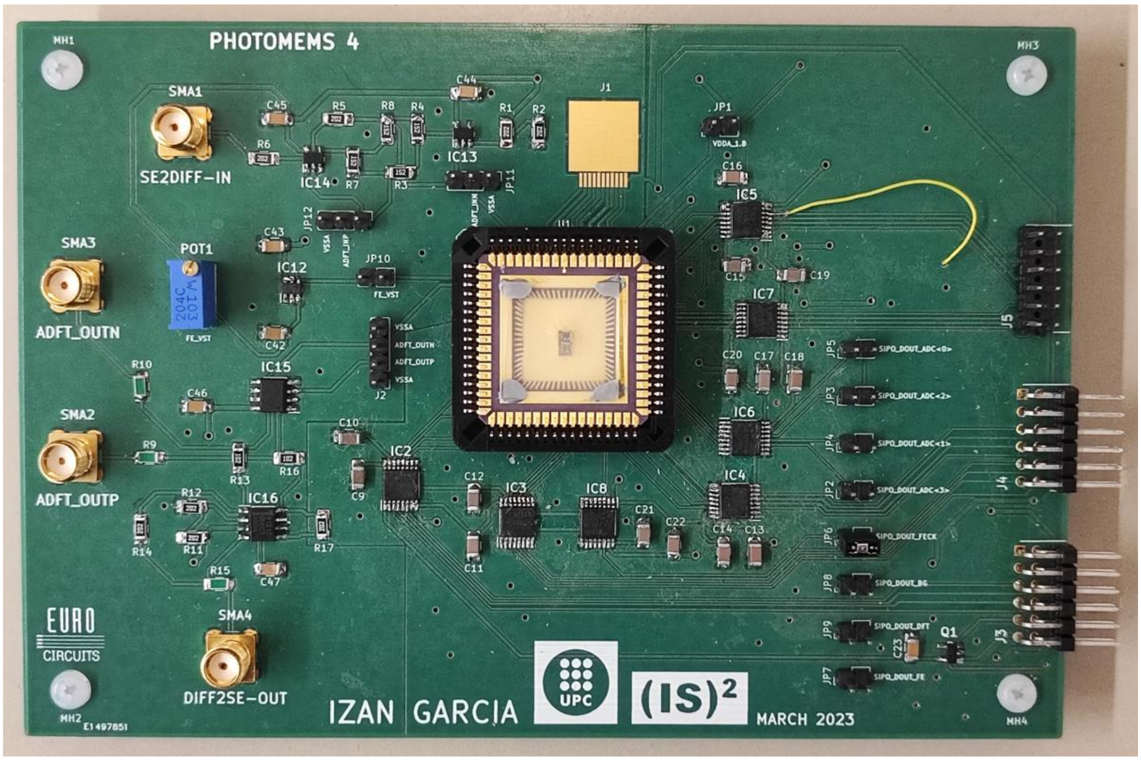


Figure 76. Top layer of the PCB

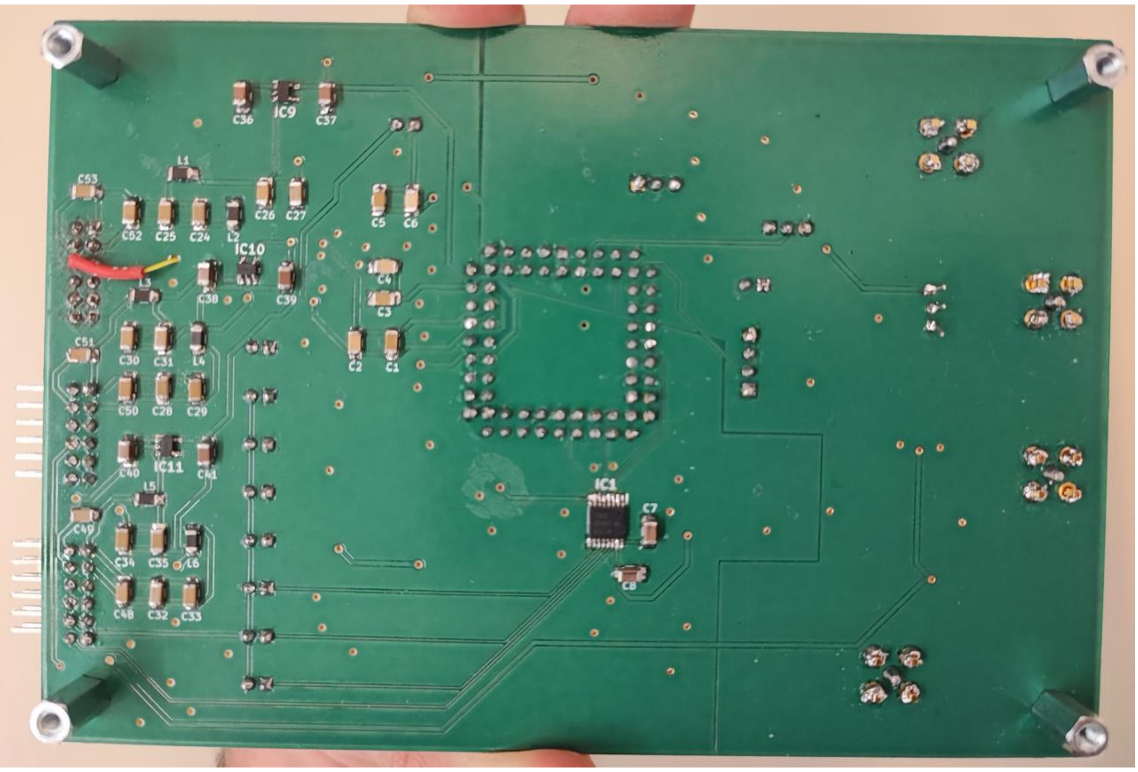


Figure 77. Bottom layer of the PCB

4.3.2. Complete system

It is shown the complete system Chip-PCB-FPGA:

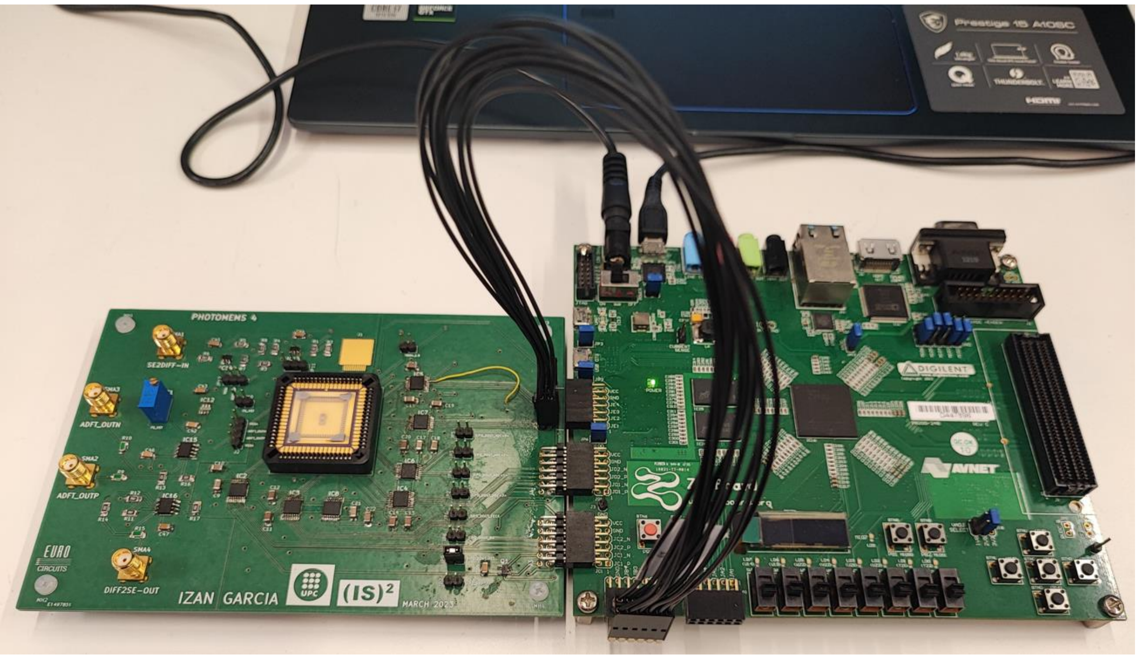


Figure 78. Complete system connected

4.3.3. SIPOs verification

In this section it is verified the operation of all the SIPOs. It is done the same verification as in the simulations with the test benches but practically in the laboratory. It is transmitted the same data sequence (10 bits) as in the test benches simulations of the Section 4.1.

To verify the SIPOs operation it has been used 5 LEDs of the FPGA, from LD0 to LD4, and some switches.

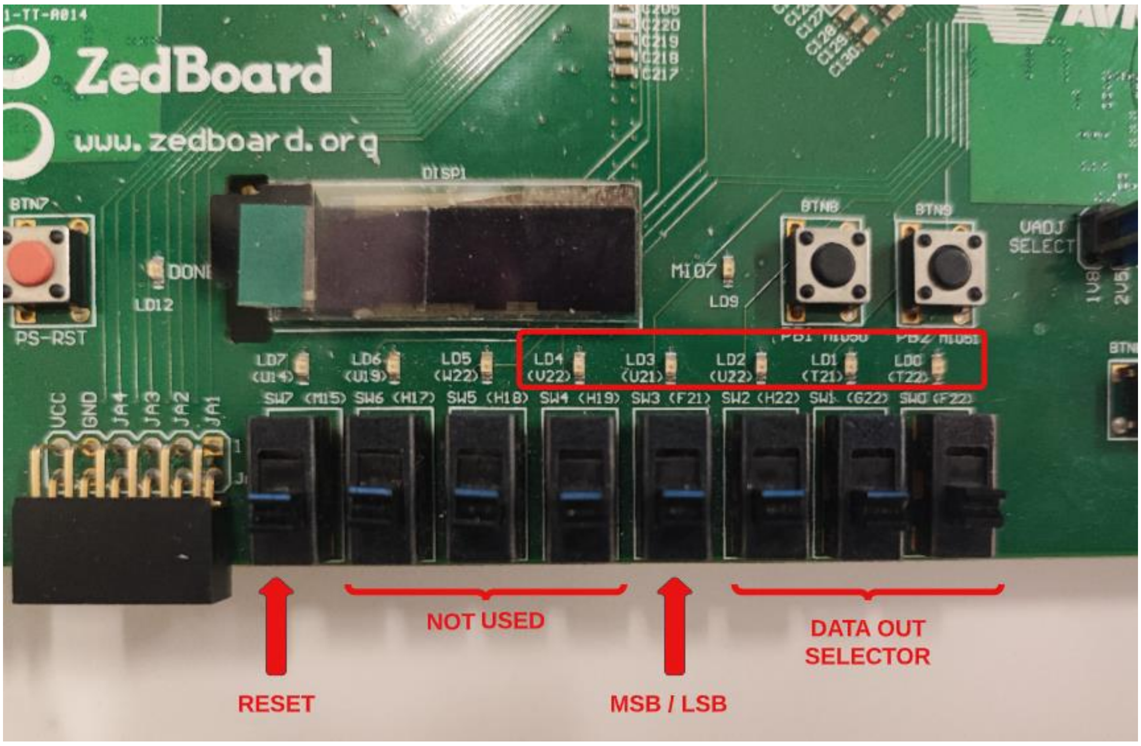


Figure 79. FPGA resources used for the verification process

In the next table is summarised the switches combinations of the data out selector to choose what SIPO output is desired to show in the LEDs.

SIPO output	Switch 2 (SW2)	Switch 1 (SW1)	Switch 0 (SW0)
FECK	OFF	OFF	OFF
DFT	OFF	OFF	ON
ADC0	OFF	ON	OFF
ADC1	OFF	ON	ON
ADC2	ON	OFF	OFF
ADC3	ON	OFF	ON
BG	ON	ON	OFF
FE	ON	ON	ON

Table 2. Combination of switches to select the output data

The following images show the verification of the SIPOs operation:

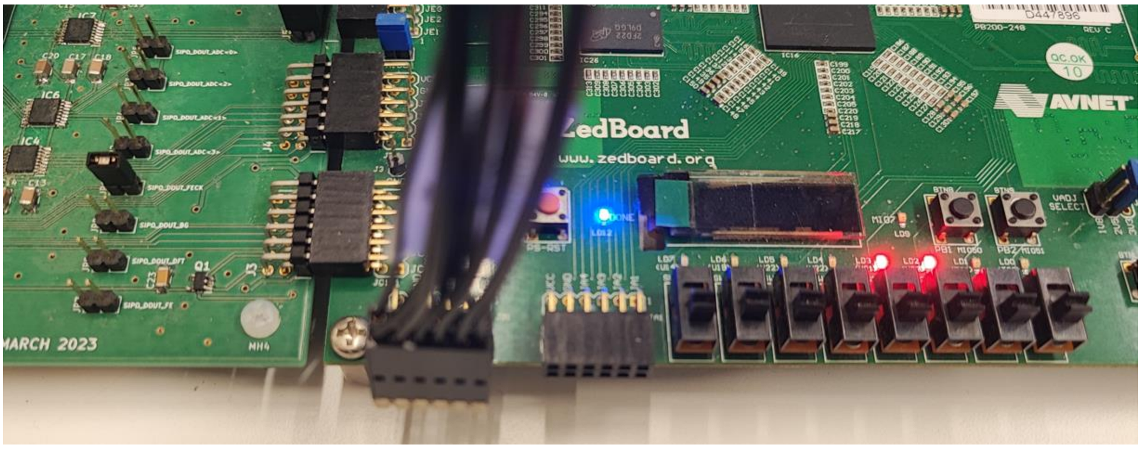


Figure 80. Reception of the data transmitted to the FECK SIPO (MSB)

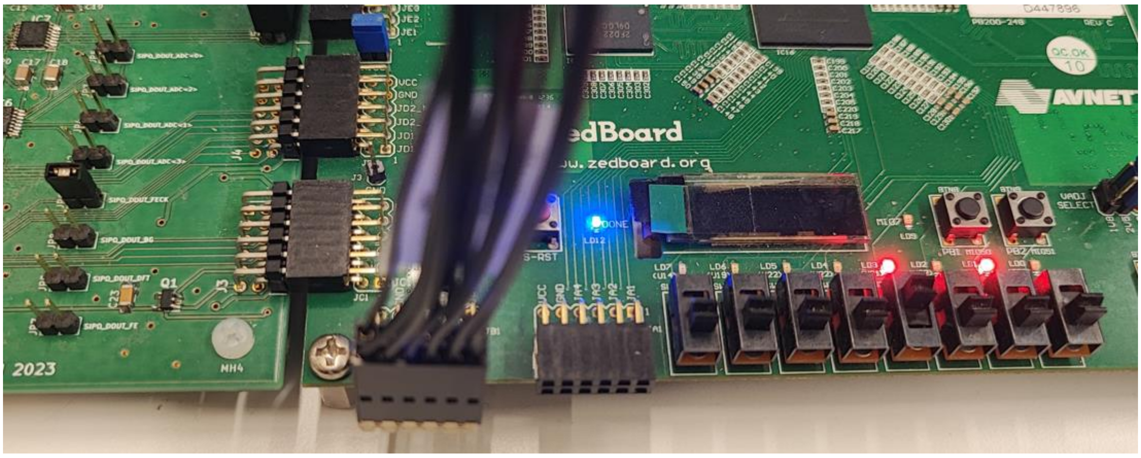


Figure 81. Reception of the data transmitted to the FECK SIPO (LSB)

The complete sequence received is “0110001010”. It can be compared with the simulation (Figure 59) that this is the expected data.

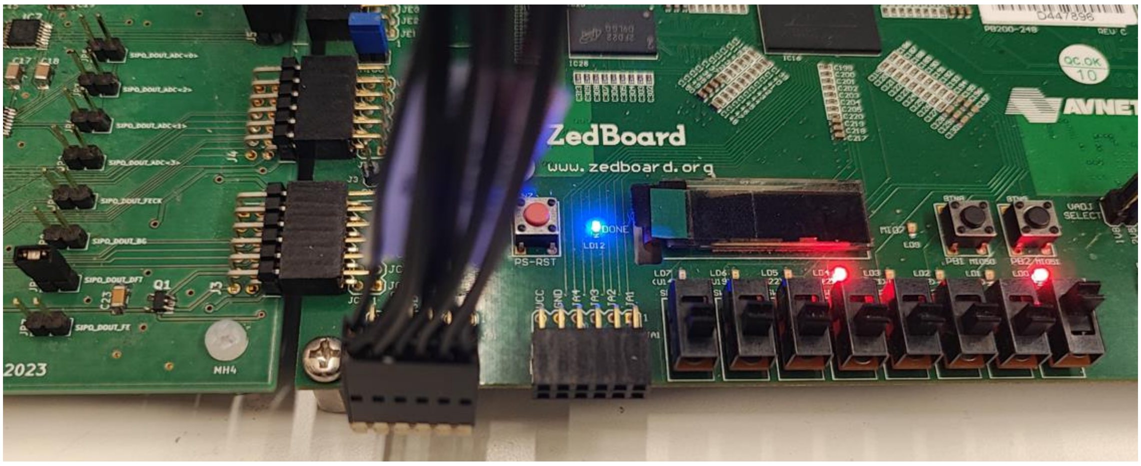


Figure 82. Reception of the data transmitted to the DFT SIPO (MSB)

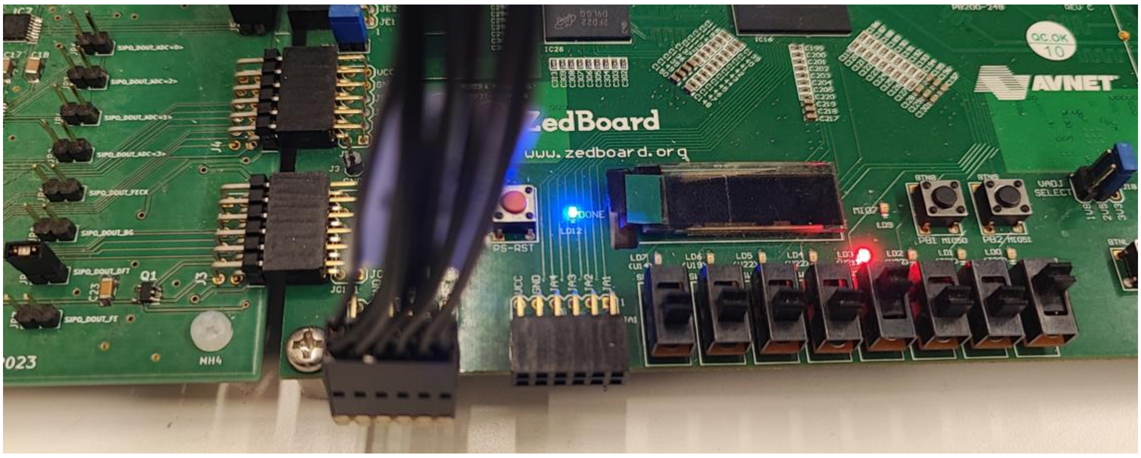


Figure 83. Reception of the data transmitted to the DFT SIPO (LSB)

The complete sequence received is “1000101000”. It can be checked with the simulation (Figure 53) that this is the expected data.

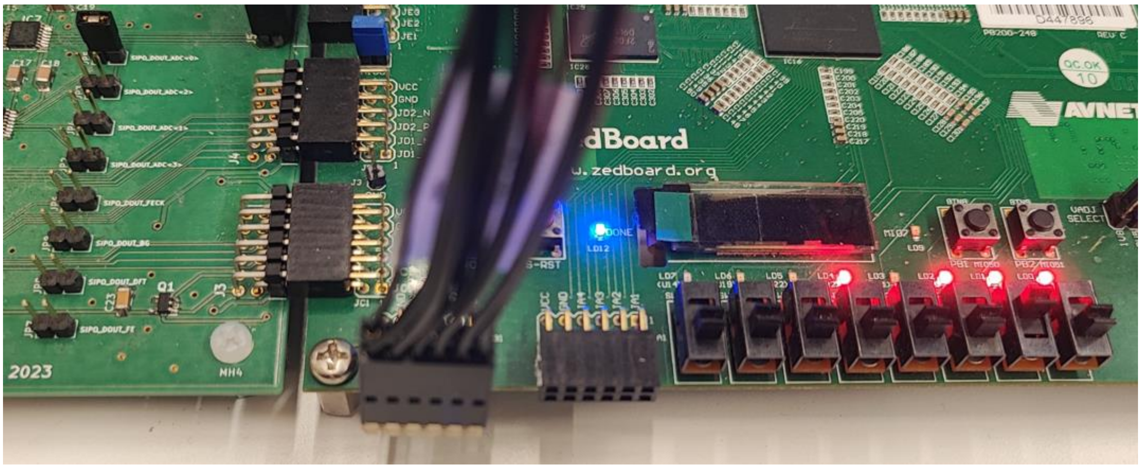


Figure 84. Reception of the data transmitted to the ADC0 SIPO (MSB)

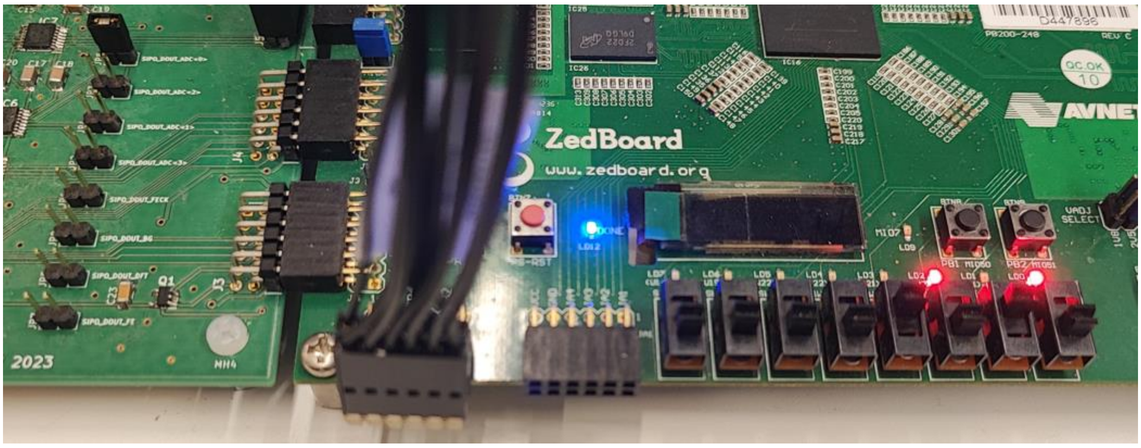


Figure 85. Reception of the data transmitted to the ADC0 SIPO (LSB)

The complete sequence received is “1011100101”. It can be checked with the simulation (Figure 67) that this is the expected data.

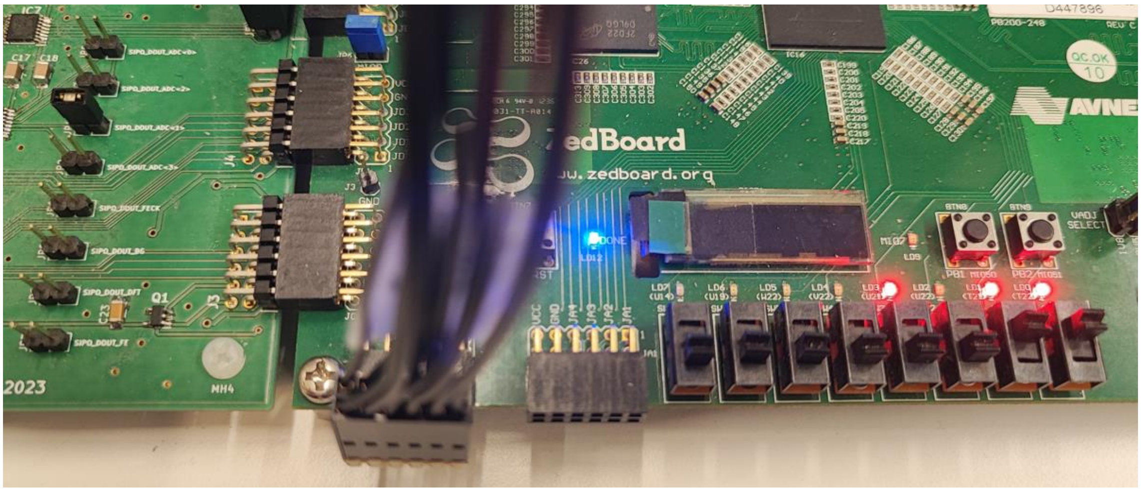


Figure 86. Reception of the data transmitted to the ADC1 SIPO (MSB)

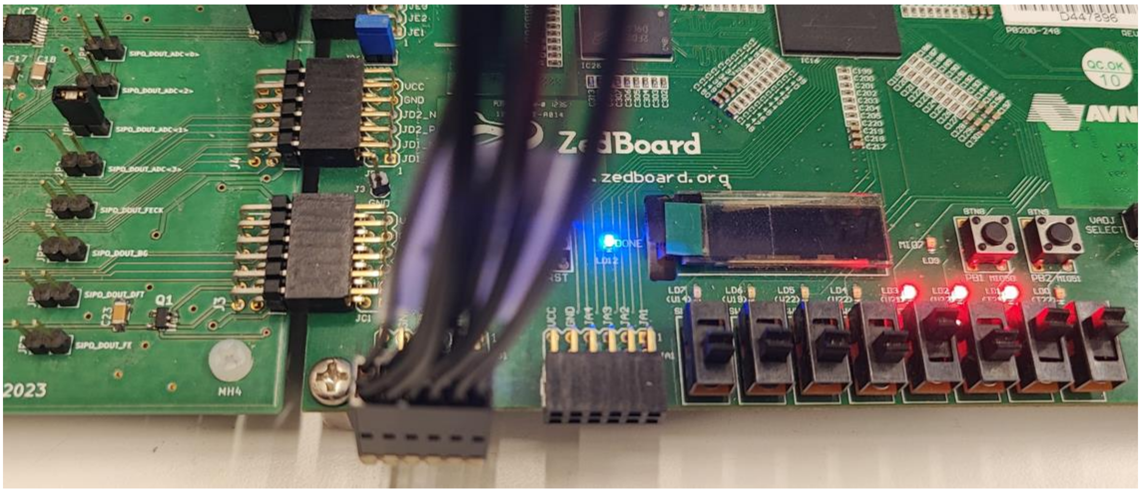


Figure 87. Reception of the data transmitted to the ADC1 SIPO (LSB)

The complete sequence received is “0101101110”. It can be checked with the simulation (Figure 66) that this is the expected data.

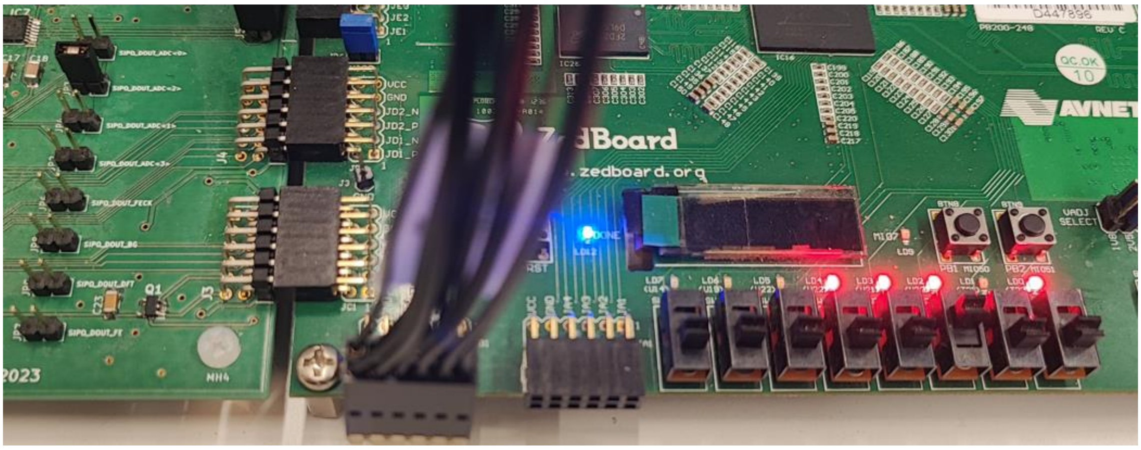


Figure 88. Reception of the data transmitted to the ADC2 SIPO (MSB)

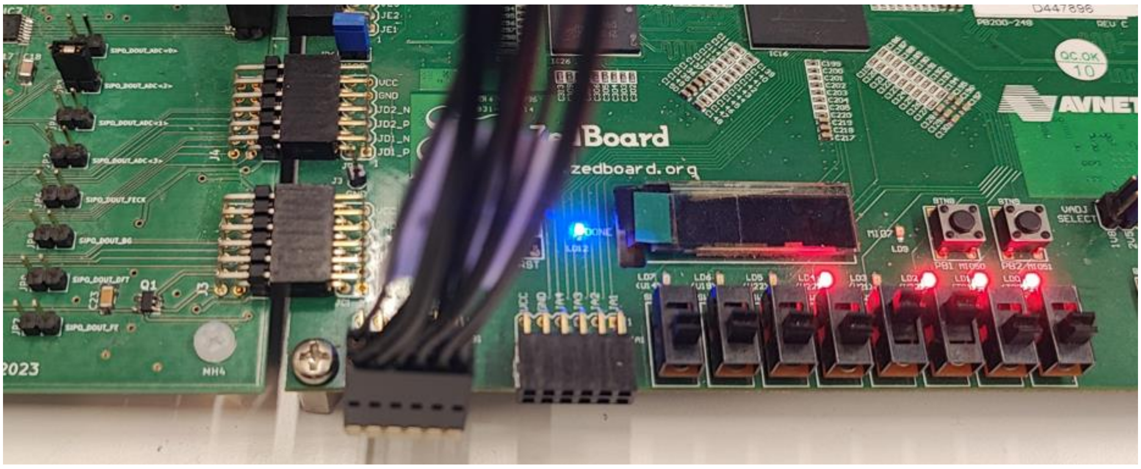


Figure 89. Reception of the data transmitted to the ADC2 SIPO (LSB)

The complete sequence received is “1110110111”. It can be checked with the simulation (Figure 65) that this is the expected data.

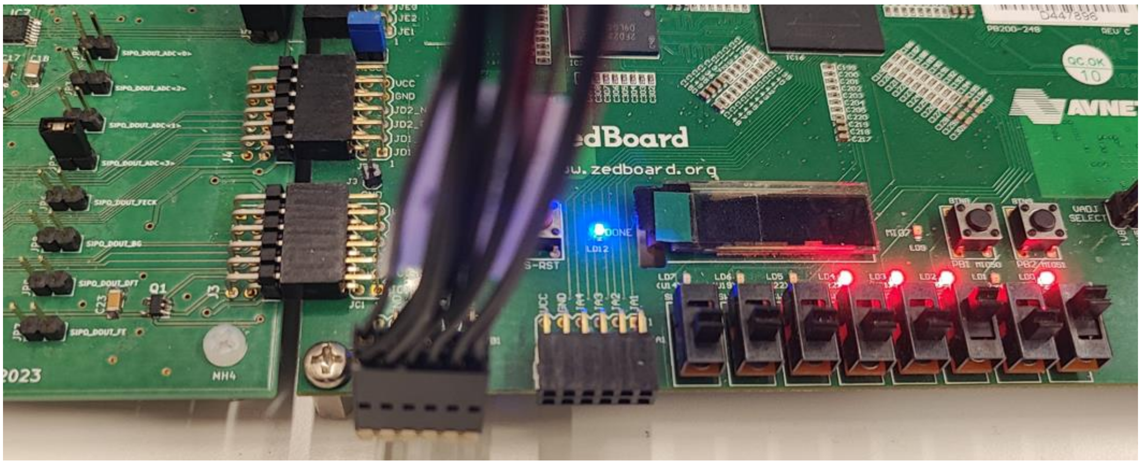


Figure 90. Reception of the data transmitted to the ADC3 SIPO (MSB)

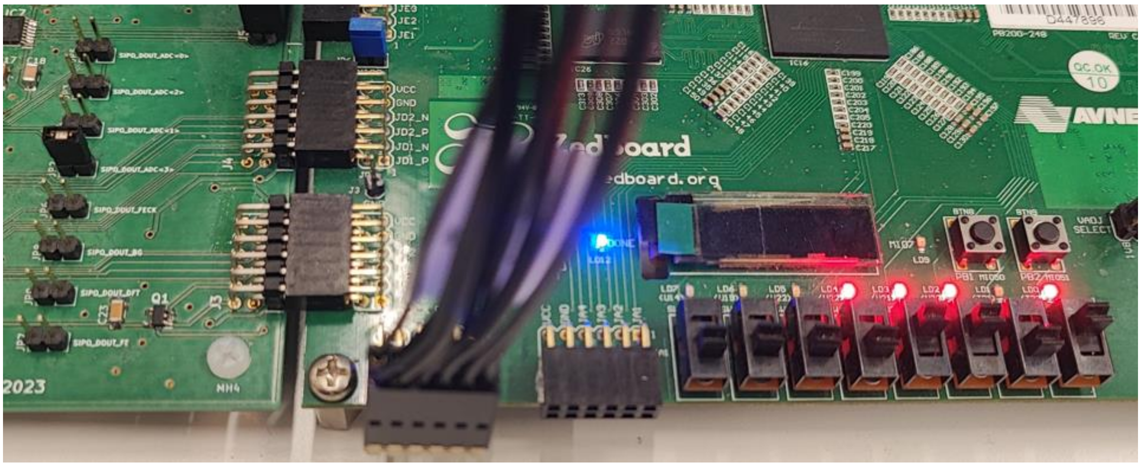


Figure 91. Reception of the data transmitted to the ADC3 SIPO (LSB)

The complete sequence received is “1110111101”. It can be checked with the simulation (Figure 64) that this is the expected data.

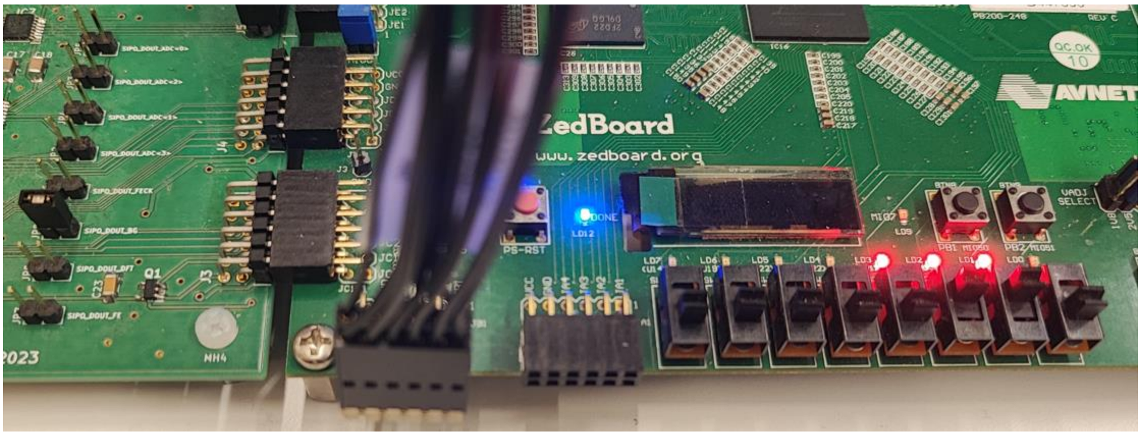


Figure 92. Reception of the data transmitted to the BG SIPO (MSB)

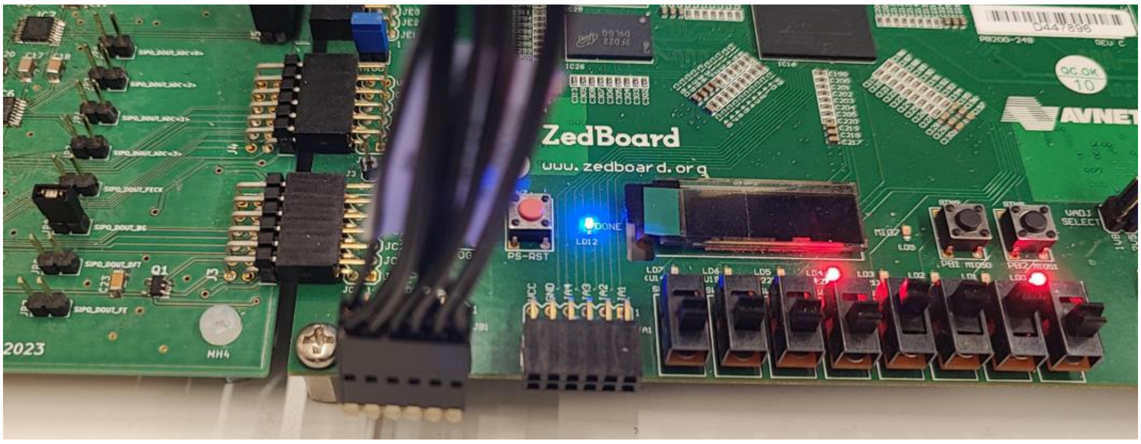


Figure 93. Reception of the data transmitted to the BG SIPO (LSB)

The complete sequence received is “0111010001”. It can be checked with the simulation (Figure 50) that this is the expected data.

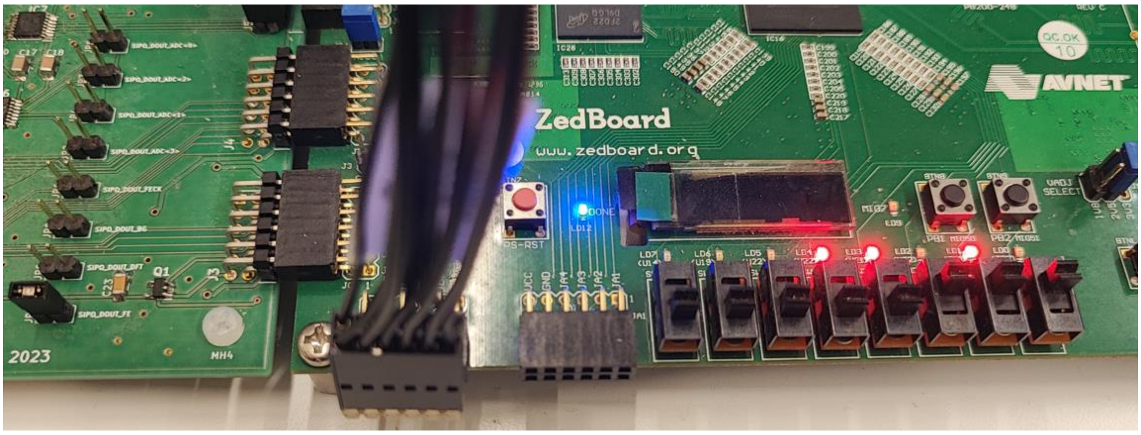


Figure 94. Reception of the data transmitted to the FE SIPOs (MSB)

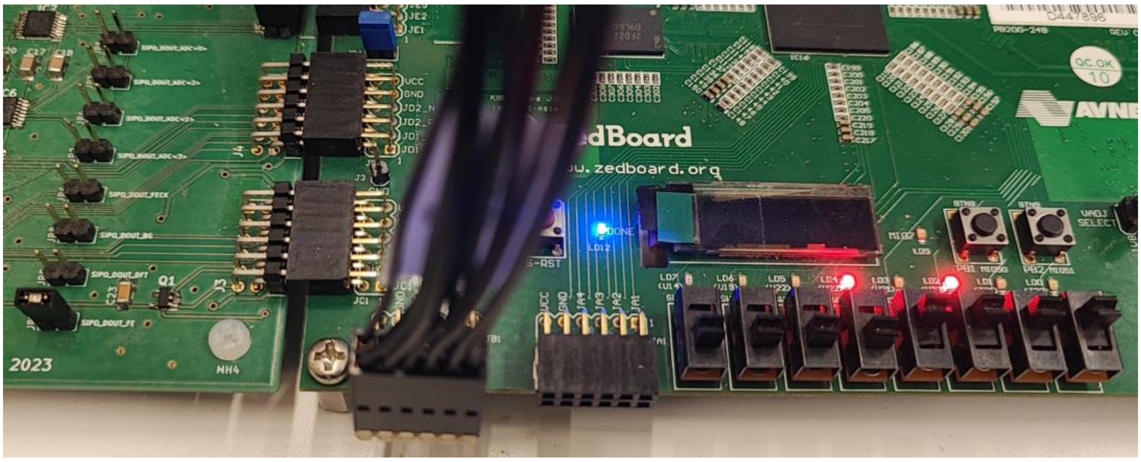


Figure 95. Reception of the data transmitted to the FE SIPOs (LSB)

The complete sequence received is “1101010100”. It can be compared with the simulation (Figure 57) that this is the expected data.

The reception of the data has been verified using both a minimum clock frequency of 10MHz and a maximum clock frequency of 100MHz. The frequency of the FPGA clock was set to 200MHz. As stated in the level shifters section (Section 3.5.1), the ideal maximum frequency to work would have been 125MHz, however, the level shifters cannot work at such a high frequency, thus, it was decided to reduce it to 100MHz. In both cases, 10MHz and 100MHz, the data was received successfully. However, the test bench to read the data sent by the SIPOs had to be adapted for high frequencies. It was necessary to adjust the test bench parameter to capture each received bit since the serial data received was delayed because of the physical elements of the whole system chip-PCB-FPGA. This adjustment had to be done when working at 100MHz but not necessary at 10MHz. This was expected as in 100MHz there are much less time to capture every bit, and this few time is highly affected by the delays contributed by the physical elements. Therefore, it is very important to be very precise and adjust the instant in which the data is captured. However, in 10MHz, despite adding also extra delays, the serial data changes much slower, then the time available to capture the data is not affected that much.

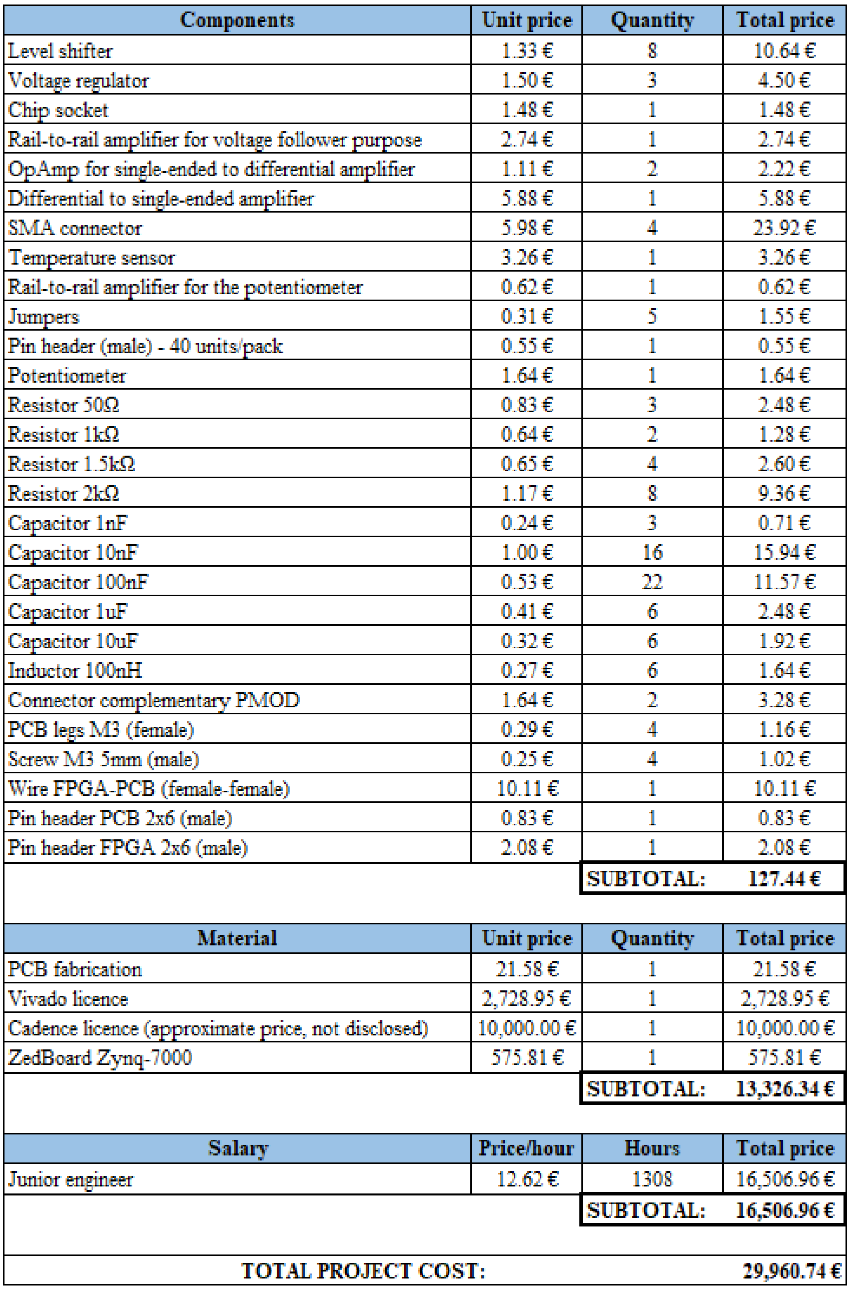
To summarize, at 10MHz the capture of each received bit in the FPGA was performed half a serial clock period after the positive edge of the clock occurred in which the SIPOs send each bit. This would be the ideal way to proceed: the SIPOs transmit the bits in the positive clock edge and the PISOs (FPGA) capture them in the negative clock edge. In contrast, at 100 MHz the capture of each bit was performed either one and a half periods or two serial clock periods after each positive serial clock edge occurred. Both capture instants were valid.

Due to the impossibility of accessing the input data of the PISO contained in the chip (these data come from the chip itself), it does not make sense to read the received data in the FPGA. That is, since the PISO input signals are unknown, they cannot be compared with the received data to see if they match. For this reason, the operation of the PISO has not been physically verified.

5. BUDGET

To develop this project, it was required to use several tools and make a PCB. In this section, it is detailed the cost of these elements as well as the cost of an engineer taking into account the amount of hours dedicated. In the budget presented below, it was only considered the cost to develop one PCB.

The project started on September until the first week of May. During the month of December there was a break in the project, thus, the total number of days dedicated was 218. The average daily hours invested were 6h, therefore, the total number of hours were 1,308h. Considering that the average cost per hour of a junior engineer in Spain is 12.62€, the total cost of hiring an engineer to develop the project is 16,506.96€.



Components	Unit price	Quantity	Total price
Level shifter	1.33 €	8	10.64 €
Voltage regulator	1.50 €	3	4.50 €
Chip socket	1.48 €	1	1.48 €
Rail-to-rail amplifier for voltage follower purpose	2.74 €	1	2.74 €
OpAmp for single-ended to differential amplifier	1.11 €	2	2.22 €
Differential to single-ended amplifier	5.88 €	1	5.88 €
SMA connector	5.98 €	4	23.92 €
Temperature sensor	3.26 €	1	3.26 €
Rail-to-rail amplifier for the potentiometer	0.62 €	1	0.62 €
Jumpers	0.31 €	5	1.55 €
Pin header (male) - 40 units/pack	0.55 €	1	0.55 €
Potentiometer	1.64 €	1	1.64 €
Resistor 50Ω	0.83 €	3	2.48 €
Resistor 1kΩ	0.64 €	2	1.28 €
Resistor 1.5kΩ	0.65 €	4	2.60 €
Resistor 2kΩ	1.17 €	8	9.36 €
Capacitor 1nF	0.24 €	3	0.71 €
Capacitor 10nF	1.00 €	16	15.94 €
Capacitor 100nF	0.53 €	22	11.57 €
Capacitor 1uF	0.41 €	6	2.48 €
Capacitor 10uF	0.32 €	6	1.92 €
Inductor 100nH	0.27 €	6	1.64 €
Connector complementary PMOD	1.64 €	2	3.28 €
PCB legs M3 (female)	0.29 €	4	1.16 €
Screw M3 5mm (male)	0.25 €	4	1.02 €
Wire FPGA-PCB (female-female)	10.11 €	1	10.11 €
Pin header PCB 2x6 (male)	0.83 €	1	0.83 €
Pin header FPGA 2x6 (male)	2.08 €	1	2.08 €
		SUBTOTAL:	**127.44 €**
Material	**Unit price**	**Quantity**	**Total price**
PCB fabrication	21.58 €	1	21.58 €
Vivado licence	2,728.95 €	1	2,728.95 €
Cadence licence (approximate price, not disclosed)	10,000.00 €	1	10,000.00 €
ZedBoard Zynq-7000	575.81 €	1	575.81 €
		SUBTOTAL:	**13,326.34 €**
Salary	**Price/hour**	**Hours**	**Total price**
Junior engineer	12.62 €	1308	16,506.96 €
		SUBTOTAL:	**16,506.96 €**
TOTAL PROJECT COST:			**29,960.74 €**

Table 3. Breakdown of expenses to carry out the project

6. CONCLUSIONS

To conclude, a review of the objectives set at the beginning of the project is carried out:

- ✓ Design of the necessary digital blocks to control and test the operation of the SIPOs and PISO blocks.
- ✓ Simulate the behaviour of the designed testing blocks.
- ✓ Simulate the design created together with the chip simulation to verify the correct behaviour of the SIPOs and PISO.
- ✓ Design the schematics to be implemented in the PCB.
- ✓ Selection of components based on technical, physical and economic criteria.
- ✓ PCB design, implementation and verification.
- ✓ Design of a test bench to generate the input stimuli and read the data received from the shift registers.
- ✓ Physical verification of the complete system: Chip-PCB-FPGA.

As seen throughout this thesis, all the objectives set have been met. The created digital designs work correctly, both theoretically and in practice. The designed PCB fulfills its task, however, two errors were detected during the verification process. The first, the impossibility of using the PMOD of the upper part of the FPGA, for this reason it was necessary to use an additional cable to connect the upper PMOD of the PCB to another PMOD of the FPGA. The second, the incorrect connection of an opamp pin of the potentiometer circuit. Both are not serious errors since they could be solved manually. However, they would be two defects to correct in a future version.

In the laboratory, it has been possible to verify the correct operation of all the SIPOs of the chip working at 10MHz and 100MHz. However, the operation of the PISO was only possible to demonstrate theoretically through simulations. This is due to the impossibility of stimulating the PISO from outside the chip with known input signals in order to compare them with the received data. Therefore, in future developments, laboratory verification of its operation could be a point to consider.

The created designs verified the operation of the shift registers by sending data to the chip and reading its responses (except in the case of the PISO). In the future, such designs could be used to control the entire operation of the chip and read the acceleration values provided by the ADC or read any other sensor or signal that is desired.

BIBLIOGRAPHY

[1] Shashidhara, H & Yellampalli, Siva & Goudanavar, Vasant. (2014). Board level JTAG/boundary scan test solution. 73-76. 10.1109/CIMCA.2014.7057760.

[2] IEEE Std 1149.1 - Standard Test Access Port and Boundary-Scan Architecture.1990

[3] Thakur, Ritu & Awasthi, Akanksha & Gupta, Ravish. (2016). A Review Paper based on Built in Self Test. International Journal of Engineering Research and. V5. 10.17577/IJERTV5IS020314.

[4] EESemi (n.d.). Built-In Self-Test (BIST) [Online] Available: https://www.eesemi.com/bist.htm

[5] S. Narayanan and A. Das, "An efficient scheme to diagnose scan chains," Proceedings International Test Conference 1997, Washington, DC, USA, 1997, pp. 704-713, doi: 10.1109/TEST.1997.639683.

[6] VLSI Universe. (2013). Scan Chains - The Backbone of DFT. VLSI Universe. [Online] Available: https://vlsiuniverse.blogspot.com/2013/07/scan-chains-backbone-of-dft.html

[7] Ruifeng Guo and S. Venkataraman, "A technique for fault diagnosis of defects in scan chains," Proceedings International Test Conference 2001 (Cat. No.01CH37260), Baltimore, MD, USA, 2001, pp. 268-277, doi: 10.1109/TEST.2001.966642.

[8] Wang, Fei & Hu, Yu & Li, Huawei & Li, Xiao-Wei. (2008). A design- for-diagnosis technique for diagnosing both scan chain faults and combinational circuit faults. 571-576. 10.1109/ASPDAC.2008.4484017.

[9] Generatec Tecnologias. (n.d.). Arquitectura FPGA [Online] Available: https://www.generatecnologias.es/arquitectura_fpga.html

[10] Digilent Inc. (2012, Aug. 1). ZedBoard Hardware User Guide [Technical report]. Digilent Inc. Available: https://digilent.com/reference/_media/zedboard:zedboard_ug.pdf

[11] Texas Instruments. (2021). SN74AVC4T774 4-Bit Dual-Supply Bus Switch [Technical report]. Texas Instruments. Available: https://www.ti.com/lit/ds/symlink/sn74avc4t774.pdf

GLOSSARY

ADC	Analog-to-Digital Converter
ATPG	Automatic Test Pattern Generation
BG	Bandgap
BIST	Built-in self-test
CMOS	Complementary Metal-Oxide Semiconductor
CUT	Circuit Under Test
DC	Direct Current
DFT	Design For Test
FE	Front-End
FECK	Front-End Clock
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
IEEE	Institute of Electrical and Electronics Engineers
LED	Light-Emitting Diode
LSB	Least Significant Bit
MEMS	Micro Electronic Mechanical System
MSB	Most Significant Bit
PCB	Printed Circuit Board
PISO	Parallel-In Serial-Out
PMOD	Peripheral Module
ROM	Read-Only Memory
SIPO	Serial-In Parallel-Out
TAP	Test Access Port
TCK	Test-Clock
TDI	Test-Data-In
TDO	Test-Data-Out
TMS	Test-Mode-Select
TRSTn	Test-Reset