Design and Implementation of a 2-Channel High Precision and High Speed Digitizing System

Master's Thesis

By

Albert Llimós Muntal

Supervisors:

Liang Liu, Andreas Johansson, Khadga Kharki and Fredrik Edman

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

LUND UNIVERSITY

2016

The Department of Electrical and Information Technology
Lund University
Box 118, S-221 00 LUND
SWEDEN

Abstract

Recently, new spectroscopic techniques have been developed such as photocurrent detected two-dimensional spectroscopy [1], which measure the response from a sample that is excited by a high intensity laser pulses. The responses are linear and nonlinear signals that have very different amplitudes; the linear signal can be from 10^3 to 10^5 times larger than the nonlinear signal. These two signals have to be digitized and transferred to a computer to be able to isolate the photocurrent nonlinear signal. This requires analog to digital converters (ADC) with very high dynamic range.

Some ADCs in the market meet these requirements but they can only sample one channel and this application requires two. Using two separate converters is not optimal due to phase delays in the sampling. Therefore, in this project, a two-channel high-speed and high-precision digitizing system is built. It is a complete system that includes both analog and digital hardware as well as software. It consists of two ADC boards that are controlled using a Field Programmable Gate Array (FPGA) included in an FPGA board. In order to communicate this modules, an interfacing board has been designed. The code elaborated in this project for the FPGA programming has been written in VSIC Hardware Description Language (VHDL). In order to control the data communication with the PC, a software interface application is developed.

The digitizing system built can function at a sample rate up to 4MS/s and has a resolution of 23 bits. This system only uses one programmable module, which reduces its cost drastically compared to two separate one channel digitizing systems.

The tests performed with the system demonstrate that even with just one programmable block, a high speed digitization and data transfer can be achieved. The data acquisition system is able to sample one million samples per channel per acquisition event in 780 ms. Moreover, it can sample up to 4MS per channel per acquisition event. It is worth mentioning that the phase delay between the two digitized signals is low, averaging 0,383°, with a standard deviation of 0,247°, which is below the maximum 1° allowed for this application.

This high-speed and high-resolution system is not only suited for photocurrent spectroscopic applications but it can also result on the improvement of other systems like high end sound cards or ultrasound imaging.

Acknowledgments

First of all I would like to thank my supervisors Liang Liu and Andreas Johansson for their advice and help through this Master Thesis. They have been helping me overcome the obstacles that I have faced in this project and this thesis would not have been possible without them.

I would also like to thank Khadga Karki for giving me the opportunity to do this project as well as for his advice. His suggestions have helped me improve the final design of the system.

I would like to thank Lund University for giving me the opportunity to do my exchange in Sweden and complete my Master Degree program.

I am also grateful to my brother, Pere Llimós, who has helped and supported me during this work.

Finally, I want to thank my home university from Barcelona (UPC) for all the knowledge and experience acquired during the Bachelor and Master program in Industrial Engineering.

Contents

List of figures

List of tables

List of acronyms

ADC	Analog to Digital Converter
USB	Universal Serial Bus
FPGA	Field Programmable Gate Array
PROM	Programmable Read-Only Memory
PLL	Phase-Locked Loop
LVDS	Low-Voltage Differential Signaling
CMOS	Complementary Metal Oxide Semiconductor
RAM	Random-Access Memory
FIFO	First In First Out
MUX	Multiplexer
MSB	Most Significant Bit
LSB	Least Significant Bit
PCB	Printed Circuit Board

Popular Science Summary

In the Chemical Physics field, new techniques have been developed where the photocurrent generated has to be measured when a source of instantaneous energy is provided (pulsed laser). Photodiode's response to this stimulus has a linear component as well as a nonlinear one depending on the energy absorption characteristics. For those techniques it is essential to process and study the nonlinear response. For this purpose, it is necessary to isolate the two signals and transfer them to a computer. This requires the digitization of the signals, which means describing the signals generating a series of number (bits) that describe a discrete set of its samples. This is done using an analog to digital converter (ADC). Given that the linear signal is a lot larger than the nonlinear, ADCs with very high dynamic range are needed, which is the ratio between the maximum and the minimum input signal that can be digitized. The digitized data has to be stored to avoid data loss and then transferred to the computer.

In the current market there are no two-channel digitizing systems that meet the requirements for this application, either its sampling rate is not high enough or its dynamic range does not meet the specifications. Therefore, the goal of this project is to build a two-channel high-speed and high-precision digitizing system suited for this application. In order to do that, two ADC boards are selected to implement the two-channel digitizer. For the storage and transfer of data a logic programmable module as well as a memory is needed. To fulfill this requirement, another board containing a field programmable gate array (FPGA) and a memory is included in the system. This board also contains a USB microcontroller that enables the data transferring from the system to the computer via USB. Finally, connecting these boards is done through an interfacing board, which is designed in this project.

The logic in the FPGA is programmed using VHDL code. The data shifted from the ADC is in a serial bus and, since its storage in the memory has to be done through a parallel bus, the first step is to deserialize it. Then, another block is implemented to manage the reading and writing from the memory since it cannot be done simultaneously. Finally, a software interface is developed based on an application programming interface provided by Opal Kelly, the manufacturer of the FPGA board. This interface contains a set of protocols and tools to build an application that enables the communication and data transfer between the system and the computer.

The digitizing system built in this work meets all the speed and precision requirements and can sample the signals up to the frequency that is needed. The storage, management and transfer of data is also done efficiently since the time required for the system to acquire all the data is significantly lower than other systems. Moreover, the phase delay between the two digitized signals has successfully been minimized as well as its variance, which was a critical aspect for this work. Finally, the cost of this system is lower than two separate one-channel digitizing systems.

The implementation of this digitizing system is not only limited to spectroscopic techniques but it can also be used in many other applications that require a high-speed and high-precision digitizing system. This can include, for example, sound cards systems or even ultrasound applications. Moreover, this project can be a good base for improvements as its performance is limited by the memory capacity as well as the transfer rate of the USB bus, modules that can potentially be changed. Finally, its proven functionality and operation method can be a base for building an integrated one-board system containing all the modules described in this project. This also opens the possibility for increasing the number of channels of the system.

1. Introduction

1.1. Background

Spectroscopy is the study of the interaction between electromagnetic radiation and matter. It is referred to the measurement of the radiation intensity as a function of wavelength.

The photocurrent signal from devices, such as photodiodes, is usually linearly proportional to the intensity of the light on the device. However, nonlinear effects can also contribute to the photocurrent at very high excitation intensity. As shown in Fig.1, high excitation intensities can be produced from commonly available pulsed lasers, as in the fluorescence detected wave-packet interferometry [2][3] and the two-dimensional electronic coherence spectroscopy [4].

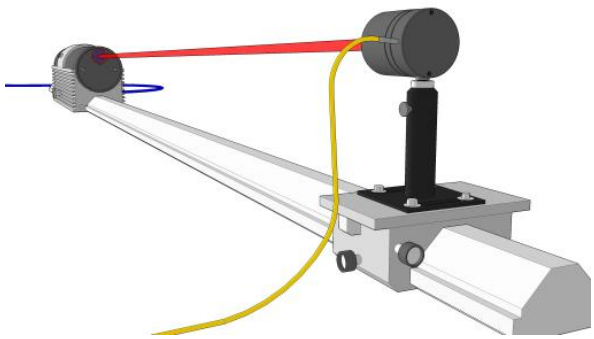


Fig. 1. Short laser pulses provide energy to a photodiode

A 10 fs short laser pulse, whose energy is 1 nJ, produces an instantaneous intensity of $4x10^{11}$ W/cm^2 when focused to a spot with a diameter of 5 µm. Some of the nonlinear responses, such as two and three photon absorption in photodiodes are routinely used to measure the pulse duration. In these measurements, one uses a photodiode whose band-gap energy is larger than the energy of the incident photon. As an example, a large band-gap semiconductor such as GaP, which has a band gap of about 2.25 eV at room temperature, does not absorb photons at 800 nm (energy of a single photon at this wavelength is about 1.55 eV). However, under

intense excitation the semiconductor can absorb two photons simultaneously and generate photocurrent as shown in Fig.2.

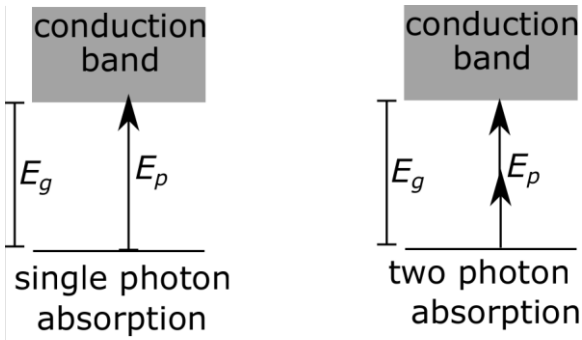


Fig. 2. Photon absorption by the conduction band

The photocurrent generated by the absorption of two photons increases quadratically with the excitation intensity; hence it is a nonlinear effect. The nonlinear effects can also contribute to the photocurrent from photodiodes with low-band gap. Commonly available Si photodiodes (band gap of 1.1 eV at the room temperature) can absorb light by the process of single photon (linear) as well as two photon (nonlinear) absorption. However, the photocurrent due to the linear absorption of the light is much larger than the photocurrent due to the nonlinear absorption. Therefore, the nonlinear effects cannot be easily isolated or used in such photodiodes. Recent studies have shown that the technique of intensity modulation of the light pulses can be used to separate the nonlinear contribution to the photocurrent from the linear contribution [5].

If the intensity of the light is modulated at frequency f, then the photocurrent from a photodiode also modulates at the same frequency. This is true only when the absorption of the photons is a linear process. If the absorption of the photons also occurs via two photon absorption then a photocurrent at 2f can be observed. Similarly, absorption of photons by higher order nonlinear processes produces photocurrent at higher harmonics of f. The ability to isolate the nonlinear photocurrent signal from the linear signal marks an important step in the understanding of how the nonlinear processes affect the functionality of photoactive devices [6]. The nonlinear signals allow the investigation of how the ultrafast processes such as carrier dynamics and relaxation processes contribute to the generation of the external photocurrent. Moreover, isolation of nonlinear signals also opens up new possibilities in the use of nonlinear interactions in device characterization, design of optoelectronic logic gates, etc.

The current setups used in the two-dimensional spectroscopy, use a number of analog filters and lock-in amplifiers to isolate certain nonlinear signals [7], which makes the setup more complex and the data acquisition cumbersome. Moreover, if one wants to measure many linear and nonlinear signals, a large number of lock-in amplifiers have to be used, which substantially increases the cost and complexity of the measurement system. Also, using a digitizer provides more stability since the system contains a digital part. Furthermore, the size of the system is reduced. Even comparing it to a flexible and compact acquisition card developed for the sensitive gas spectroscopic techniques [8], the size is significantly reduced when using a digitizer instead of complex lock-in filter system. Therefore, it can be placed closer to the detector reducing the noised caused by interferences from other electronic signals.

A digitizer is used to measure both the amplitudes and the phases of the linear signal as well as the different nonlinear signals with respect to the reference. The comparative studies of the amplitudes and the phases of the different nonlinear signals at the different modulation frequencies enable the investigation of the effects of the nonlinear light-matter interactions on the functioning of the photoactive devices. The digitizer is also used to measure different nonlinear signals in advanced spectroscopic measurements, such as fluorescence and photocurrent detected two-dimensional spectroscopy. However, the modulation technique is still novel and important technical developments are necessary to increase its wider use and applications. One of the limitations of the current measurements is that the linear signal has to be electronically filtered out in order to measure the weak nonlinear signals. Usually the linear signals due to single photon absorption is about 10^3 to 10^5 times (three to five orders of magnitude) larger than the signal due to two photon absorption. Similarly, the signals due to higher order nonlinearities are progressively weaker. A 14 bit digitizer could have a dynamic range of about $16x10^3$, which is good enough to separate linear and two-photon photocurrent from the total signal in some cases. However, such digitizers do not have enough dynamic range to measure higher order signals together with the linear signal.

To be able to achieve that purpose, a two channel digitizer has to be built with a resolution of at least 23 bits, which can provide a maximum dynamic range of 140 dB. This is important to be able to digitize two signals with very different amplitudes. Moreover, its bandwidth needs to be greater than 1 MHz as for this application it is necessary to sample signals up to that frequency. This application also requires a high sample rate of 4MS/s. All of these requirements are added to the fact that the system has to be able to sample two channels simultaneously. This is because in the

experiment, one of the channels is used to digitize reference while the other one is used to digitize the signal.

1.2. Challenge of the design

Firstly, the system has a high-resolution and high-speed demands. Moreover, it needs to have high sampling data capacity (more than 1Ms), which leads to a very high amount of data to be collected and transferred. Therefore, the system requires a high data throughput.

Another challenge for this design is to have low latency. To achieve it, all the data managing, storage and transfer has to be done efficiently.

There are not any two-channel digitizing systems available in the market that meet the requirements, they are either low resolution or the sampling speed is not high enough. Consequently, if a high speed high precision two channel system is needed it has to be custom built. In this work, the target is to build a high speed high precision two channel system, and one of the challenges is to minimize the phase delay that appears when using two separate systems to sample and keep it as constant as possible.

Furthermore, the main cost of the full system is the programmable logic module, which controls each channel. In a digitizing system this module can be up to 75% of the total cost. Thus, a fully integrated two channel system that only requires one programmable logic module can lead to a significant cost reduction. For this reason, the last challenge of this project is to build a system with a low cost, at least lower than two separate digitizing systems.

2. System requirements and component selection

In this chapter, section 2.1 presents the system requirements that are that the design has to meet. Secondly, section 2.2 discusses the reason for the selection of the chosen components for the design.

2.1. System requirements

It is required to build a digitizing system with a sample rate up to 4MS/S and that has a bandwidth of at least 1MHz. The high dynamic range needed for the application already mentioned entails a minimum resolution of 23 bits. One logic programmable module is also required to be able to control the ADCs and manage and transfer all the data to the computer.

It is also important that this system's price is lower than buying two separate systems. As explained above, using just one logic module already decreases the price, so it is relevant that it is not increased when interfacing two ADC channels. Consequently, the price of the system has to be below the price of two separate digitizing systems.

Using one logic module to control both ADC helps improve phase alignment between the two sampled signals. Acknowledging that the data collected from both channels has to be processed and compared, it is essential to sample those signals simultaneously, reducing as much as possible any phase delays. It is required to have a phase delay of less than 1° and its standard deviation has to also be below 1°.

Since the aim is to build a system that transfers the data to a computer, it is necessary to calculate the minimum transfer speed rate needed. The ADC samples at a maximum rate of 4MS/S and each sample has to have at least 23 bits:

$$Thoughput = 2 \cdot \left(4\ MSPS \cdot \frac{23\ Mbit}{1\ MS}\right) = 184\ Mbit/s \qquad (1)$$

Therefore, as shown in (1) the system to transfer the data to the computer must have a transfer rate of at least 184Mbit/s. Transferring the data directly from the converters to the computer can only be done if all the modules included in the system have a high enough transfer rate. Otherwise, data overrun can take place, which has to be avoided.

Finally, the latency of the system is also a critical aspect and the aim for this system is to be able to acquire 1MS from each channel in less than 1 second for every acquisition event.

2.2. Component selection

Firstly, it is essential to select a converter that meets the requirements. The resolution is also one of the key aspects, as it has to be over 23 bits. Therefore, a search for ADCs with this resolution is done as shown in Table1.

Part Number	Resolution (Bits)	Sample Rate (max)	#Input Channels	Input Range (V)	Interface
ADS1675	24	4MS/s	1	3.5	Serial
AD7760	24	2,5 MS/s	1	3,5	Parallel
ADS1672	24	625kS/s	1	3.25	Serial
ADS127L01	24	512kS/s	1	3	Serial

Table 1. Analog to digital converters

It can be noticed that the options are very limited. Moreover, the second most important aspect is a high sampling rate, which is required to be at least 4 MS/s. Thus, the only converter than meets both the requirements is the ADS1675.

At this point, one option is to build a board for the converter to obtain the conversion system. However, as one-channel digitizing boards are available in the market, it seemed convenient to look for an already existing board before building one around the converter. When looking for a board that could meet the requirements, the reference board Texas Instruments ADS1675REF stood out, which is a reference board that uses exactly the ADC mentioned in Table 1.

The ADS1675REF is a reference board made of two boards, an ADC board and an FPGA board, which can be seen in Fig. 3. The first one includes a high-speed and high-precision analog-to-digital converter (ADS1675) with an analog input driver circuit. It can operate at speeds up to 4MS/S, speed required for the applications in the two-dimensional spectroscopy techniques. Nevertheless, it can also operate at lower speeds (either 1MS/S, 500kS/s, 250 kS/s or 125 kS/s). This offers the possibility to sample at different rates, which may be interesting depending on its application. The accuracy of this ADC is 24-bit. However, it is important to notice that when operating at high speed modes, the 24th bit is held low. Therefore, it has a resolution of 23 bits at high speed modes (4MS/S and 1MS/S). This is not a problem since the resolution requirement for the system is 23 bits. Also, it can work with two different voltage levels, LVDS and CMOS. In LVDS mode, some control signals will be differential, which will be indicated in section 3.3.3.

This converter also contains a dual path filter that allows the user to select between two post-processing filters. On one hand, the low-latency filter provides a short settling time which is ideal for applications with large instantaneous changes. However, its bandwidth is 355 kHz, which is lower than the required value (1MHz) and therefore it is not useful for this application. However, the wide-bandwidth path provides an ideal frequency response for AC measurements with a bandwidth of 1.7MHz, which is greater than the one required (1 MHz). This means that using this filter the needed bandwidth can be reached.

The characteristics of the ADC board meet the requirements for the digitizing system.

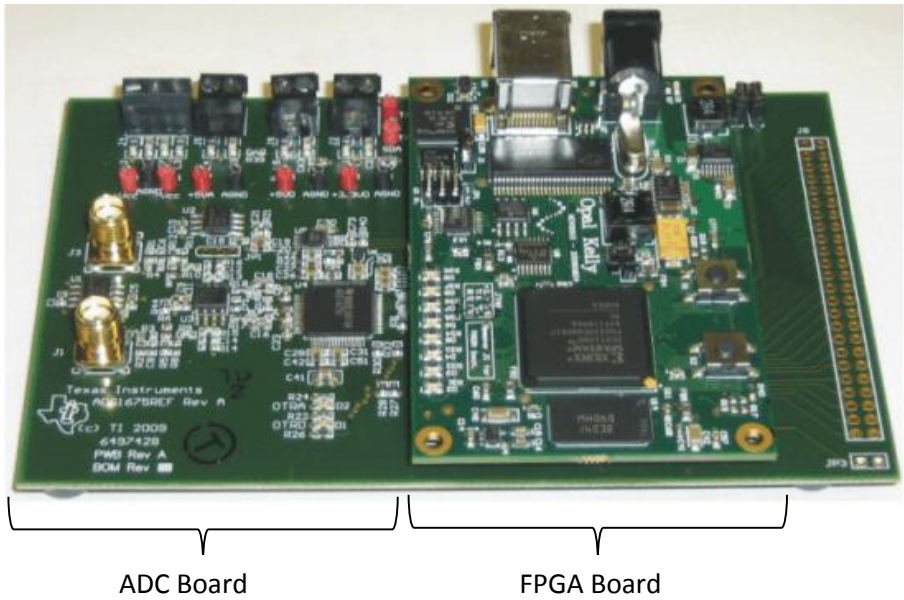


Fig. 3. Reference board ADS1675REF

The outputs of the ADC board as well as the control signals are connected through two expansion connectors to an FPGA board (XEM3010), which is placed on top of the ADC board as shown in Fig. 3. It is interesting to analyze if this board can also be used for this project.

The XEM3010 board is an FPGA board featuring a Spartan-3 FPGA as well an SDRAM, a PROM and a USB 2.0 interface. It communicates with the ADC board through two 80-pin expansion connectors. This FPGA can be used as the logic module needed for the project to manage and transfer all the data.

The XEM3010 board also has a USB micro CY68013A implemented. It is a USB 2.0 which means that its transfer rate is up to 480 Mbit/s. However, the real data transfer to a computer that can be ensured to have regardless of the equipment used is around 240 Mbit/s. In this case, it is suited for this project, since the transfer rate required is 184 Mbit/s. However, as it will be explained in section 3.1.3, the software application that enables the communication between the FPGA and the PC does not have a high enough transfer rate. Therefore, a memory is needed to be able to store all the data and avoid data overrun from happening.

The memory in the XEM3010 can be used to store the collected data and transfer it to the computer when needed. This way, whenever data is stalled or the computer cannot handle more data at that moment, the memory can store it until that is no longer an issue. For the spectroscopic applications, a large amount of data has to be sampled and saved in a file for further processing, at least 1 MS per channel per data acquisition event. Given that the required resolution is at least 23 bits for both channels, the storage capability that is needed is 46Mbits minimum. Since the memory in the board is a 32MB 16-bit wide SDRAM, its depth is greater than the required, which makes it suitable for the system.

Finally, the board also includes a Cypress CY22393 multi-output PLL. It is a triple PLL-clock generator that can provide up to five clocks. Three of the clocks are connected to the input pins of the FPGA while the other two are generated and sent through the expansion ports to the ADC board. Consequently, the same clock signal that controls the ADC can externally be connected to a second ADC through the corresponding expansion pin, which can significantly reduce the skew between the two channels.

The XEM3010 board provides access to over 110 I/O pins out of the total 320 pin existing in the Spartan-3 FPGA. It has access to the SDRAM memory through 39 pins and its communication with the on-board USB microcontroller is done through 26 other pins. 8 Pins are connected to the LEDS and two more to buttons on the board, while the communication with the ADC board takes 16. This leaves many unused FPGA pins that are connected to free pins in the expansion connector, which can be used to communicate with a second ADC channel.

This FPGA board is used in this project since it meets all the requirements and allows the implementation of a second channel without the need of another logic block.

3. System design

This chapter presents the design of the system solution for this work. Section 3.1 introduces the system-level diagram of the system and the interface between the different modules. Secondly, in section 3.2, the logic inside the FPGA is presented and the most important modules are described in dept. Finally, the interface board used to implement the second channel is detailed in section 3.3.

3.1. System-level diagram

The block diagram of the digitizing system containing all the modules is presented in Fig.4. It contains 5 different elements; two ADC boards, an interface board, an FPGA board and lastly the PC software. The connection between the first ADC and the FPGA can be done using the expansion connectors. However, the connection between the second ADC board and the FPGA has to be done externally through an interfacing board, which will be presented in section 3.3.

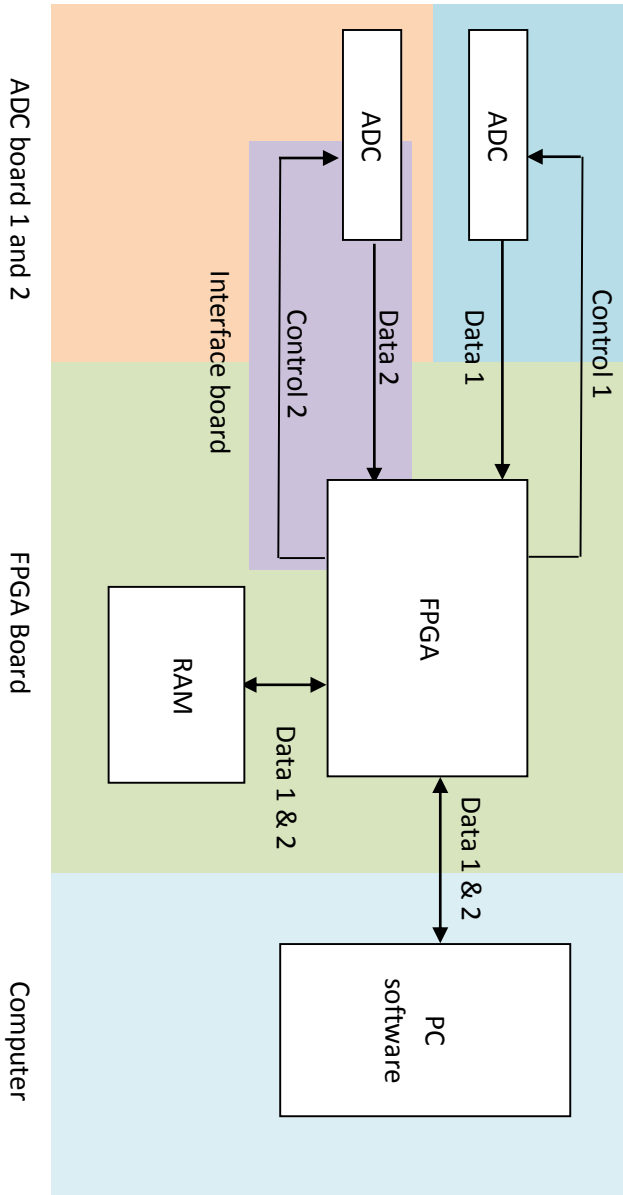


Fig. 4. System level diagram

3.1.1. Second ADC channel

Given that two channels are required, an additional ADC has to be included in the system. The ADS1675 converter's datasheet [9] presents the possibility of synchronizing multiple converters to obtain a multichannel system. To achieve it, it is necessary to use the same clock and the same start signal for both converters as shown in Fig.5.

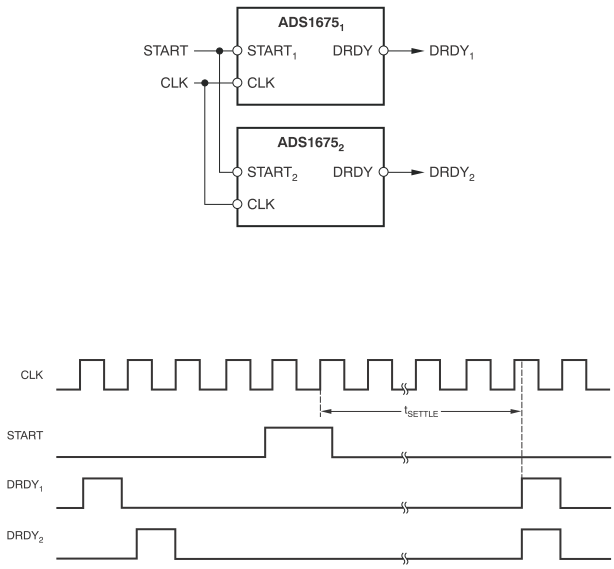


Fig. 5. Synchronization of multiple converters [9]

The data is digitized when the ready signals (DRDY) are triggered after the start pulse (START). As shown in Fig. 5. both the ready signals (DRDY) are triggered at the same time by the converters, which means that both conversions are done simultaneously. This configuration minimizes the phase delay between samples taken from the channels which is one of the main requirements for the system.

The most efficient way to add another converter is to include another ADS1675 ADC board in the system, exactly as the one used for the first channel. The converters are controlled by the same FPGA. Thus, a single

FPGA handles all the data, minimizing the amount of programmable logic used in the system. This is possible due to the fact that one of the ADS1675REF's board expansion connector disposes of a series of unused pins that are routed to the FPGA. These pins can be used to transfer the data from the second ADC board to the FPGA. Nevertheless, an interfacing board has to be built in order to connect the three boards together, not only for physical reasons but also to accomplish the proper pin connections to communicate the FPGA with both converters.

3.1.2. ADC control signals

The ADC board's function is to sample the input analog signal and transfer it to the FPGA. The interface board enables the connection between the second ADC board and the FPGA, while the first one is directly connected to it. This board's conversion characteristics and performance are controlled by the FPGA through the control signals.

Each ADC converter is entirely controlled by pins since there are not any registers. The pins controlling the conversion characteristics are shown in Table 1.

Signal	Meaning	Type
DR0	Sample rate (0)	INPUT
DR1	Sample rate (1)	INPUT
DR2	Sample rate (2)	INPUT
FPATH	ADC filter selection	INPUT
CS	Chip selector	INPUT
LL_Config	Filter configuration	INPUT
LVDS	LVDS or CMOS	INPUT
CLK_SEL	Internal/external clock	INPUT
PDWN	Powerdown	INPUT
DRDY_P	Data ready (P)	OUTPUT

DRDY_N	Data ready (N)	OUTPUT
DOUT	Sampled data (P)	OUTPUT
/DOUT	Sampled data (N)	OUTPUT
SCLK	Clock generated by the PLL in the ADC (for the output data)	OUTPUT
/SCLK	Clock generated by the PLL in the ADC (for the output data)	OUTPUT
START	Start signal	INPUT
FPGA_CLK	Clock signal	INPUT

Table. 1. Control signals for the ADC

For the control and readout of the ADS1675, a certain sequence of events has to take place.

First of all, on the power up or reset of the device, only the clock signal (FPGA_CLK) is required. This clock is generated by the PLL of the FPGA board (not from any FPGA pin). When sampling at high speeds, another PLL located inside the ADC converter generates a clock (SCLK) that is 3 times the frequency of the input clock. This PLL needs a settling time ($t_{LPLLSTL}$) in order for the SCLK to start running. This time is required to close the loop. Once the PLL settles, the data ready signal (DRDY) gives out a pulse to indicate that the lock is complete (as seen in Fig.6). At the same time, the desired values for the configuration pins must be set.

Later, the device expects a high value on the start signal in order to start the conversion. Once this happens, after the settling time of the filter (t_{settle}), the conversion begins and the DRDY signal gives out a pulse to confirm it.

This sequence of events (PLL loop closing, filter settling and DRDY pulse) must also take place when the CLK frequency is changed or whenever the mode is changed (from high-speed to low-speed or vice versa).

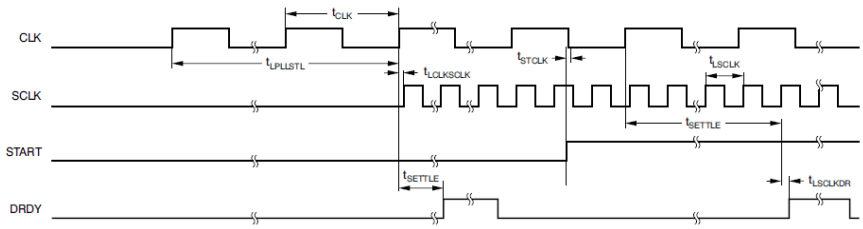


Fig. 6. ADC control [9]

As shown in Fig.7, after each conversion the digitized data is shifted (serial) through the DOUT pin, starting with the MSB bit and ending 24 t_{SCLK} later with the LSB.

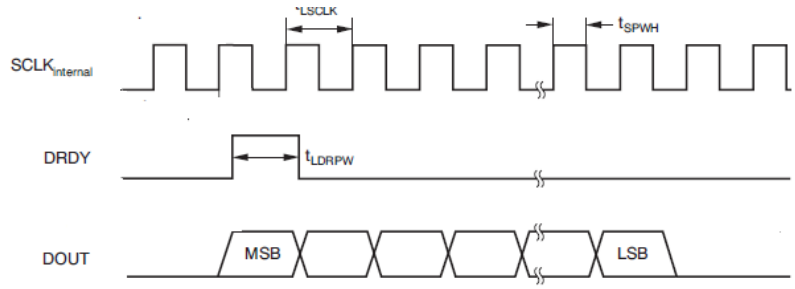


Fig. 7. ADC data output [9]

A single conversion is achieved by giving out a pulse on the start pin after the settling time. If the user wants to perform a continuous conversion, the start signal has to be held high (instead of just a pulse) and the board continues digitizing the data, shifting the most significant bit (MSB) of the following data just after the least significant bit (LSB) of the prior one.

Finally, the signals DR0, DR1 and DR2 control the sampling rate as shown in Table 2. The converter can work with different voltage level configurations: low-voltage differential signaling (LVDS) and CMOS. This is controlled by the LVDS signal while the filter used by the ADC is set by the FPATH signal. If the filter used is the low latency filter, its configuration can be chosen between single cycle settling mode and a fast response mode. This can be seen in Table 3.

DR2	DR1	DR0	DRATE (kSPS)
0	0	0	125
0	0	1	250
0	1	0	500
0	1	1	1000
1	0	0	2000
1	0	1	4000

Table. 2. ADC sampling rate configuration

LVDS	Voltage levels	Config [3]	Filter Path	Config [4]	Low lat. Config.
0	LVDS	0	Wide-Bandwidth	0	Single cycle settling
1	CMOS	1	Low Latency	1	Fast response

Table. 3. ADC voltage level, filter selection and filter configuration

3.1.3. Software

Opal Kelly's Front Panel software provides an end-to-end communication between a software application and the FPGA. It is designed to provide controllability and observability for the FPGA designs.

The host interface consists of different modules that allow Front Panel to control and observe the design. It contains logic that enables the microcontroller on the device to communicate with each of the endpoints (Wires, Triggers and Pipes). An endpoint is a module that works as an external pin of the FPGA, data can be transferred through them into the

computer. This structural relationship between the different endpoints is displayed in Figure 8.

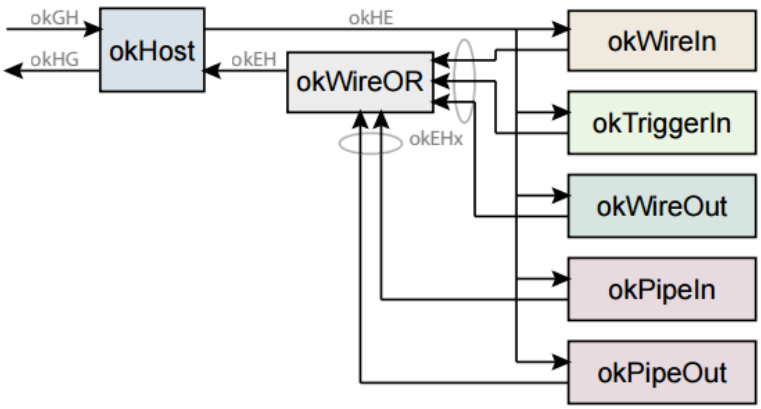


Fig. 8. The host interface [10]

Each of these endpoints has a different functionality, which is explained below.

- okHost: Enables the signal connection between the FPGA pins and the USB microcontroller.
- Wires (In/Out): Single 16-bit input/output bus (respectively) connected to the design and act as asynchronous connections. All the wires are updated simultaneously on the host interface clock.
- Triggers (In/Out): Single 16-bit input/output bus used to produce a single-cycle trigger pulse on any of the bits of the bus which is synchronized to the clock signal ep_clk.
- Pipe (In/Out): This module provides a way to transfer multi-byte data synchronously from the host to the target. Since the host is the master, it is necessary that the target interface is very responsive to incoming pipe data. When the data is in a block fashion the transfer will not be successful even if the target can keep up with the throughput.

In order to control, configure and transfer data through these endpoints Opal Kelly provides a FrontPanel API, which is a powerful C++ interface to

communicate with the FPGA. Basically, it contains libraries with a large number of functions to communicate with the FPGA, data transferring to the endpoints being one of them. It also provides the capability to modify the FPGA configurations like the PLL (clock generator). This functionality broadens the application base of the integration modules and offers the opportunity to develop a customized application required for the project.

Finally, it is important to emphasize that the software is the bottleneck of the system. As explained in section 2.2, the USB interface has a transfer rate higher than the required throughput. However, what prevents the system from not needing a memory are the Front Panel API's functions. Reading data from the endpoint cannot be done continuously whenever there is a big amount of data to be transferred. The transfer instance to read the data automatically deasserts pausing the reading process and retakes the data read when possible. This reduces the effective data transfer making the use of a memory indispensable

3.2. FPGA Logic

This chapter describes the digitizing system structure, implementation and operation. Section 3.2.1 presents structural overview of the hardware design and explains its operation. Finally, Section 3.2.2 provides an in-depth description of each block.

3.2.1. Structure overview and operation

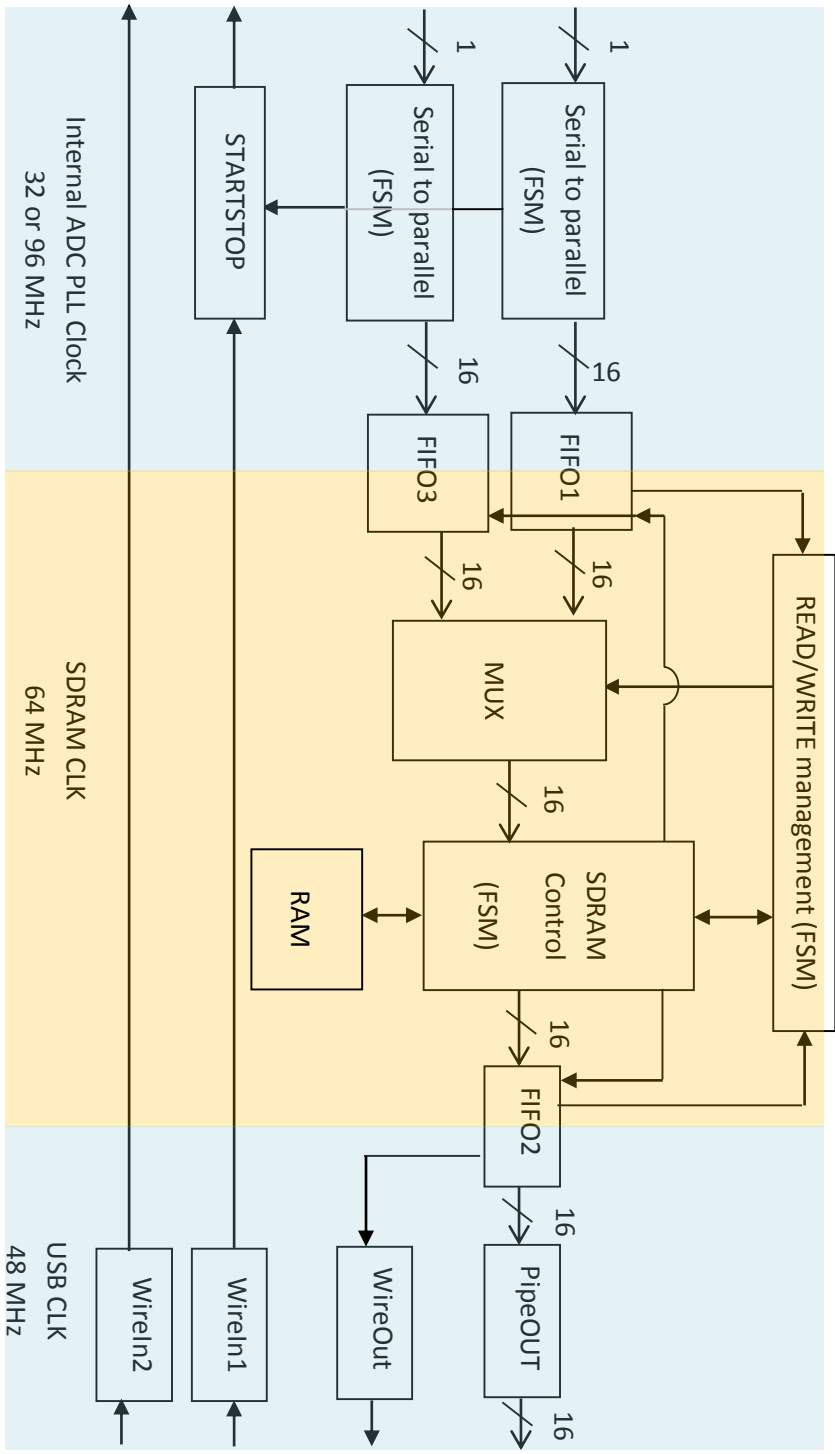


Fig. 9. Structure overview of the system

The whole system structure is shown in Fig.9. It consists of three different clock domains. The first clock domain is determined by the ADC configuration. The ADC uses a 32 MHz clock generated by the PLL on the FPGA board. Then, depending on the sampling rate (high or low speed mode), a PLL inside the ADC triples the frequency of the clock or keeps it same. The resulting clock (either 32 or 96 MHz) is used by the ADC to shift the data in a serial bus to the FPGA. The data is then deserialized and put into a parallel bus using the clock provided by the ADC. The block that controls the start signal send to the ADC is the STARTSTOP block, which is also connected to the computer.

The data from the ADC has to be stored in the memory to avoid overrun from happening. Nonetheless, the RAM memory operates at 64 MHz, clock provided by the PLL on the FPGA board. A change of a clock domain requires the implementation of a first in first out memory (FIFO) whose input and output can operate at different clock frequencies.

These FIFOs cannot have the required depth to store all the data sampled by the converter due to the FPGA memory limitations. Therefore, the data is transferred to the external RAM memory (in the XEM3010 board but not in the FPGA). The Read and Write management block work in conjunction with the SDRAM Control block in order to manage the data and control the reading and writing of the memory.

The data is retrieved from the RAM and transferred to the PipeOut endpoint, which sends it to the computer via USB. This endpoint, as well as the other WireIn and WireOut endpoints enable the FPGA to communicate to the computer.

Finally, the USB microcontroller uses a clock signal that comes from the USB itself and its frequency is different from the RAM clock frequency (48MHz instead of 64MHz). Therefore, a third FIFO memory (FIFO2) has to be implemented to ensure the correct functioning of the system.

3.2.2. Block description

In this section, all the important modules are presented and its function in the system is explained.

- Serial to Parallel

The “Serial to Parallel” block is implemented as a finite state machine that deserializes the data shifted by the ADC through a serial bus. Its interface is shown in Fig.10. As stated previously, this block operates with a clock generated by the PLL inside the ADC.

The data reading is preformed when the ready signal (DRDY) from the converter is triggered, as explained in section 2.1.2. Then the values of the serial bus on the consequent clocks is stored and send to the FIFO block memory. The memory requires an enable signal besides the clock signal to store the input data. Thus, an enable signal is set high during the writing process.

Every read cycle lasts 24 clock cycles, since that is the number of bits per sample. However, provided the RAM’s input width is 16 bits, it makes more sense to shift the parallel data through a 16 bit bus. As an alternative, another module between the FIFO and the RAM would be needed to convert the 24-bit parallel data to a 16-bit.

Lastly, an additional signal is used to count the number of digitized samples, which is used by the STARTSTOP block to control the start signal (conversion).

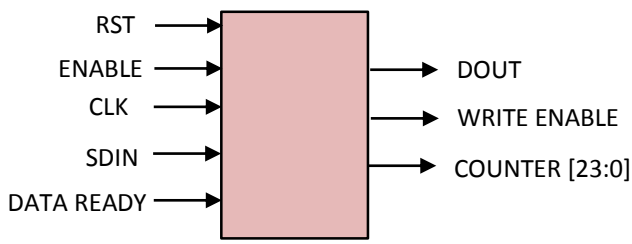


Fig. 10. Serial to parallel block interface

- SDRAM controller

This module is a controller for the RAM and it is provided by the board manufacturer, Opal Kelly. It is designed for managing the data transfer between a FIFO, the RAM and another FIFO and it has to be controlled by another component. Fig. 11 and Fig.12 show the block structure's interface.

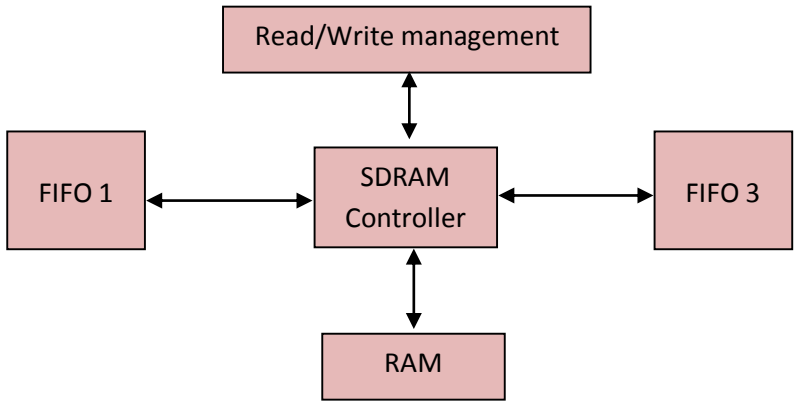


Fig. 11. SDRAM controller's block interface

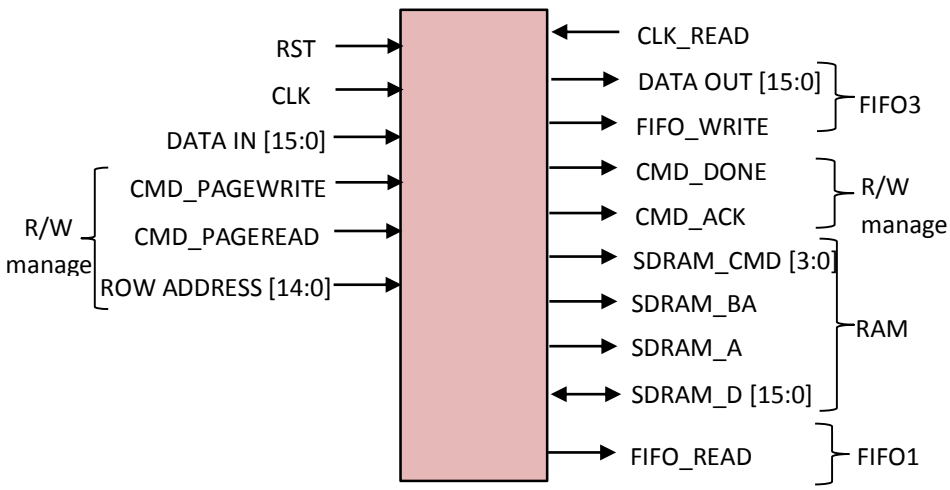


Fig. 12. SDRAM controller's interface

When the controller receives a write signal (cmd_pagewrite), it sends the appropriate sequence of write signals to the RAM (sdram_ba and sdram_a) and it enables the read signal to FIFO1. Then, the controller receives the data provided by the FIFO and stores it into the RAM using the sdram_d signal. This signals type (input/output) changes depending on the operation being performed (reading or writing respectively).

Whenever the read signal is active (cmd_pageread), the controller generates the appropriate sequence of signals (sdram_ba and sdram_a) and reads the data from the RAM through the sdram_d signal. Finally, it transfers these data to the FIFO2 while generating a trigger for the write signal.

This design requires the implementation of a multiplexer between the FIFOs supplying the data and the controller to select the source of data. Thus, the data from the FIFO will be received by the controller one clock later causing a storage of a 16-bit non-valid value. In order to solve this issue the read enable is send one clock prior to the original design, counteracting the extra clock delay added by the multiplexer.

- Read and write management

This block's function is to control the reading and writing in the RAM via the SDRAM controller block. It is the most important block in the system since it ensures that all the digitized data is properly stored and transferred without any loss.

The RAM can only be accessed once per cycle and consequently reading and writing cannot be two isolated processes, they have to be coordinated by this block. This block's interface is shown in Fig. 13.

The main priority is to prevent FIFO 1 and 3 from being filled when the converter is still functioning, since that would cause data loss. Consequently, when the data count of one of these FIFOS reaches a certain threshold a write signal (cmd_pagewrite) is sent to the RAM controller to proceed to write part of the data stored in the FIFOS into the RAM. This memory can read and write blocks of 512 16-bit data.

The other function of this block is to extract data from the RAM and store it into the last FIFO. This operation priority, however, only has to take place if the writing operation is not required at that moment (remember that both cannot take place at the same time). Another requirement for the

writing action to start is for the last FIFO to be able to store one block of data, which is the purpose of the input write data count signal.

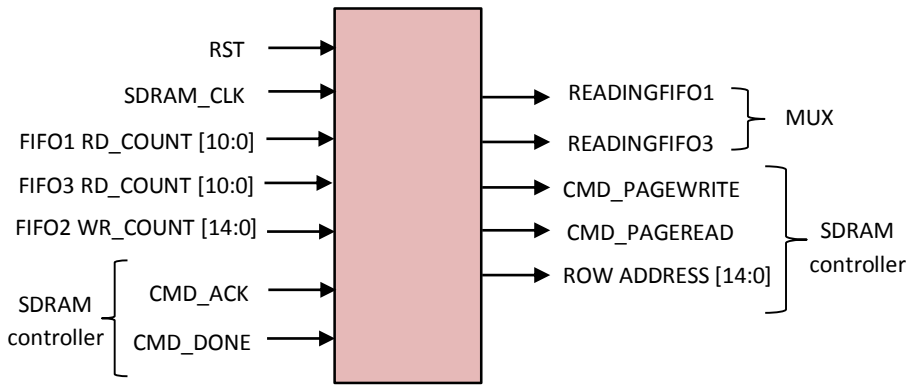


Fig. 13. Read and write management block's interface

The communication between this block and the SDRAM controller when reading and writing is very similar and works as follows. When a block of data has to be transferred the "cmd_pagewrite" or "cmd_pageread" is set high (writing or reading operation respectively) and the row address is read from this block. While the controller is transferring the data this block remains in a waiting state until it is finished and the controller triggers the "cmd_done" signal. Then, this block goes back to its initial state and checks again for the need to read or write another block of data.

A multiplexer is implemented since the controller has been designed for just one data input and this system has two channels. The two signals connected to it are used to multiplex the output of the FIFOs depending on the read data count signal that has triggered the writing process. These signals also serve another purpose as the FIFO read enabler signal from the controller need to be inversely multiplexed into two read enabler signals (one for each FIFO1 and FIFO3).

- Storage Blocks: FIFO1, FIFO2 and FIFO3

Three FIFOs are implemented in the design, one after each ADC and the last one after the RAM. They all have independent read and write clock since they operate in different clock domains as shown in the interface of the system (Fig. 9). The first two writing process operates with the ADC clock (32 or 96 MHz) while its reading frequency is the same as the RAM (64 MHz). The signal that enables the writing in the FIFO (WR_EN) is set to high by the "Serial to Paralel" block whenever the data put into a parallel bus (DATA IN). As explained previously in this section, the reading process is controlled by the SDRAM controller through the read enabler signal (RD_EN). The FIFO memory interface is presented in Fig.14.

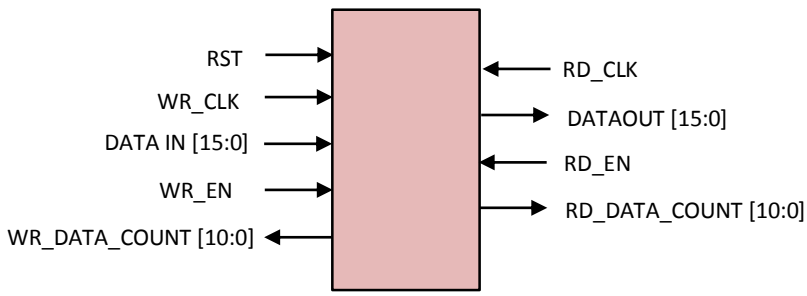


Fig. 14. FIFO memory's interface

FIFO2 has exactly the same structure as the other ones with different read and write clock signals for the same reason. The only thing that differs from the others is its depth, which also causes the write and read data count signals to increase in length. The signals indicate the amount of data written in the memory and the amount of data that can be read from it respectively.

- Endpoints

Several endpoints are implemented for the transfer of data between the logic and the PC software. Similar to the RAM, these endpoints, have a width of 16 bits.

First, the PipeOut endpoint transfers the data converted from the ADC. When the software executes the adequate command, a read signal is sent to FIFO2 and the data goes through this endpoint into the computer via USB. This reading process is done whenever the amount of data written in FIFO2 reaches a threshold, which is the maximum amount that the software can read at a time. This is explained more in depth in the software section (section 4.2). In any case, a signal has to be sent to the computer to indicate that the read can be done: that is the function of the WireOut endpoint.

Two other endpoints are implemented to transfer data from the computer to the FPGA. The WireIn2 carries all the control signals that determine the configuration of the ADC (sample rate, filter, etc...). The FPGA simply routs this data to the appropriate pins.

The remaining endpoint, WireIn1, is used to send the trigger signal that initiates the sampling process. It is not directly wired to the start pin in the FPGA but to the STARTSTOP block. The reason for that is not about the start of the sampling process but rather the end of it. The sampling is the fastest process in the system while the storage and transfer of the data takes longer. For this reason it is optimal to stop the sampling of the signals whenever the ADC has sampled the desired amount of data instead of waiting until whole of the data is transferred to the computer. Otherwise, the sampling of extra unnecessary data slows down the data storage as the RAM can only be accessed either for reading or writing, not both.

3.3. Interface board between the ADC and the FPGA

In this section, the design of the interfacing board to connect the second ADC channel to the FPGA is presented. Firstly, section 3.3.1 presents the location of the needed signals to control the second ADC through the FPGA. Secondly, the board designed to connect those signals is introduced in section 3.3.2. Finally, the specific connections between the different expansion connectors are detailed in section 3.3.3.

3.3.1. Connections needed to control the second ADC

The signals required for the proper functioning and control of the second-channel converter are the same as the ones for the first board, which are mentioned in Table 1 (section 3.1.1). As shown in Fig.15, all the control signals are connected to J6.

Controlling both boards with the FPGA board is possible since the FPGA has some unused pins in the original design that are wired to the FPGA board connection pins. At the same time, when this board connected to the ADC board, these same pins are wired to a free connector in the ADC board. These unused pins are pins 15 to 65 (except the ones connected to ground) from the expansion connector J7 and can be seen in Fig. 15. Therefore, all the signals from the FPGA into the second board have to go through these pins. However, two signals have to be connected in a different way; the START signal and the clock signal (FPGA_CLK). In order for both ADC's to sample simultaneously, the delay between the input starting signals (START) has to be minimal. Routing the start signal through the FPGA could cause an unknown delay which would vary depending on the design implementation. Consequently, it is wired directly from the connector to the second board making the delay constant. As explained in section 1.3. reducing the delay is important, but keeping it constant is even more crucial. This is because a constant delay can be fixed in the data processing while a variable delay will lead to unpredictable errors.

The second signal that stands out is the FPGA_clk, which is the clock generated by the PLL in the FPGA board. This signal is not connected to the FPGA and for this reason it cannot be connected to the second board through the FPGA logic; it has to be connected externally.

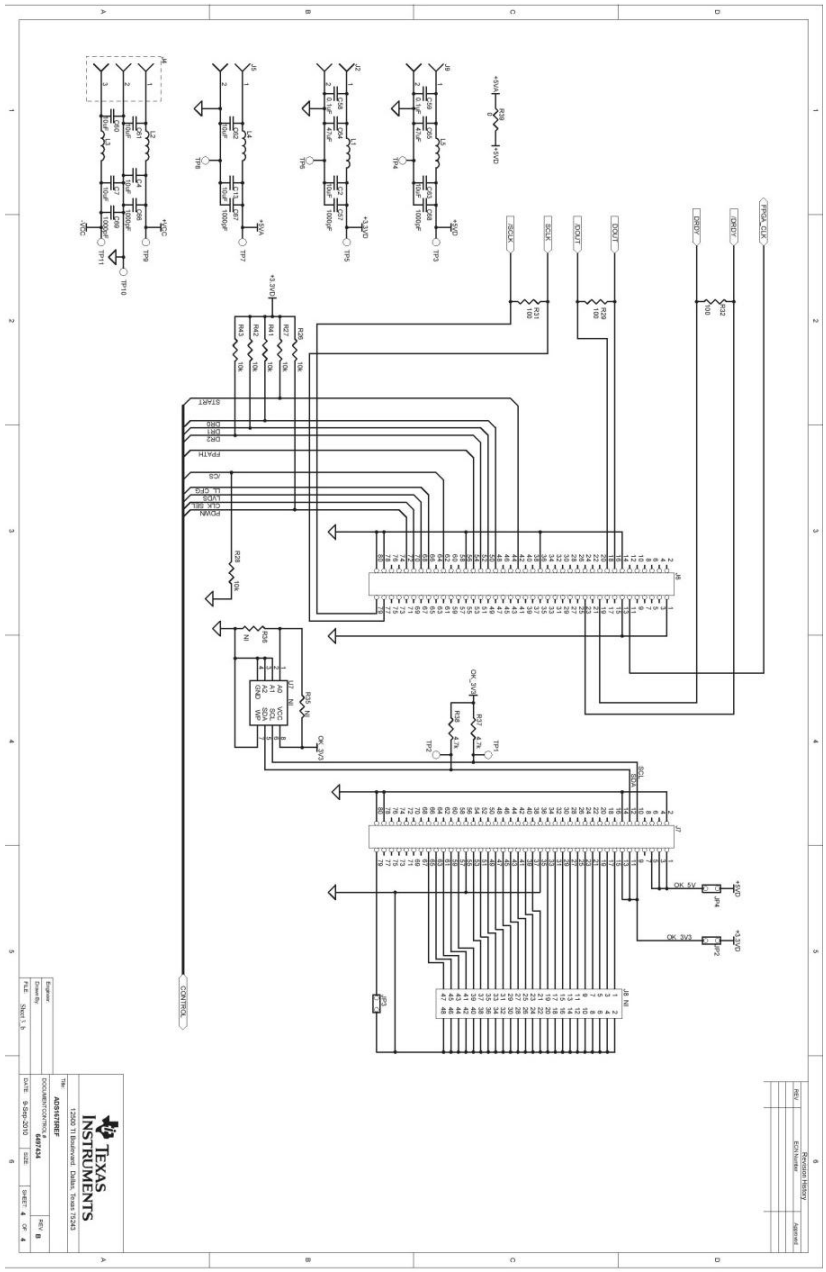


Fig. 15. ADC board connections to the expansion pins [11]

3.3.2. Connection board

Control signals have to be connected from the J7 connector of the first board to the J6 connector of the second board. For this purpose, the free connector in the reference board could be used. However, it is also necessary to connect two signals (FPGA_clk and start) from the J6 connector of the first board to the same connector and pin of the second board. Additionally, all the grounded pins from both boards also have to be connected, which means that even the J7 connector is used.

Considering that all the connectors are used and given that the FPGA board connections to the first ADC board have to remain the same, the best option to achieve a two channel digitizing board is to build a PCB board. This board can be placed between the ADC and the FPGA board enabling the original connections of the first channel while connecting the FPGA to the second ADC. It also facilitates the ground connections between boards as well as the proper wiring of the start and clock signals. Moreover, since no additional components are needed, directly connecting the pins through a PCB board reduces the potential delay and phase difference that can appear.

3.3.3. Design and pin connections

The board designed is placed between the first ADC board and the FPGA through 4 expansion connectors (2 upwards, 2 downwards). It also has two more connectors (downwards) that are connected to the second ADC board as shown in Figure 16. This solution, then, uses the expansion connector rather than the free connector (for practical purposes it is the same).

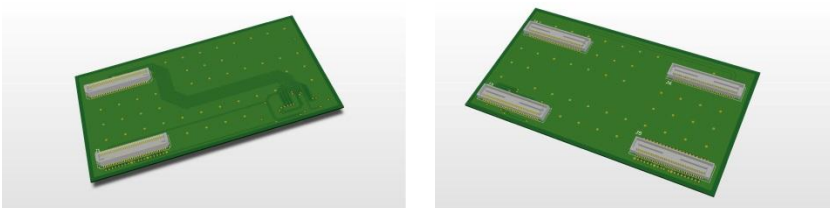


Fig. 16. PCB interfacing board

The connections between the first board and the FPGA board remain unchanged when adding the connection board. This way, all the signals from the first board are already connected to the FPGA. Therefore, all the pins from the J6 connector (first ADC control signals) are connected directly to the FPGA board through the connector just above (J1). Same applies for J7 connector, although it only carries ground and power supply signals (this is the connector used for the second board control signals). These connections are shown in Table 4.

For the second channel, the control signals generated by the FPGA are connected to the J7 first board connector, which is wired to the J6 second board connector through the interfacing board. This expansion connector is unused in the first board. All the control signals except the start and clock signals are connected this way as shown in Table 5.

Signal (ADC1)	Meaning	J6 (first board)	FPGA Pin	TYPE
DR0	Sample rate (0)	48	H13	OUTPUT
DR1	Sample rate (1)	50	H14	OUTPUT
DR2	Sample rate (2)	52	J18	OUTPUT
FPATH	ADC filter selection	54	J17	OUTPUT
CS	Chip selector	62	G18	OUTPUT
LL_Config	Filter configuration	66	E18	OUTPUT
LVDS	LVDS or CMOS	68	E17	OUTPUT
CLK_SEL	Internal/external clock	70	D18	OUTPUT
PDWN	Power down	72	D17	OUTPUT
DRDY_P	Data ready (P)	19 (*)	R16	INPUT
DRDY_N	Data ready (N)	23 (*)	P16	INPUT
DOUT	Sampled data (P)	16 (*)	U18	INPUT
/DOUT	Sampled data (N)	18 (*)	T18	INPUT

SCLK	Clock generated by the PLL in the ADC (for the output data)	77 (*)	F10	INPUT
/SCLK	Clock generated by the PLL in the ADC (for the output data)	79 (*)	E10	INPUT
START	Start signal	42	L14	OUTPUT

Table. 4. Connections between the first ADC board and the FPGA board

Signal (ADC 2)	Meaning	J7 (First board, same as FPGA)	J6 (second board)	FPGA Pin	Type
DR0_2	Sample rate (0)	15	48	U1	OUTPUT
DR1_2	Sample rate (1)	17	50	T1	OUTPUT
DR2_2	Sample rate (2)	19	52	R2	OUTPUT
FPATH2	ADC filter selection	21	54	R1	OUTPUT
CS2	Chip selector	23	62	P2	OUTPUT
LL_Config2	Filter configuration	25	66	P1	OUTPUT
LVDS2	LVDS or CMOS	27	68	N2	OUTPUT
CLK_SEL2	Internal/external clock	29	70	M1	OUTPUT
PDWN	Power down	31	72	L2	OUTPUT
DRDY_P2	Data ready (P)	37	19 (*)	K2	INPUT
DRDY_N2	Data ready (N)	39	23 (*)	K1	INPUT
DOUT2	Sampled data (P)	43	16 (*)	K4	INPUT

/DOUT2	Sampled data (N)	41	18 (*)	K5	INPUT
SCLK2	Clock generated by the PLL in the ADC (for the output data)	79	77 (*)	F9	INPUT
/SCLK2	Clock generated by the PLL in the ADC (for the output data)	77	79 (*)	E9	INPUT

Table. 5. Connections between the first ADC board and the FPGA board

It is worth mentioning that the choice of pins for the J7 expansion connector is not relevant for most of the signals since there are 24 free pins that are directly connected to the FPGA. However, it is relevant for the differential signals and the clock signal. Some of the signals generated by the ADC are differential signals. Thus, they require a differential input buffer, which is only implemented in certain pins of the FPGA(*). Also, the clock signal generated by the second ADC (SCLK2_clk) requires an input pin suited for a clock signal.

The start signal and the FPGA clock require a different routing. As explained previously, these signals cannot be routed through the FPGA, which means that they are connected directly from one ADC pin to the other externally through the interfacing board. For this reason, the connector and pin number are the same for both boards as shown in Table 6.

The ADC board and the FPGA board external power supply are not independent. The ADC board is supplied externally with a 5V, 9V and -4V. Besides, it also requires a $3{,}3V_{DC}$ supply, which is supplied by the FPGA board through the expansion connector pins 9, 11 and 13. It is necessary to connect them to power the second channel board for its functioning. This has been done ensuring that the FPGA board is capable of supplying voltage to both boards simultaneously

Similarly, the FPGA board demands a 5V supply even though this voltage is applied externally only to the ADC board. The expansion connectors enable this supply to get to the FPGA board through pins number 1, 3 and 5. Nevertheless, as the first ADC board is already supplies the required voltage, there is no need to connect these pins of the second board.

Finally, the rest of the ground and supply signals from each board are connected to ensure the proper functioning of the system. All of the connections are presented in table 6 and 7.

Signal	J6 (First board / FPGA)	J6 (second board)	FPGA Pin
Start	42	42	L14
FPGA_clk	11	11	-- (PLL,FPGA board)
Ground	1,13,14,36,56,78,80	1,13,14,36,56,78,80	--

Table. 6. FPGA J6 to second board connections

Signal	J7 (first board)	J7 (second board)	FPGA Pin
Power supply	9,11,13,2,14,35,55	9,11,13,2,14,35,55	--
Ground	1,13,14,36,56,78,80	1,13,14,36,56,78,80	--

Table. 7. FPGA J7 to second board connections

4. Software

The development of the application has been done using Microsoft Visual Studio and is mainly based on the FrontPanel API library. It contains a large amount of instances which are at the software developer's disposal facilitating the building and customization of the final product. The software flow diagram block of the application is shown in Fig.17 and it is explained in section 4.1 and section 4.2.

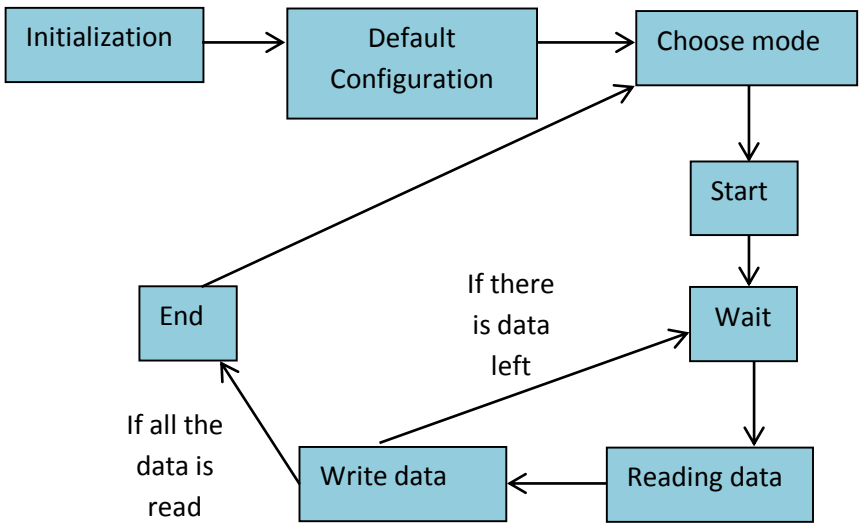


Fig. 17. Software flow diagram

4.1. Initialization and configurations

First of all, the libraries are loaded and the Front Panel host interface is enabled. Then, the PLL in the ADC board is programmed to generate the appropriate clocks, which are a 64MHz clock to the FPGA and a 32MHz clock to the ADC boards. These clocks are set at the outputs 0 and 3, respectively. Once this is done, the RAM is initiated and the default configuration of the device is set. After the configuration is completed, the start signal is triggered to validate the new configuration.

A basic menu is displayed on the screen, which enables the user to change the configuration of the device or to start the sampling process.

For a configuration change, the software asks for all the necessary information (data rate, filter and voltage levels) and transfers the information to the FPGA. Given that the configuration signals are routed directly from the endpoint to the FPGA output pin, it is only necessary to send the adequate value to the WireIn and trigger the start signal once (as in every new configuration setting). Each bit of the value sent through the WireIn endpoint represents an output pin value of the FPGA which is a configuration signal to the ADC. This configuration values are presented in Table 8.

Config [2:0]	DRATE (kSPS)	000	125 ksPS
		001	250 ksPS
		010	500 ksPS
		011	1000 ksPS
		100	2000 ksPS
		101	4000 ksPS
Config [3]	Voltage levels	0	LVDS
		1	CMOS
Config [4]	Filter Path	0	Wide-Bandwidth
		1	Low-Latency
Config [5]	Low latency filter config	0	Single cycle settling
		1	Fast response

Table. 8. Configuration values

4.2. Data acquisition

The software provides the user the option to choose between different amount of samples (1MS, 2MS and 4MS) and also the type of data output (text file or binary file). The main use for this system is to write all the data in a binary file, as it is extremely faster; doing it in a text file is just an additional feature as it takes around five times the amount of time compared to the binary output.

Regardless of the option selected by the user, the process to obtain the data is the same. The software sends a start signal through the WireIn1 endpoint. Then, the application keeps reading the signal sent through the Wireout endpoint that indicates that there is a block of data in the FIFO2 to be read. When this happens, the PipeOut endpoint is triggered and a block of data is transferred to the application. Finally, it is written into the corresponding file. This process is repeated until all the required data has been transferred to the file. The only difference between the different modes is the amount of blocks of data transferred and the type of file where the data is written.

When everything is finalized, the application proceeds to close the file so it doesn't remain open and some values of the FPGA are reset to be ready for the next conversion. Since it is interesting to be able to do successive conversions, the data is written in a new file with a different name every time. This way the data is not overwritten.

5. Results and verification

5.1. Results

The system has been tested using a function generator as the source of the analog signal on the input of one of the channels, while the other has been set as a constant voltage value. The testing setup is shown in Fig.18.

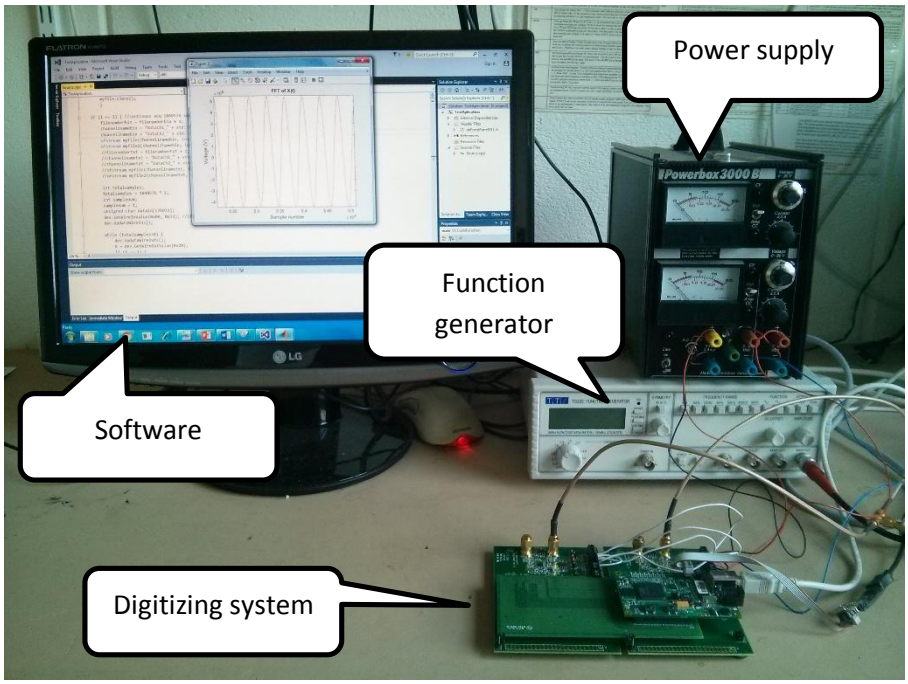


Fig. 18. Testing equipment

The result obtain is a sinusoidal signal as well as a constant signal obtained are shown in Fig.19.

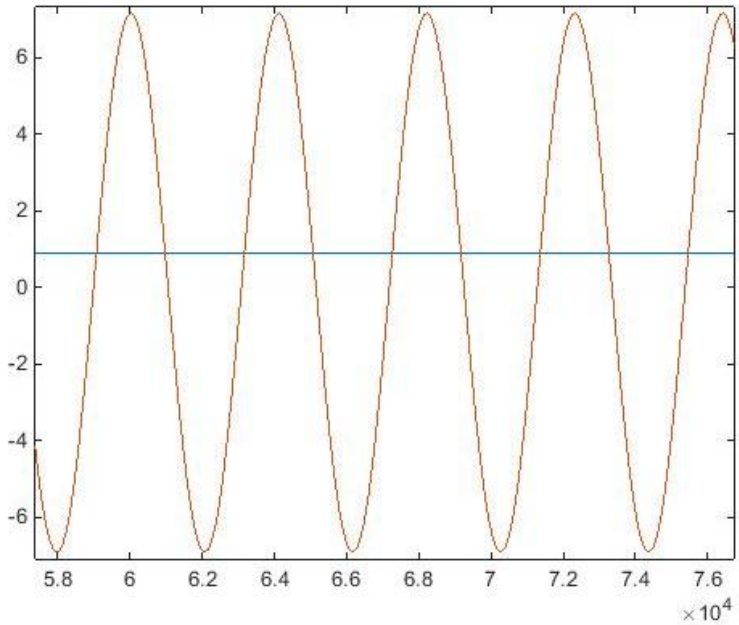


Fig. 19. Digitized signal obtained from the system

It is important to analyze more accurately the validity of the measurements as well as the systems bandwidth. In order to verify that the system can sample signals up to 1 MHz and that the input signal is sampled properly, the Fast Fourier Transform (FFT) of the digitized signal is calculated.

In order to present a proper graphic the y-axis is logarithmic (in dB) and it is normalized and adapted to the input voltage signal so that 0 dB represents the maximum amplitude. Also, the x-axis is normalized to the sample frequency (4 MHz). As shown in Fig.20, the only amplitude peak is at the signal's fundamental frequency (0,25f/fs = 1 MHz). The noise in the signal is very small, around 100 dB below the amplitude peak. However, this value should be higher in a digitizing system with 23 bits of resolution as calculated in (2), where N is the number of bits

$$DR = 6.021 \times N + 1.763 = 140{,}25\ dB \qquad (2)$$

The reason for that are the limitations of the ADS1675 itself. Its dynamic range at 125ksPS is 111 dB and it drops to 103 dB when working at 4MS/s [9]. Therefore, this system does not decrease the performance of the ADC as its dynamic range is very close to the maximum achievable with that ADC.

After the peak at 1 MHz, the amplitude drops as a consequence of the filters implemented in the system. It can be concluded that the system's bandwidth meets the required 1 MHz. Just as an indicator, the vertical line indicates the maximum bandwidth of the ADC, which is 1,7 MHz.

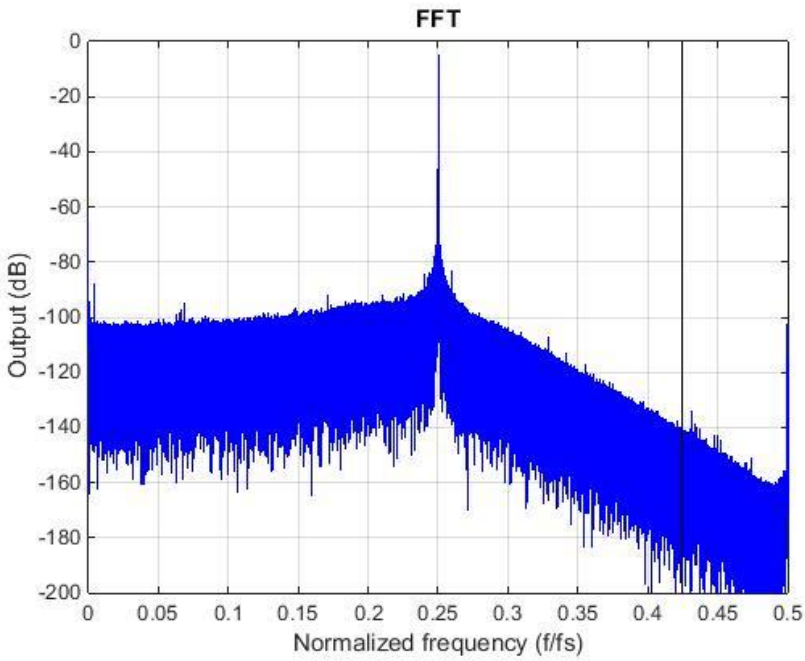


Fig. 20. FFT of the digitized signal

5.2. Latency of the system

A crucial aspect to be analyzed is the latency of the system, which is the time between the start of the digitizing process and the end of the data storage in a file. The values of the latency depending on the number of samples acquired can be found in Table 9. One of the requirements of the project was to acquire 1 MS from each channel and transfer them to the computer in one second or less (as that is one of the requirements). This is accomplished since the system takes 780 ms to acquire that amount of data.

Number of samples per channel (Msamples)	Time (ms)
1	780
2	1530
4	2670

Table. 9. Time required to acquire the data

It is interesting to compare these values to the time required for the reference board (ADS1675REF) to take this amount of samples, even though it only has one channel. The reference board is able to acquire 1Msamples from one signal in 2830 ms for a single data acquisition. Although that design has a more elaborated interface, the time does not include plotting the data, just writing it into a file. Therefore, a significant improvement in the speed of the system has been accomplished. One of the reasons for this is the optimization of the start signal pulse length. The original design sets the conversion signal (START) to high until the system transfers the desired amount of samples. This means that the converter keeps sampling even after the desired number of samples have been digitized. This slows down significantly the data storage process of the system. The system built in this project, as explained in section 3.2.1, controls the start signal through the FPGA logic instead of the computer software allowing it to set the start signal to low whenever the converter has sampled the data. This reduces the time drastically since, for example, in a 1MS conversion only 250ms are required for the ADC to sample while the

rest is just for data transferring. Thus, improving the data transferring makes a big difference.

Another improvement also contributes to this time shortening. In the reference board design not only the start signal is maintained high but also the system always samples the maximum data possible and then stores in a file the amount desired. This is why the system takes the same amount of time when sampling 1kS or 1MS. This is not optimal and is improved in this project's system, only acquiring and transferring the needed data.

5.3. Number of samples

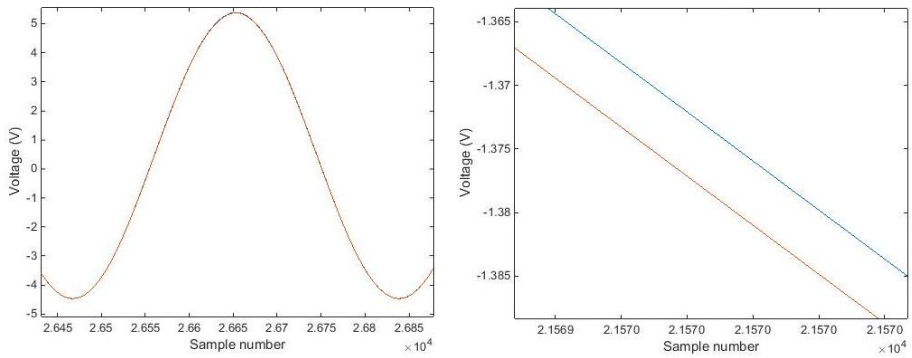
The design allows the user to acquire up to 4 MS per channel per data acquisition event. As explained in section 5.2, the software is the bottleneck of the system since its effective data transfer is not high enough. For this reason, all the data digitized in a conversion has to be able to be stored at once in the RAM to ensure that there is no data loss. Therefore, the maximum amount of data that can be digitized per data acquisition event is limited by the capacity of the RAM. This is the only way to ensure that there is no data loss.

Given that the resolution of the ADC is up to 24 bits (when in low-speed mode), a 4 MS acquirement requires 24 Mbytes of memory, as it shown in (3). The RAM memory included in the FPGA board has a depth of 32-Mbyte, which means that the acquisition of 6 or 8 MS per channel can lead to potential data loss. The maximum number of samples that can be stored is improved from the one channel reference design, since it only allows a maximum of 1 MS.

$$2\ channels \times \frac{4\ MS}{1\ channel} \times \frac{3\ Mbytes}{1\ MS} = 24\ Mbytes \qquad (3)$$

5.4. Phase delay

In order to analyze the phase delay of the system, the same signal has been split and connected to each analog input of the boards. Fig. 21 (A) shows the result of sampling both signals simultaneously while Fig.21 (B) is zoomed in to be able to see the phase delay.



(A) (B)

Fig. 21. Phase delay

The phase delay between both signals is 0,0025 rad (≡ 0,143°). This delay is small and below the maximum 1° acceptable for this application. Nevertheless, the most important aspect, once it has been checked that the delay is small, is that its standard deviation is not greater than 1° for different sample acquisition events. To check this, multiple measurements have been performed and the variance of the delay has been calculated. Table 10 presents the values of the measurements while Table 11 shows the mean, standard deviation and variance of the phase delays from the samples. The phase shift between the signals is small as well as its standard deviation. The mean of the different samples is 0,318° and its standard deviation is 0,247°, both lower than the maximum 1°required.

Measurement	Phase delay (rad)	Phase delay (°)
1	0,0025	0,143
2	0,0032	0,183
3	0,0142	0,813
4	0,0101	0,578
5	0,0035	0,200
6	0,0025	0,143
7	0,0054	0,309
8	0,0018	0,103
9	0,0024	0,138
10	0,0100	0,572

Table. 10. Phase delay of the signals

Mean (°)	0,0056	0,318
Variance($°^2$)	1,863e-5	0,0610
Std. dev. (°)	0,00432	0,247

Table. 11. Properties of the phase delay

5.5. Cost

The main concern about the cost is that it needs to be below the price of two separate digitizing systems. As shown in Table 12, the reference board ADS1675REF costs 5.000 SEK. This board consists of the ADC board and the FPGA, which are considered to cost 1000 SEK and 4.000 SEK respectively. Therefore, the ADC board used for the second channel adds 1.000 SEK to the system's cost. Finally, the interfacing board used to

achieve the two-channel system cost 1.200 SEK making the total cost 7.200 SEK. On the other hand, buying two ADS1675REF boards is 10.000 SEK. Therefore, the total cost of this system is 28% lower than buying two separate digitizing systems.

It is worth mentioning that for this project the cost of the interfacing board is 1.200 SEK because it had to be specially fabricated for this project. However, if this system was to be built in multiples, the price of the interfacing board would be lower as it is a PCB that only has the connections between the different expansion connectors, it does not contain any components.

Module	Price (SEK)
ADS1675REF	5.000
ADC board	1.000
Interfacing board	1.200
Total	7.200

Table. 12. Cost of the system

6. Discussion and Conclusions

In this work, the aim was to build a digitizing system suited for photocurrent detected two-dimensional spectroscopy techniques. A 23-bit 4MS/S two-channel digitizing system was required with a bandwidth of at least 1MHz. The system built in this project consists of two ADC boards connected to an FPGA board and using an interfacing board to build the second channel, a board designed in this project.

The results obtained when testing show that the system has the minimum bandwidth required of 1 MHz with an amplitude of more than 10^5 times the noise's amplitude. The time required for a 1 MS data acquisition event is below 1s (780ms), which is even lower than the time required for a one channel board with the same sample rate and resolution (ADS1675REF). This indicates that the FPGA implementation of the system improves the data sampling and transferring. Moreover, the Front Panel API software is the bottleneck of the system when transferring data which could be solved by using another FPGA, changing or upgrading the method to transfer the data to the computer (e.g. Ethernet) or even changing the whole FPGA board.

The maximum sampling capacity for a single data acquisition event is 4 MS per channel. This is limited by the capacity of the RAM in the FPGA board since it has to be ensured that data overrun does not occur. However, this amount of data meets the requirements for the project since it was required to be able to sample at least 1MS per data acquisition event.

The results show that the phase delay is low (0,318°) and its standard deviation is 0,247°, both below the maximum required value (1°). This means that the skew between the channels is small and its small standard deviation provides the possibility to diminish it through further data processing.

The size of the system built is a fraction of the size of a lock-in amplifier. Hence, in the spectroscopic techniques, it can be placed closer to the detector, which significantly reduces the noise to due to other interfering electronic signals. Finally, it is also worth mentioning that 24 bit digitizers are common in high end sound cards. The ones currently used sample at a rate of 192 KS/s and its bandwidth is about 90 kHz. Therefore, this implementation is about 20 times faster and has a bandwidth of more than 1 MHz, which allows high precision measurement of sounds at ultrasonic frequencies. Many applications in sonar, ultrasound medical imaging and vibration testing can benefit from this implementation.

7. Future work

In this section different suggestions are presented regarding future work.

- In order to obtain a system with better performance, all the modules in described in this project can be implemented in a single board. This can reduce the final cost of the system and improve its performance, like the phase delay since the length of the connections can be reduced. This eliminates the need for 3 different modules and therefore its size is reduced. Also, it means that only one board has to be supplied instead of the two ADC boards.
- Given that the software required to communicate with Xilinx FPGAs is the bottleneck of the system, relevant improvements can be made on speed if another FPGA is used. In this case, it is not necessary to implement all the system in one board but rather building another FPGA board replacing the FPGA itself. This also provides the option to include a RAM with greater capacity, which leads to an increase in the amount of data that can be sampled per data acquisition event.
- Another interesting aspect to consider is the increase on the number of channels. This, however, implies the need for a completely new system which would be based on this project. While the operation base is the same, for many reasons it is necessary to build a new system. First of all, the RAM does not have enough capacity to store such amounts of data from 4 channels. In second place, the FPGA capacity to implement FIFO memories would not be enough. Finally, the USB 2.0 transfer date is not high enough, which means that a USB 3.0 or another data transfer module has to be implemented.

References

[1] Karki et al. (2014), "Coherent two-dimensional photocurrent spectroscopy in a PbS quantum dot photocell" , *Nature Communications 5,* 5869 doi:10.1038/ncomms6869

[2] P.F. Tekavec et al. (2006), "Wave packet interferometry and quantum state reconstruction by acousto-optic phase modulation", *J. Chem. Phys*. 125 194303

[3] N.F. Scherer et al. (1991), "Fluorescence-detected wave packet interferometry: Time resolved molecular spectroscopy with sequences of femtosecond phase-locked pulses", *J. Chem. Phys*. 95, 1487

[4] P. F. Tekavec et al. (2007), "Fluorescence-detected two-dimensional electronic coherence spectroscopy by acousto-optic phase modulation", *J. Chem. Phys.* 127, 214307

[5] Karki et al. (2016), "Phase-synchronous detection of coherent and incoherent nonlinear signals", *Journal of Optics,* 18, 015504

[6] Chemla, D. S. & Shah, J. (2001) "Many-body and correlation effects in semiconductors", *Nature* 411, 549–557

[7] Karki et al. (2013), "Generalized lock-in amplifier for precision measurements of high frequency signals", *Review of Scientific Instruments,* 84, 115101

[8] M. Andersson et al. (2007), "Flexible lock-in detection system based on synchronized computer plug-in boards applied in sensitive gas spectroscopy", *Rev. Sci. Instrum*. 78, 113107

[9] Texas Instruments, "*4MS/S, 24Bit Analog-to-Digital Converter*", SBAS416D datasheet, Dec. 2008 [Revised August 2010]

[10] Opal Kelly, "*FrontPanel user's manual*", Opal Kelly Incorporated Portland, Oregon, 2005

[11] Texas Instruments, "*ADS1675REF*", SBAU162A user's guide, Dec. 2009 [Revised September 2010]