

Article

Low Complexity System on Chip Design to Acquire Signals from MOS Gas Sensor Applications

Juan B. Talens ¹, Jose Pelegri-Sebastia ^{1,*} and Maria Jose Canet ²¹ IGIC Institute, Campus Gandia, Universitat Politècnica de València, 46730 Gandia, Spain; juatafe@upv.es² Instituto de Telecomunicaciones y Aplicaciones Multimedia, Campus Gandia, Universitat Politècnica de València, 46022 Valencia, Spain; macasu@eln.upv.es

* Correspondence: jpelegri@upv.es

Abstract: Analog signals from gas sensors are used to recognize all types of VOC (Volatile Organic Compound) substances, such as toxic gases, tobacco or ethanol. The processes to recognize these substances include acquisition, treatment and machine learning for classification, which can all be efficiently implemented on a Field Programmable Gate Array (FPGA) aided by Low-Voltage Differential Signaling (LVDS). This article proposes a low-cost 11-bit effective number of bits (ENOB) sigma-delta Analog to Digital Converter (ADC), with an SNR of 75.97 dB and an SFDR of 72.28 dB, whose output is presented on screen in real time, thanks to the use of a Linux System on Chip (SoC) system that enables parallelism, high-level programming and provides a working environment for the scientific treatment of gas sensor signals. The high frequency achieved by the implemented ADC allows for multiplexing the capture of several analog signals with an optimal resolution. Additionally, several ADCs can be implemented in the same FPGA so several analog signals can be digitalized in parallel.



Citation: Talens, J.B.; Pelegri-Sebastia, J.; Canet, M.J. Low Complexity System on Chip Design to Acquire Signals from MOS Gas Sensor Applications. *Sensors* **2021**, *21*, 6552. <https://doi.org/10.3390/s21196552>

Academic Editors: Ilaria Fratoddi, Anna Staerz and Nicolae Barsan

Received: 28 July 2021

Accepted: 27 September 2021

Published: 30 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: sigma delta; ADC; SoC; Altera; DE-1-SOC; MOS gas sensor; LVDS; FPGA

1. Introduction

In order to understand the metal oxide semiconductors (MOS) which are used as gas sensors in multiple applications [1–4], we must consider them at the molecular level. Volatile organic compounds such as methane can react with oxygen, yielding carbon dioxide and water as products (Equation (1)):



MOS sensors mainly use tin dioxide, indium oxide or tungsten oxide as semiconductors and alumina as substrate [5,6]. The use of a catalyst reduces the amount of heat necessary to provoke the reaction. Figure 1 shows the sensor reaction in two cases: clear air and in reaction with methane. The dashed square is the gas sensor, which includes four terminals, a heater and Al_2O_3 as substrate. In clean air (Figure 1a), a tin oxide semiconductor has high resistivity which prevents electron movement. When methane gas is introduced (Figure 1b), the reaction seen in Equation (1) occurs. As a result of this movement of electrons, the resistivity of the gas sensor decreases and allows current flow in the sensor. This type of sensor, which has a resistive output, is simple to use and inexpensive [4,6]. When connecting a resistor R as shown in Figure 1a, a voltage divider is formed, and the output voltage can be measured. That is, the gas sensor becomes an analog-output sensor.

Due to the nature of the reaction and the polarity of the current, the measured voltage is always positive. On the other hand, it presents some thermic noise and an offset voltage due to the heater provided and the material's conductivity.

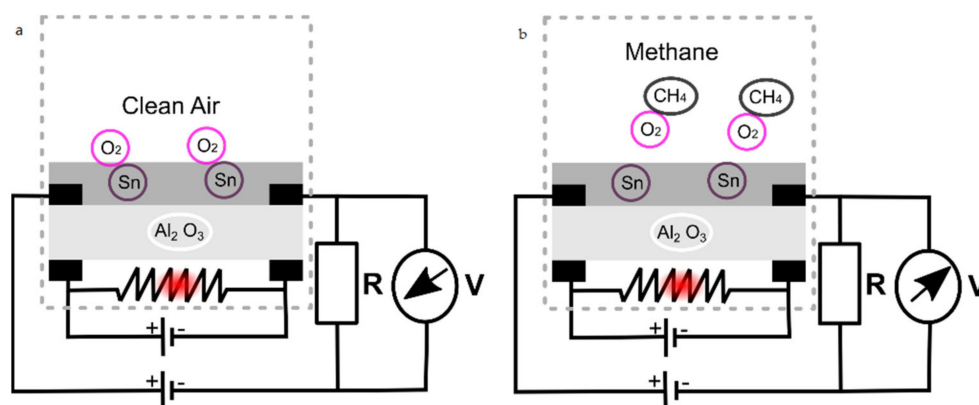


Figure 1. (a) Sensor reaction with clean air. (b) Sensor reaction with methane.

As some substances such as urine or feces have VOCs [7–10], gas sensors are an excellent option to obtain information from them without contact. Many electronic noses (e-Noses) have been proposed in the literature [11–15] to analyze data from gas sensors. The first stage of an e-Nose consists of an array of gas sensors connected to an ADC, that is, the acquisition stage, which is the focus of this paper. Figure 2 shows the usual stages in an e-Nose that classifies substances. After the acquisition, the digital signal can be processed (filter, windowing, normalization, offset compensation) in the same e-Nose or in a PC. Lastly, machine learning and classification stages are usually neuronal networks [16,17] in charge of training the system and recognizing the sample from gas sensor data [18], respectively.

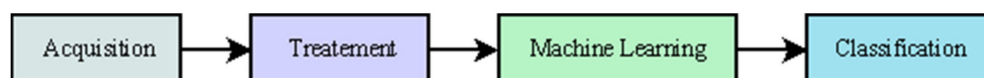


Figure 2. Stages in an e-Nose classifier.

In this paper, a solution for the acquisition stage that can be implemented by taking advantage of the LDVS input of an FPGA with a Linux System on Chip (SoC) system is proposed, which allows the implementation of the complete e-Nose Classifier in Figure 2 in the same device. Thus, our proposal is an alternative to microprocessor boards such as Arduino's and ADC-specific boards, which cannot be used for complex machine learning and require a PC. Our proposal also has the advantage of including an operating system, which allows us to develop other interesting applications such as the capture visualization.

Regarding the implementation of the acquisition stage using LDVS inputs of an FPGA, several proposals can be found in the literature. In [19], an 8-bit Approximation Register (SAR) and a 10-bit sigma delta are proposed, which require 135 and 1 k Lookup Tables (LUTs), respectively. In both approaches, an RC circuit connects the 1-bit DAC output to the FPGA LVDS input. The first solution proposes a 4-bit SAR, but only the most significant bit is connected to the RC. This is a low-cost solution for low-frequency analog input signals (up to 1 kHz). The second solution in [19] consists of a sigma-delta modulator with a sampling register and a cascade integrated comb (CIC) filter. This is more complex, but it can be used for analog input signals of up to 50 kHz. In [20], a 12-bit sigma-delta ADC similar to the one proposed in [19] is presented. The main difference between both proposals is that [20] uses an equalizer and a finite impulse response (FIR) filter apart from the CIC that adds more complexity to the design (700 LE-Logic Elements are required).

The purpose of this paper is to design an ADC for analog e-Nose signals. The main characteristic of these signals is their low frequency, so we do not need complex ADC techniques that include a CIC or an equalizer. The ADC proposed in the paper is based on the sigma-delta technique, but the CIC filters in [19,20] are substituted by a registered accumulator. The proposed solution is valid for low-frequency analog signals and allows a reduction in the complexity of the ADC.

Apart from the implementation of the ADC on FPGA, the scope of this paper also covers the hardware configuration of the SoC and the software programming that allows one to view the output of the gas sensor in a video graphics array (VGA) monitor. For the application under study, as many ADCs as gas sensors in the e-Nose need to be implemented, which can be achieved by implementing several ADCs in the same FPGA and/or multiplexing the gas sensors' outputs.

This paper is organized as follows: the ADC's specifications, the evaluation board used, and the test of the designed ADC are detailed in Section 2; Section 3 is dedicated to the implementation (hardware and software); Section 4 describes the results and conclusions are derived in Section 5.

2. Specifications, Materials and Test

This section presents the specifications of the input signal, which is obtained from an e-Nose, and the materials used to test the implemented sigma-delta ADC, including the board DE1-SoC from Terasic [21].

2.1. Input Signal

As commented above, the analog signal to acquire is obtained from an e-Nose. In [22], an e-Nose named MOOSY32, which is composed of 32 sensors, is used to identify an olfactory pattern of prostate cancer in urine. This e-Nose uses a sample rate of 100 Hz, 14-bit and 12-bit ADC National Instruments boards. Figures 3 and 4 show the captured signals, truncated to two decimal places, that were used to identify patterns. As can be seen, the signals change slowly (the steepest slope found corresponds to a frequency of 15 mHz) and they are always positive, lower than 2 V and with some offset.

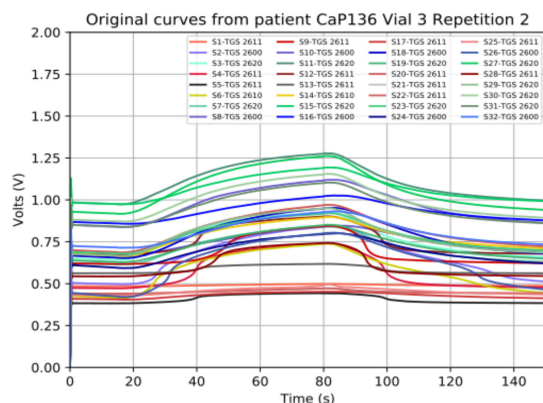


Figure 3. Curves of a prostate cancer patient extracted with the e-Nose of 32-sensor MOOSY32 after the ADC conversion.

In this paper, we propose substituting the ADC National Instruments boards with a sigma-delta ADC implemented on an FPGA, where post-processing could also be implemented in the future. As the signal only requires two decimal places after conversion, our target is to obtain an ADC with a minimum of 9 bits of resolution.

2.2. Board DE1-SoC from Terasic

The proposed sigma-delta ADC for MOS gas sensor signals was implemented on a Terasic DE1-SoC [21], which is a development kit DE1-SoC. This card is a development kit around a system on chip (SoC) that combines a Cyclone V FPGA with an ARM cortex-A9 (HPS4). This FPGA contains 85 k programmable elements.

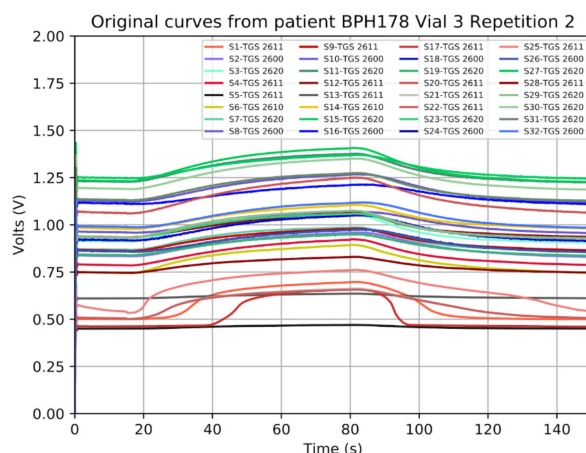


Figure 4. Curves of benign hyperplasia patients extracted with the e-Nose of 32-sensor MOOSY32 after the ADC conversion.

Table 1 details the FPGA included in the DE1-SOC card.

Table 1. Specifications FPGA DE1-SoC.

FPGA Device
Cyclone V SoC 5CSEMA5F31C6 Device
Dual-core ARM Cortex-A9 (HPS)
85 k Programmable Logic Elements
4450kbits embedded memory
6 Fractional PLLs
2 Hard Memory Controllers
Fclk max 50 MHz

2.3. Test

2.3.1. Initial Test Using Simulink

First, a simulink model of the sigma-delta ADC was created to validate the design. Then, after implementing the ADC on the FPGA, a final test was performed. Figure 5 shows the simulink model of the designed ADC that was used to simulate its behavior before starting the implementation on FPGA.

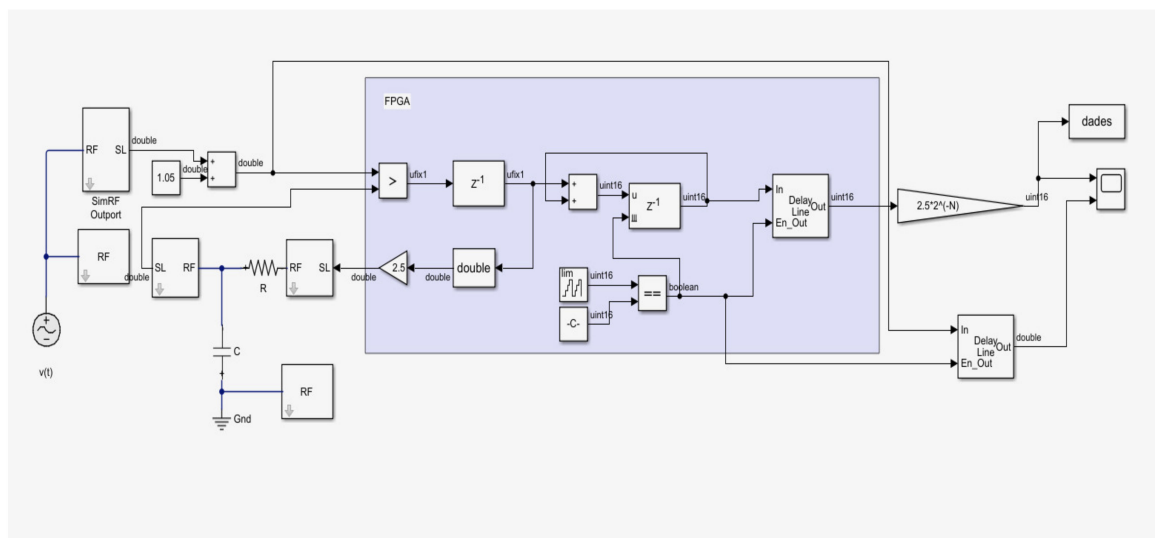


Figure 5. Test model $\Sigma\Delta$ implemented in Simulink.

2.3.2. Final Test

Once the desired results were obtained with Simulink, the sigma-delta ADC converter was implemented on FPGA (see Section 3) and then tested using the scheme shown in Figure 6. The analog input signal was obtained from a generator Rigol DS1054, configured to generate a low-frequency sine signal. The ADC was implemented on the FPGA, but it also required an external RC circuit. This RC circuit and the signal generator were connected to the GPIO pins, whereas a monitor, a mouse and a keyboard were connected to their corresponding ports of the development board. In this way, the use of an external PC to perform the test and to process the output of the ADC was avoided. The implemented ADC was managed by the operating system (HPS-hard processor system), which also controlled an SD card with the Linux system, the keyboard and the mouse. The digital output of the ADC was represented on the screen and stored to be processed. In addition to the ADC component, the VGA display drivers were also implemented on the Cyclone V FPGA. In this way, the signal generated is captured and showed in real time.

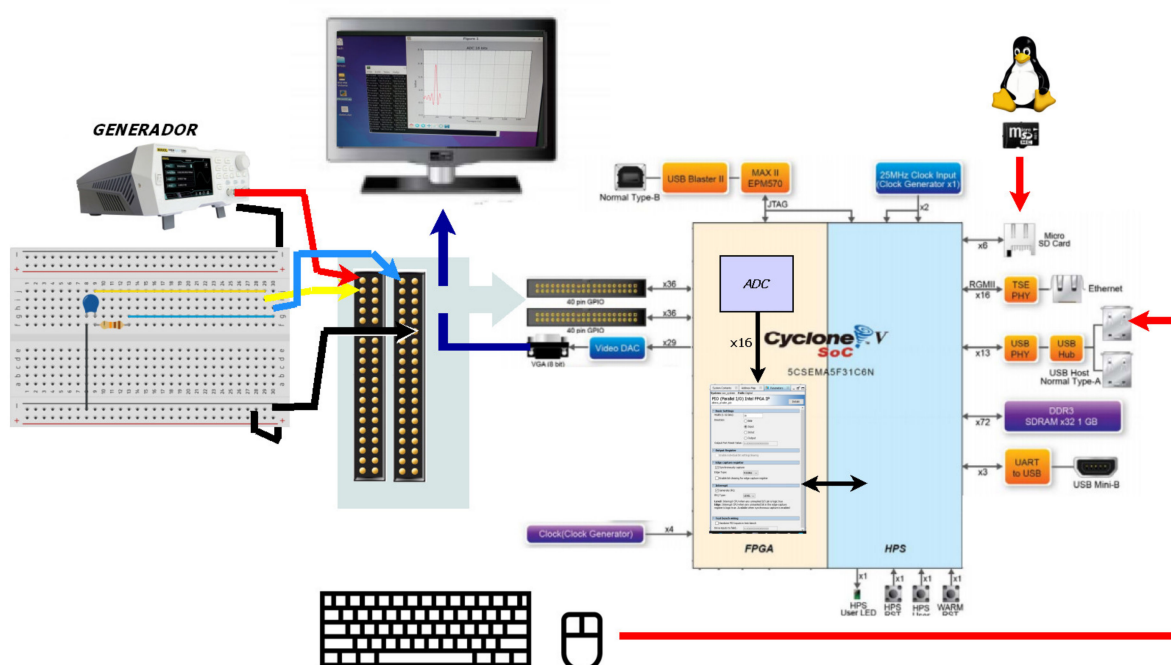


Figure 6. Test of the ADC converter.

3. Implementation

The system developed consists of a few elements mapped as hardware on the FPGA and an app to control that hardware. This section details the designed low-complexity system on chip design for the signal acquisition of MOS gas sensor applications, which includes: (a) the ADC; (b) the mapping of this ADC in an Altera Intellectual Property (IP), which is necessary for the HPS to control the ADC; (c) the application to visualize the captured signal on the monitor. The implemented ADC requires only 172 ALMs (Adaptive Logic Modules) on the FPGA.

3.1. ADC

As said above, the proposed ADC is based on the sigma-delta technique, but this is simplified thanks to fact that signals captured by e-Noses have very low frequencies. The block diagram of our proposal can be seen in Figure 7. Five stages can be distinguished: an

RC filter, a comparator (implemented as LVDS), a 1-bit DAC, an accumulator and a sample and hold (S&H) register. All stages work at a high frequency, (f_H) but the S&H register works at a low frequency (f_L), that is, the output sample frequency, $f_s = f_L$. To guarantee a resolution of 9 bits, 16 bits are used internally for the accumulator and S&H register. For this reason, f_H was set to $f_H = f_L \cdot 2^{16}$. The accumulator can grow from 0 to 65,535; if the input signal is greater than the registered input, then the output of the accumulator was increased by one; if not, it remained the same. In this way, the register of the accumulator stores the value of the input signal after 65,536 cycles. Then, the S&H captures this value at f_L and the register of the accumulator is cleared (every 65,536 cycles) to start to process the following sample of the analog input.

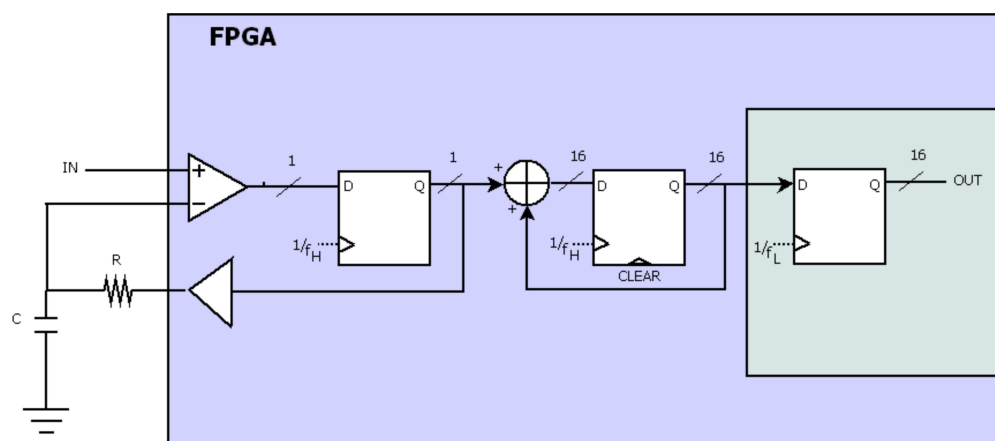


Figure 7. Theoretical model $\Sigma\Delta$ implemented.

3.1.1. Filter RC

The RC filter (Figure 8) integrates the output of the 1-bit DAC. The output of the RC filter was compared with the analog input (IN) using a LVDS input of the FPGA. The values of R and C were obtained so that a pole is provided at $f_L = f_s$. Figure 9 shows the frequency response of the filter for $f_s = 100$ Hz.

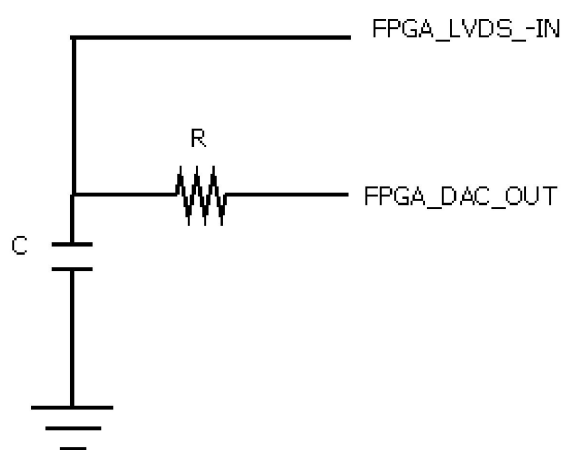


Figure 8. Stage filter RC.

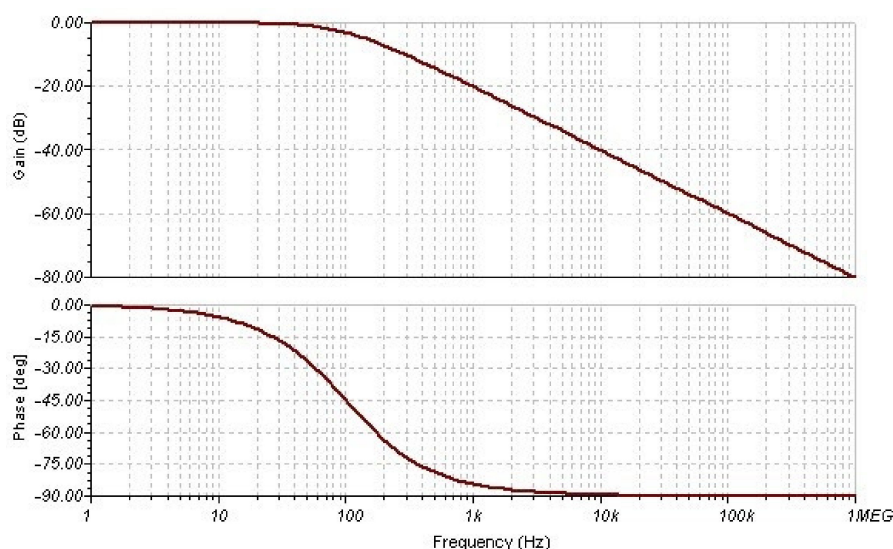


Figure 9. Filter response pole in 100 Hz, $R = 10.61 \text{ k}\Omega$, $C = 150 \text{ nF}$.

3.1.2. LVDS

LVDS input is similar to an operational amplifier; it has an inverting input, a non-inverting input, and an output, as can be seen in Figure 10a. It works non-linearly, internally subtracting both inputs [9] and giving $\text{FPGA_LVDS_OUT} = 1$ if the result is positive and $\text{FPGA_LVDS_OUT} = 0$ if the result is negative. They are manufactured with a certain hysteresis because their purpose is to offer a stable output with small variations in voltage. This produces a dismount of its resolution [12]. The threshold or minimum differential input voltage required by the component for a common mode voltage of 1.25 V is 100 mV [17].

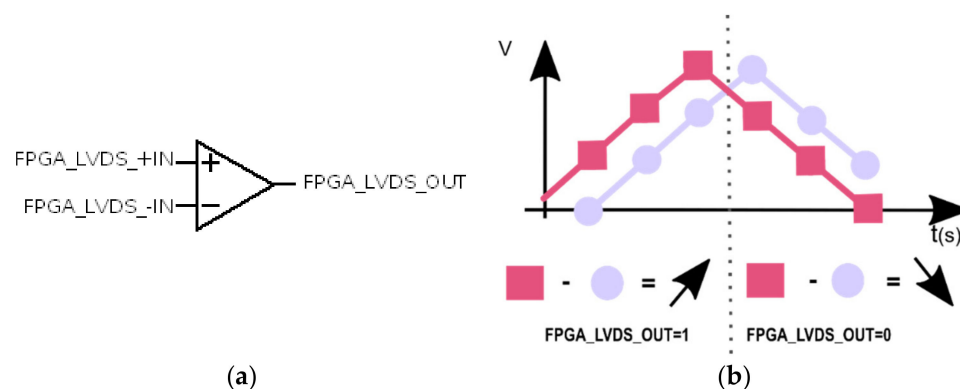


Figure 10. (a) Scheme LVDS and (b) stage LVDS.

Figure 10b represents the inputs of the LVDS, the analog input from the sensor (in squares) and the output of the RC filter (in circles), which comes from the 1-bit DAC that works at f_H . When the slope of the analog input is positive, the output of the LVDS will be 1; in any other case, it will be 0. The LVDS input cannot detect positive slopes lower than its minimum differential input voltage. When this happens, the 1-bit DAC output will be 0 and the voltage in the LVDS non-inverting input will decrease thanks to the integration made by the RC filter, so the differential input voltage increases. If the differential input voltage is higher than the minimum, the LVDS will be 1 (the positive slope is detected); if not, the 1-bit DAC output will be 0 and the LVDS non-inverting input will decrease again. In this way, the differential input voltage is continuously increased until the positive slope is detected.

3.1.3. 1-bit DAC

It is necessary to obtain a delayed copy of the input signal to be digitized. Figure 11 shows the 1-bit digital-to-analog converter consisting of a 1-bit register at frequency f_H . Its output is connected to an FPGA output buffer to be integrated by the RC filter, and also to the integrator. In our e-Nose application, we need $f_s = f_L = 100$ Hz, so f_H should be $100 \times 2^{16} = 6,553,600$ Hz. Thus, 65,536 digital-to-analog conversions are required to obtain an output sample at f_L .

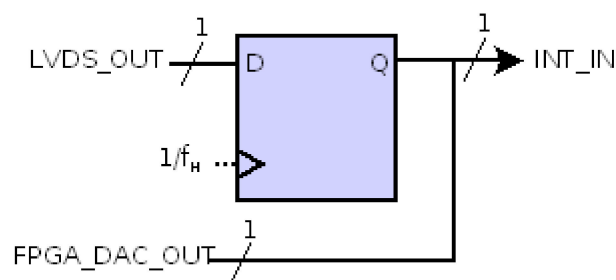


Figure 11. Stage 1-bit DAC.

Figure 12 shows the output of the 1-bit DAC (INT_IN in Figure 10b) when the analog signal (IN in Figure 6) is sinusoidal (amplitude = 0.5 V, offset = 0.5 V and frequency = 50 Hz). To see a representative output, f_H was set to 65,536 Hz. Note the frequency modulation of the 1-bit DAC output, which follows the slope of the analog input signal.

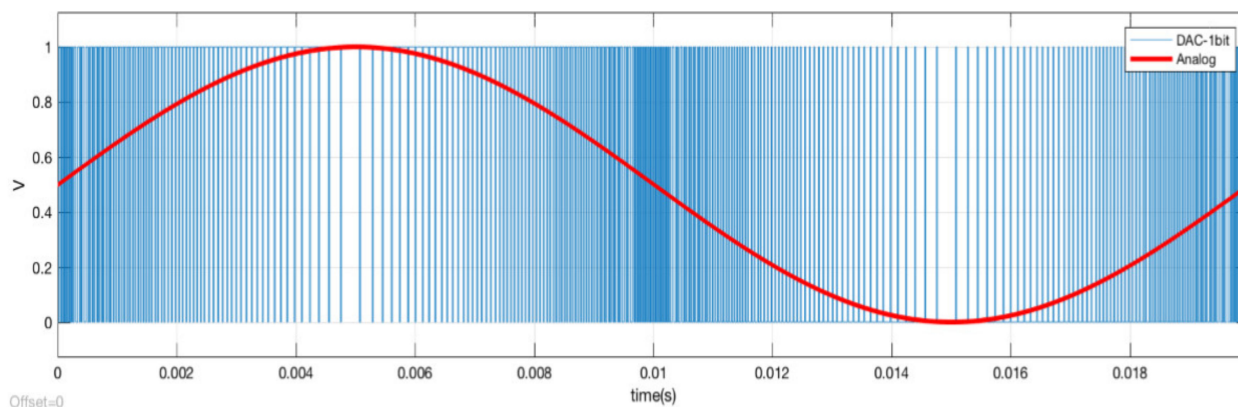


Figure 12. Sinusoidal signal input analog $v(t) = 0.5 + 0.5 \sin(2\pi 50t)$ output.

3.1.4. Input/Output Intel FPGA Intellectual Property

The implemented ADC is integrated as an Altera Intellectual Property (IP) and connected to bus avalon with altera_avalon_pio component. In this way, the ADC is connected to the HPS (Hard Processor System) as can be seen in Figure 6. Figure 13 shows how it is configured. This component is an input because it collects data from the ADC converter output to be read by the operating system. This is possible thanks to the memory mapping of the component within the system bus.

3.2. Application to Visualize the Captured Signal

In addition to the design of the ADC, we also designed an application to store and represent the captured signal on-screen. To achieve this, the ARM needs to use the Linux operating system (Ubuntu, London, UK) to control the resources of the development board, including the FPGA. Next, the configuration of the board and the design of the program used to obtain this representation are explained.

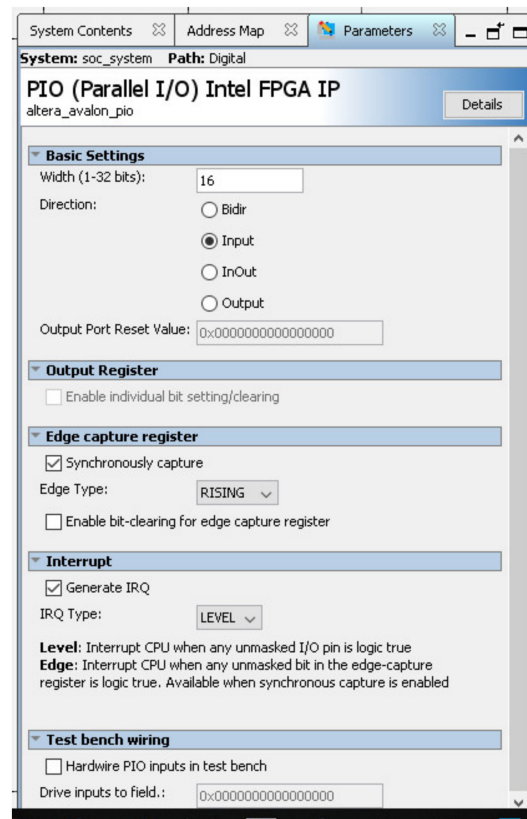


Figure 13. Input component altera_avalon_IO with 16 bits.

3.2.1. Configuration of Board DE1-SoC to Use SoC

First, the file system on an SD card needs to be prepared with the file system and the system image, and it should include:

- A Pre-loader or secondary U-boot loading program generated with bsp-editor tool. The U-boot loads three files from the first FAT partition of the SD.
- The Device Tree Blob or device tree, which tells the Kernel the hardware that is connected. Some details need to be input manually by editing the tcl files and removing duplicates from the dts file before generating the dtb file.
- The Raw Binary File or bit stream to configure the FPGA system. This file is generated by the convert programming files tool (Quartus).
- Linux kernel or system kernel downloaded from ©Terasic.
- Linux Root File system or file system and folders to mount the Linux. Fortunately, the ©Terasic website offers an image of the SD with a system of examples that can be used as a starting point.

On the other hand, the following components need to be created using the Platform Designer tool:

- FPGA Clock Signal from a board of 50 MHz.
- Clock signal from a pll component (f_H).
- ADC component designed in this paper.
- Additional input component to connect the ADC component with the bus.
- HPS component, interrupt capturer component and VGA controller component, which can all be obtained from the example given by ©Terasic.

Finally, the IP catalogue tool is used to generate IPs for the LVDS inputs and for the buffer output; input pins are configured as LVDS and output pins as buffers.

3.2.2. Program Design

The HPS on the evaluation board has two cores, and each core has one thread, but it has to complete more than two tasks: be aware of the operative system interruptions, read data from the ADC and plot these data on-screen in real time. The data from the ADC change every 10 ms, so the data-reading task needs to have priority over the rest of the tasks. Figure 14 shows the flowchart of the developed script, which contains two threads: one to read the data, which uses a token to block the CPU while it is reading and otherwise puts this thread to 'sleep', and another for the representation, called Animation, which can be executed while the reading thread is 'sleeping'. Once the last sample of the ADC is read, all the captured data are written to a file.

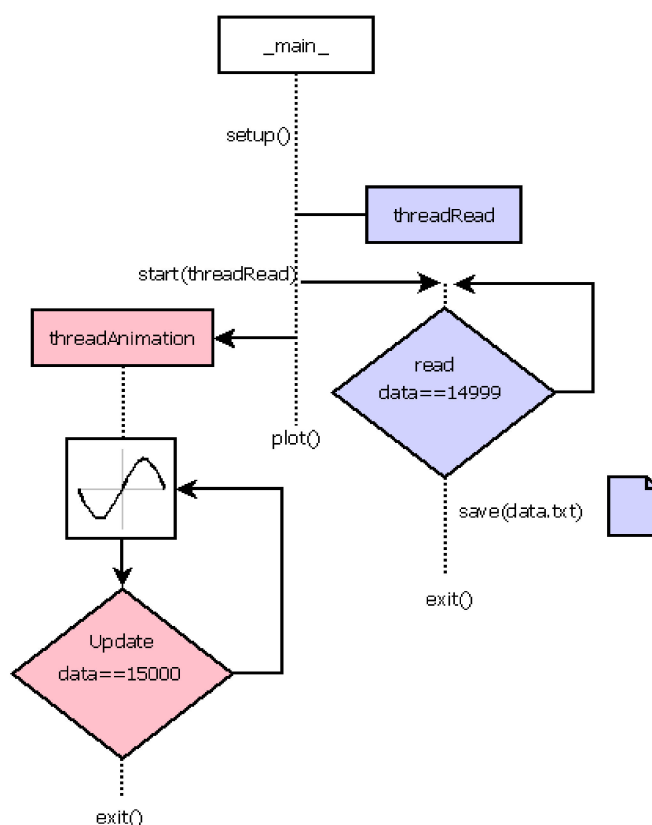


Figure 14. Python3 script flowchart.

An alternative to our proposal is to use both cores in parallel. In this case, two independent processes are required (reading and representation), but both require the same input data (the ADC output). As processes do not include data or memory space, a pipe needs to be used, so more time is required to run a process and put the other process 'on hold' when a core needs to attend to other processes in the system. On the contrary, the change of execution thread in our proposal, which uses two threads in the same process, is faster due to the fact that threads share data and memory space.

4. Results

This section presents the results obtained from the simulation and from the implementation of the proposed ADC.

4.1. Simulation Results

First, we obtained the effective number of bits (ENOB) of the ADC Simulink model for a 16-bit sine input signal. Table 2 shows the parameters of the ADC converter (f_L , f_H , R

and C) and the resulting ENOB, 14.75 dB, which is obtained from the SFDR (Spurious Free Dynamic Range) measure in Figure 15 using Equation (2), where N is the number of bits.

$$\text{SFDR} = 6.02N - 3.92 \text{ dB.} \quad (2)$$

Table 2. Simulated signal acquisitions and ENOB results.

f	f _L	N (Bits)	f _H (Hz)	R (Ω)	C (F)	ENOB (Bits)
35.5 mHz	100 Hz	16	6,553,600	10.61 × 10 ³	150 × 10 ⁻⁹	14.75

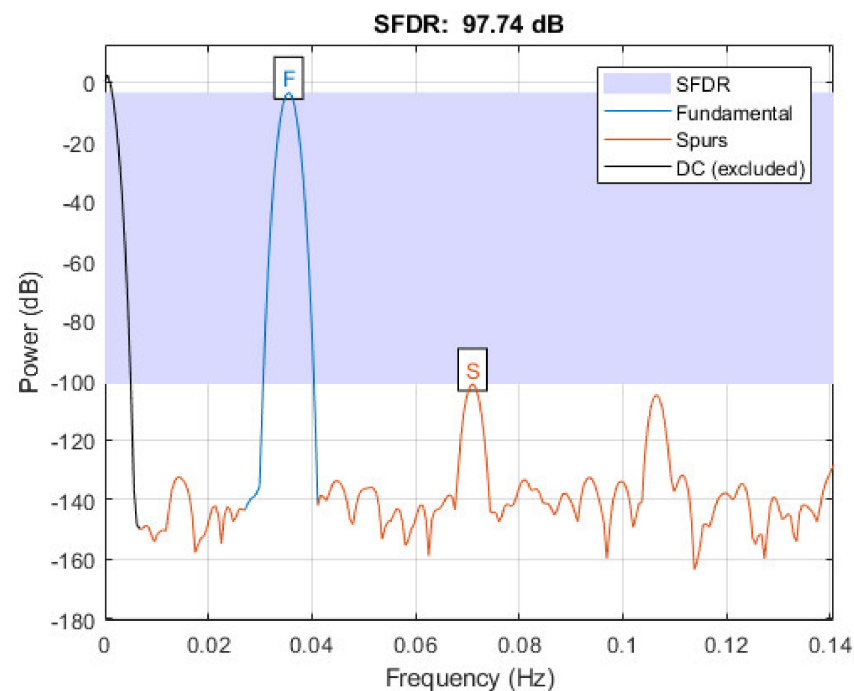


Figure 15. SFDR from 35.5 mHz Matlab signal simulation.

Note that an ideal comparator is used in this model, so the non-linearity of the LDVS is not affecting the measure.

4.2. Implementation Results

In this section, the results of the test of the ADC implemented on Terasic DE1-SoC are detailed. To obtain these results, we used the scheme showed in Figure 6, with a sine input signal of 35.5 mHz, a maximum at 2 V and a minimum at 100 mV.

4.2.1. Gain and Offset

Next, the output of the implemented ADC needs to be adjusted to obtain a signal with the same amplitude and offset as the sine input. In this way, we are compensating for the non-linearities introduced by the LVDS input.

Figure 16 compares the ADC's output with the ideal sine signal. A slight difference in gain and offset can be observed, which is compensated by the software (multiplying by $0.1014/0.1118 = 0.907$).

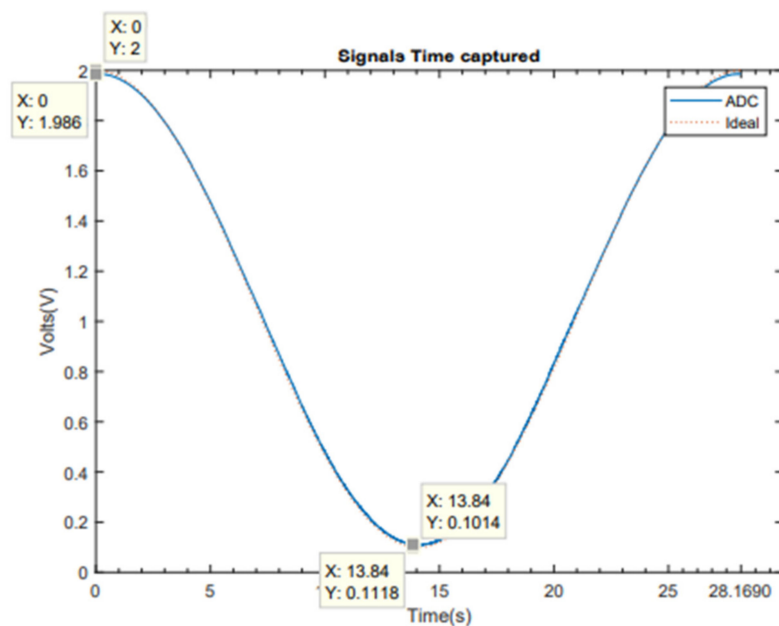


Figure 16. Ideal signal with captured signal.

Figure 17 shows the ADC’s output once the gain is compensated, so the difference in the offset is calculated by $2.012 - 2 = 12 \text{ mV}$, $0.1124 - 0.1 = 12 \text{ mV}$. This offset is also compensated by the software, so we obtain a sine signal with the same amplitude and offset as the input.

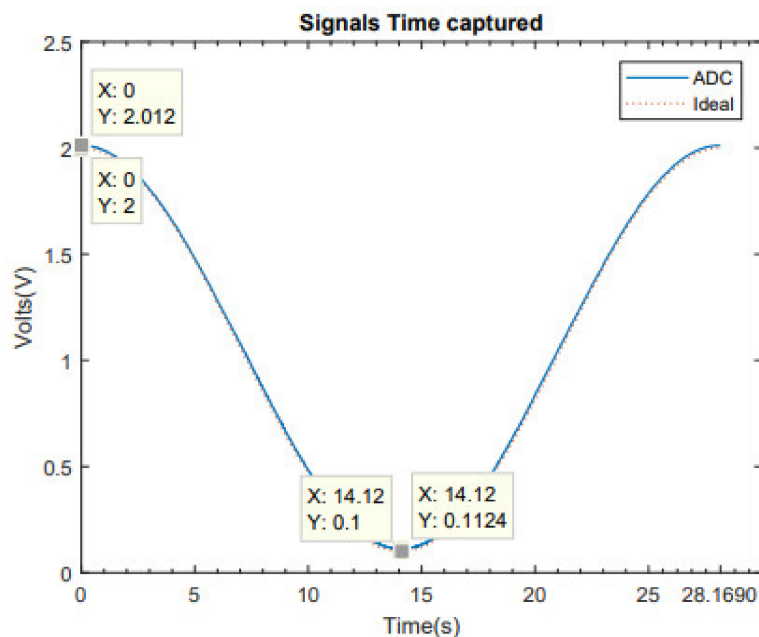


Figure 17. Ideal signal with captured signal with gain adjustment.

4.2.2. Effective Number of Bits (ENOB and SFDR)

The effective number of bits (ENOB) for a sine input signal is obtained with (3):

$$\text{ENOB} = (\text{SNR}_{\text{db}} - 1.761) / 6.02 \tag{3}$$

Figure 18 shows the power spectral analysis of the captured signal. The measured signal-to-noise ratio (SNR) is 75.97 dB, so the effective number of bits is 12 when (3) is applied. SFDR can also be measured in Figure 19 as 72.28 dB, and (2) can be used to obtain

the effective number of bits, which is 11, which suggests a 1-bit reduction in comparison with the SNR measurement.

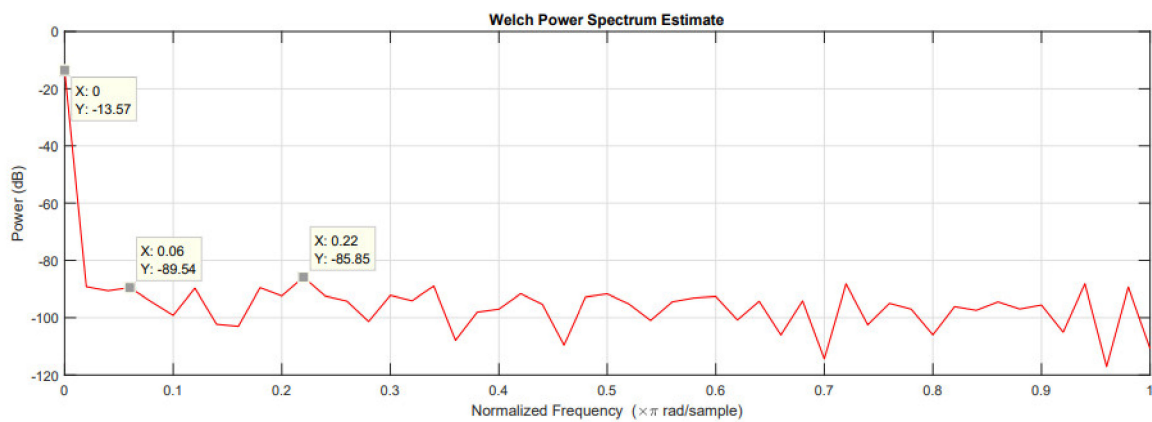


Figure 18. Power spectrum estimation for the ENOB calculation.

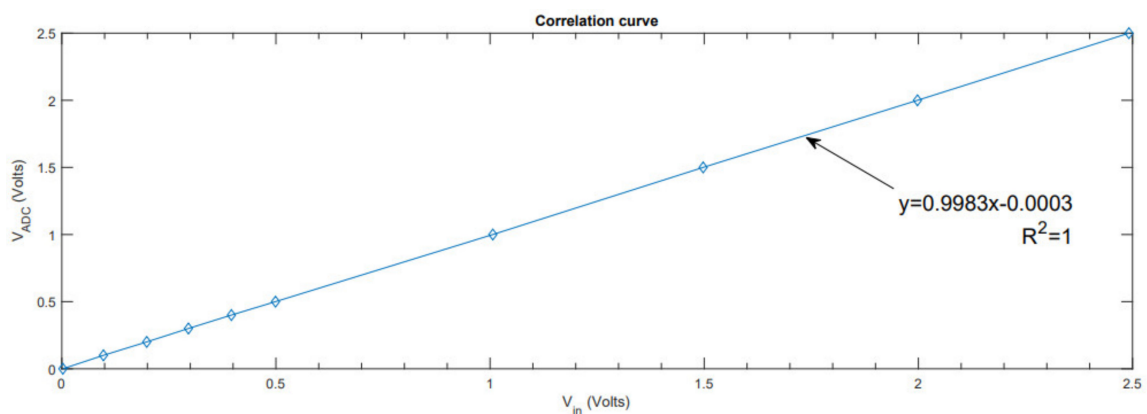


Figure 19. Transfer function with captured signal.

4.2.3. Transfer Function

Figure 19 shows the transfer function of the ADC, that is, the output with respect to its input. From this figure, we can conclude that the system works linearly.

4.2.4. Frequency Sweep

The implemented ADC was also tested with sine input signals of different frequencies that emulated the slope of the e-Nose signals showed in Figures 3 and 4 (from 6 mHz to 15 mHz). The signals captured using the test in Figure 6 (real measurement) can be seen in Figure 20. The results of the test were identical to the ones showed above.

Finally, Figure 21 shows the ADC output capture for an input signal with a sweep from 1 mHz to 3 Hz (the generator scroll was used to get the input signal). This figure shows the measured results obtained using the test shown in Figure 6.

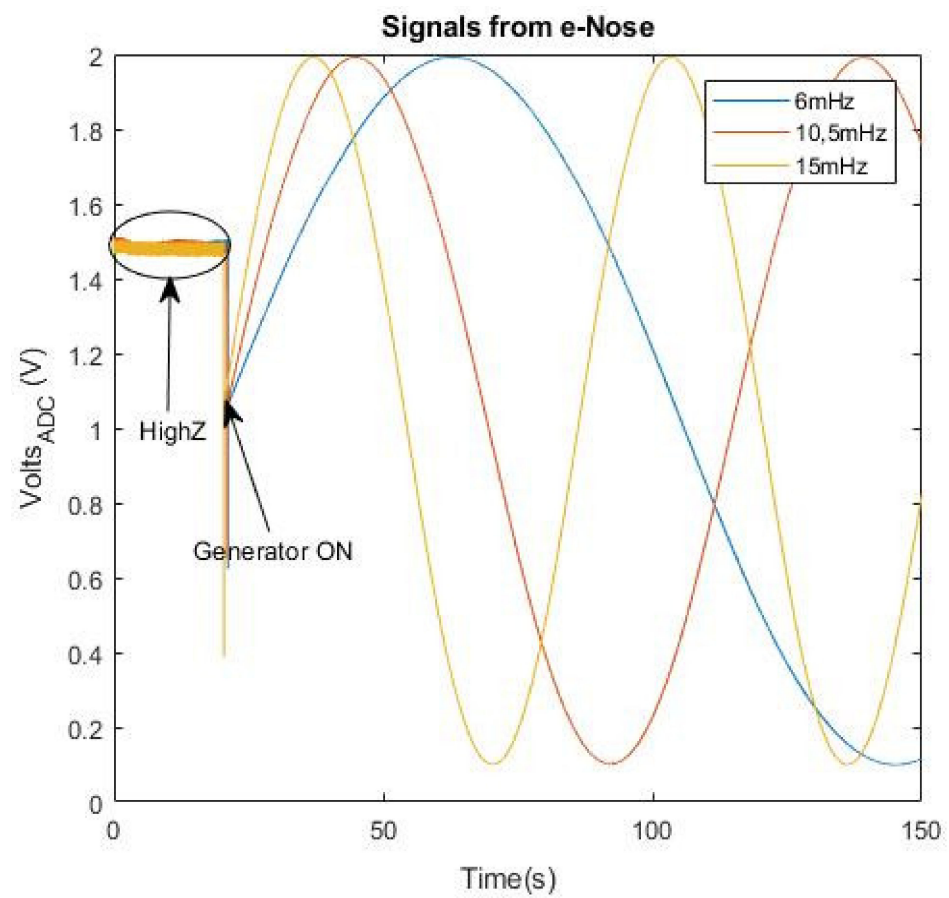


Figure 20. Signals for 6 mHz, 10.5 mHz and 15 mHz.

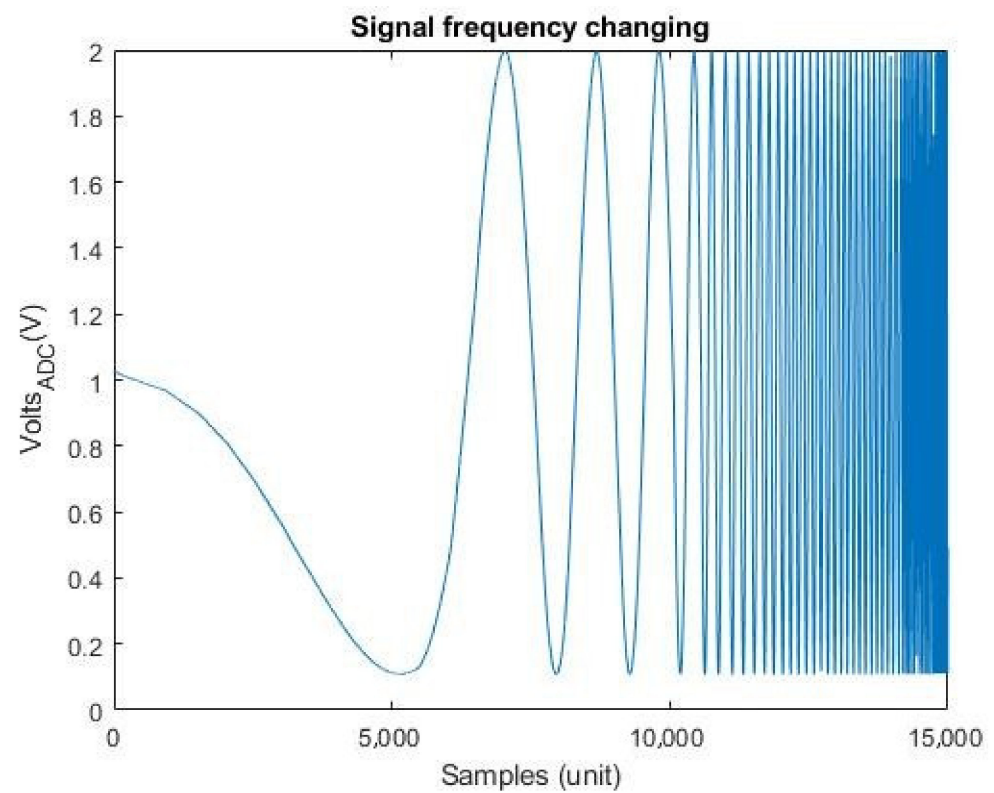


Figure 21. Sweep of a signal 1 mHz to 3 Hz.

5. Discussion

In this section, we compare the implementation results of our proposed ADC with [19, 20], which propose ADCs taking advantage of the LDVS inputs of an FPGA. Table 3 shows the resolution, the ADC type, the resources on the FPGA used in each case and the type of application (low or high frequency). An extra column has been added to show the total number of four-input LUTs required by each design. It should be taken into account that this comparison is not fair because all designs were not implemented using the same device. As explained in Section 2.1, our application requires a minimum of 9 bits of resolution, so the 8-bit SAR proposed in [19] is not adequate even though it only needs 135 LUTs. Among the rest of the proposals, our design requires fewer four-input LUTs and meets our application's resolution requirement.

Table 3. Comparison of the implementation results of our proposed ADC with [19,20].

	Type and Bits	Resources (Device)	4-Input LUTs	Application
[19]	8-bit SAR	135 LUTs (Lattice XP2-17)	135	Low frequency
[19]	10-bit sigma delta	1000 LUTs (Lattice XP2-17) + 2 sysDSP blocks	1000	High frequency
[20]	12-bit sigma delta	700 LE (Stratix EP1S25)	700	High frequency
Our proposal	11-bit sigma delta	172 ALMs (Cyclone V)	344	Low frequency

It is worth mentioning that the work presented in this paper has added value because an SoC system was used and this allowed us to visualize the captured signal in real time, as shown in Figure 6, without using a PC. Additionally, this fact allows the implementation of a complete e-Nose classifier in the same SoC (acquisition, treatment, machine learning and classification).

6. Conclusions

In this work, an ADC converter to digitalize signals from an e-Nose with a gas sensor is proposed. It reached a resolution of 11 bits, which is enough for our application, with fewer resources than similar proposals, because it takes advantage of the signal characteristics: non-periodic, positive and with smooth changes. The designed ADC has an SNR of 75.97 dB, a SFDR of 72.28 dB and linear transfer function V_{ADC}/V_{in} . This proposal was verified for one gas sensor, but it is scalable for an array of sensors. This can be achieved by implementing up to 13 ADCs in the same FPGA working in parallel (there are 40 inputs in the FPGA card that can be configured as LVDS, and three inputs per ADC are required). If more than 13 sensors are required, the output of several sensors can be multiplexed. As the DE1-SoC card allows working with frequencies of 50 MHz, each ADC can work with $50 \text{ MHz}/6553600 \text{ Hz} = 7$ MOS gas sensors multiplexed in time.

After including the required ADCs for the sensor array, the classification of samples could be carried out by implementing neuronal networks in the same FPGA, which optimizes the use of resources because it avoids the use of a PC or a similar device.

Author Contributions: Conceptualization, J.P.-S. and M.J.C.; Data curation, J.B.T.; Funding acquisition, J.P.-S.; Investigation, J.B.T. and J.P.-S.; Methodology, J.B.T. and M.J.C.; Project administration, J.P.-S.; Resources, J.P.-S.; Software, J.B.T., J.P.-S. and M.J.C.; Supervision, M.J.C.; Validation, J.P.-S. and M.J.C.; Writing—original draft, J.B.T. and M.J.C.; Writing—review & editing, J.B.T., J.P.-S. and M.J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the I + D + I Program of the Generalitat Valenciana (AICO/2016/046), Spain.

Data Availability Statement: The following are available online at <https://youtu.be/k0M14nm-XKw>, 389 Video S1: Real time acquisition.

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

References

1. Sanaeifar, A.; Mohtasebi, S.S.; Ghasemi-Varnamkhasti, M.; Ahmadi, H. Application of MOS based electronic nose for the prediction of banana quality properties. *Meas. J. Int. Meas. Confed.* **2016**, *82*, 105–114. [[CrossRef](#)]
2. Singh, H.; Bhaskerraj, V.; Kumar, J.; Mittal, U.; Mishra, M.D.; Nimal, F.; Cupane, A.T.; De, M.; Qi, I.; Feng, P. Food Industry Production. *IEEE Sens. J.* **2016**, *16*, 1–5.
3. Schiffman, S.S.; Wyrick, D.W.; Nagle, H.T. Effectiveness of an electronic nose for monitoring bacterial and fungal growth. In Proceedings of the 7th International Symposium on Olfaction and Electronic Noses, Brighton, UK, 20–24 July 2000; pp. 1–8.
4. Meixner, H.; Lampe, U. Metal oxide sensors. *Sens. Actuators B Chem.* **1996**, *33*, 198–202. [[CrossRef](#)]
5. Watson, J.; Yates, R.A. Solid-State Gas Sensor. *Electron. Eng.* **1985**, *5*, 47–57.
6. Batzill, M.; Diebold, U. The surface and materials science of tin oxide. *Prog. Surf. Sci.* **2005**, *79*, 47–154. [[CrossRef](#)]
7. Filipiak, W.; Mochalski, P.; Filipiak, A.; Ager, C.; Cumeras, R.; EDavis, C.; Agapiou, A.; Unterkofler, K.; Troppmair, J. A Compendium of Volatile Organic Compounds (VOCs) Released By Human Cell Lines. *Curr. Med. Chem.* **2016**, *23*, 2112–2131. [[CrossRef](#)] [[PubMed](#)]
8. Wilson, A.D. Application of Electronic-Nose Technologies and VOC-Biomarkers for the Noninvasive Early Diagnosis of Gastrointestinal Diseases. *Sensors* **2018**, *18*, 2613. [[CrossRef](#)] [[PubMed](#)]
9. Vernia, F.; Valvano, M.; Fabiani, S.; Stefanelli, G.; Longo, S.; Viscido, A.; Latella, G. Are Volatile Organic Compounds Accurate Markers in the Assessment of Colorectal Cancer and Inflammatory Bowel Diseases? A Review. *Cancers* **2021**, *13*, 2361. [[CrossRef](#)] [[PubMed](#)]
10. Nagle, H.T.; Schiffman, S.S.; Gutierrez-Osuna, R. How and why of electronic noses. *IEEE Spectr.* **1998**, *35*, 22–31. [[CrossRef](#)]
11. Yan, J.; Guo, X.; Duan, S.; Jia, P.; Wang, L.; Peng, C.; Zhang, S. Electronic Nose Feature Extraction Methods: A Review. *Sensors* **2015**, *15*, 27804–27831. [[CrossRef](#)] [[PubMed](#)]
12. Wilson, A.D. Advances in electronic-nose technologies for the detection of volatile biomarker metabolites in the human breath. *Metabolites* **2015**, *5*, 140–163. [[CrossRef](#)] [[PubMed](#)]
13. Rusinek, R.; Gancarz, M.; Nawrocka, A. Application of an electronic nose with novel method for generation of smellprints for testing the suitability for consumption of wheat bread during 4-day storage. *Lwt* **2019**, *117*, 108665. [[CrossRef](#)]
14. Fine, G.F.; Cavanagh, L.M.; Afonja, A.; Binions, R. Metal Oxide Semi-Conductor Gas Sensors in Environmental Monitoring. *Sensors* **2010**, *10*, 5469–5502. [[CrossRef](#)] [[PubMed](#)]
15. Yang, D.; Gopal, R.A.; Lkhagvaa, T.; Choi, D. Metal-oxide gas sensors for exhaled-breath analysis: A review. *Meas. Sci. Technol.* **2021**, *32*, 102004. [[CrossRef](#)]
16. Scott, S.M.; James, D.; Ali, Z. Data analysis for electronic nose systems. *Microchim. Acta* **2006**, *156*, 183–207. [[CrossRef](#)]
17. Markom, M.; Shakaff, A.M.; Adom, A.; Ahmad, M.; Hidayat, W.; Abdullah, A.; Fikri, N.A. Intelligent electronic nose system for basal stem rot disease detection. *Comput. Electron. Agric.* **2009**, *66*, 140–146. [[CrossRef](#)]
18. Lozano, J.; Santos, J.; Horrillo, M. Classification of white wine aromas with an electronic nose. *Talanta* **2005**, *67*, 610–616. [[CrossRef](#)] [[PubMed](#)]
19. Lattice Semiconductor. Leveraging FPGA and CPLD Digital Logic to Implement Analog-to-Digital Converters. In A Lattice Semiconductor White Paper February. 2010. Available online: <https://www.embedded.com/leveraging-fpga-and-cpld-digital-logic-to-implement-analog-to-digital-converters/> (accessed on 26 September 2021).
20. Sousa, F.; Mauer, V.; Duarte, N.; Jasinski, R.P.; Pedroni, V.A. Taking advantage of LVDS input buffers to implement sigma-delta A/D converters in FPGAS. In Proceedings of the 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512), Vancouver, BC, Canada, 23–26 May 2004; Volume 1, pp. 3–6.
21. Terasic. DE1-SoC. 2019. Available online: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=205&No=836> (accessed on 10 February 2021).
22. Cupane, M. Application of MOOSY32 eNose to Assess the Effects of Some Post Harvest Treatments on the Quality of ‘Salustiana’ Orange Juice. *J. Biosens. Bioelectron.* **2015**, *6*, 4–7.