



Improvement of The ADC Resolution Based on FPGA Implementation of Interpolating Algorithm

Youssef Kebbat, A Ndaw

► To cite this version:

Youssef Kebbat, A Ndaw. Improvement of The ADC Resolution Based on FPGA Implementation of Interpolating Algorithm. International Journal of New Technology and Research (IJNTR), 2016, 2 (1), pp.100 - 103. insu-01344677

HAL Id: insu-01344677

<https://insu.hal.science/insu-01344677>

Submitted on 1 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improvement of The ADC Resolution Based on FPGA Implementation of Interpolating Algorithm

Y. Kebbati, A. Ndaw

Abstract— This paper exposes a method that gives us the possibility to use a low accuracy Analog-to-Digital Converter (ADC) in high-resolution measurements. We increase the resolution of a 12-bits ADC to 16-bits by adding samples which are calculated using Shannon interpolate algorithm. Thus, the digital signal has high resolution compared to measurements. Specific hardware architecture was developed to implement the algorithm in FPGA. The great advantages of the proposed design are an enhancement of ADC resolution and the continuous time is modeled as a white noise which is generated by the FPGA itself, obviating the need of an external noise source. Results were presented in order to confirm the method

Index Terms— Analog-to-Digital Converter, Interpolation, Shannon signal-reconstruction theory

I. INTRODUCTION

Many applications require measurements using an analog-to-digital converter (ADC). Such applications will have resolution requirements based in the signal's dynamic range, the smallest change in a parameter that must be measured, and the signal-to-noise ratio (SNR). For this reason, many systems employ a higher resolution off-chip ADC. However, the higher the desired accuracy, the higher the ADC cost. Higher ADC accuracy is achieved by designing hardware to quantize the analog signal amplitude into the digital signal with a higher code-word length. Practical ADCs have finite word lengths. To effectively strike a balance between system cost and accuracy, higher conversion accuracy is achieved by calculating new samples. Then, the digital signal is processed in software through an FPGA. This scheme, which process samples and adds additional bits of accuracy to the 12-bit ADC conversion, is explored in this paper.

II. THEORY AND METHODS OVERVIEW

A. Theory

Analog signal amplitude is quantized into digital code words with a finite word length. This process of quantization introduces noise in the signal called "quantization noise". The smaller the word length, the greater the noise introduced. Quantization noise can be reduced by adding more bits into the ADC hardware design. This noise can also be reduced in software by two methods: oversampling the ADC or processing digital signal. The oversampling ADC methods

are well established and well described in literature [1-6]. Processing digital signal and a few associated terms are explained in the following sections.

Voltage resolution of an ADC is defined as the ratio of full scale voltage range to the number of digital levels that are accommodated in that range. It is a measure of the accuracy of the ADC. The higher the resolution, the higher the number of levels accommodated in the voltage range and, consequently, the lower the quantization noise, as shown in equation 1.

$$\text{Voltage resolution} = 1 \text{ LSB value} = \frac{\text{full scale voltage range}}{2^N - 1} \quad (1)$$

where *LSB* is the last significant bit and *N* is the number of bits.

For example, if the full scale measurement voltage range is 0 to 3 Volts, and the ADC bit resolution is 12 bits, then the ADC voltage resolution can be calculated to be 0.7326 mV/bit. This means the conversion of continuous voltages is noise free if the continuous voltage is an integral multiple of the voltage resolution. Any intermediate continuous voltage is rounded off to suit a voltage level that is an integral multiple of the voltage resolution

The measure of the extent to which the signal is corrupted with quantization noise after analog-to-digital conversion is given by the signal-to-quantization noise ratio. Signal-to-Quantization Noise Ratio (*SNRQ*) is defined as the ratio of the root mean square value of the input analog signal to the root mean square value of the quantization noise. The *SNRQ* of an ideal *N*-bit ADC is given by equation 2.

$$SNRQ = 6.02N + 4.77 + 20 \log_{10}(LF) \text{ [dB]} \quad (2)$$

where, *N* is the number of bits or the word length, and *LF* is the loading factor, which is defined as the ratio of the root mean square value of the input analog voltage to the peak ADC input voltage.

When the input analog signal is sinusoidal *LF* = 0.707, then *SNRQ* is given by equation 3.

$$SNRQ\text{-Max} = 6.02N + 4.77 - 3 = 6.02N + 1.77 \text{ [dB]} \quad (3)$$

From equation 3, it is clear that the improvement in the SNR of the ADC is 6.02 dB per bit. The higher the number of bits associated with the ADC, the higher the *SNRQ*. For example, the *SNRQ*-MAX of a 12-bit ADC is 74.01 dB and that of a 16-bit ADC is 98.09 dB. A cost-effective method of improving the resolution of the ADC is developing software to suitably process the converted analog-to-digital signal to achieve the same effect as a higher resolution ADC.

Y. Kebbati, University of Orléans/ LPC2E laboratory in CNRS, Orléans, France, (e-mail: Youssef.kebbati@cnrs-orleans.fr).

A. Ndaw, University of Orléans, Orléans, France, (e-mail: aladjj.ndaw@etu.univ-orleans.fr).

B. Oversampling method

Oversampling is the first method to achieve this aim. The main microcontroller producers sustain a theory based on the number of oversampling bits that you want to obtain. A formula is calculated that involves taking a number of measurements equal to a power of 4, and dividing the whole sum to a power of 2 as shown in equation 4. The power is equal to the number of oversampling bits that you want to obtain [9, 10]

$$\text{Result} = \frac{\sum_{i=1}^{4^N} \text{ADC}_i}{2^N} \quad (4)$$

where ADC is one conversion of maximum 2^x bits, Result is final oversampled result of maximum 2^{x+N} bits, x is converter's implicit number of bits and N is number of oversampling bits.

For example, the analog signal should be oversampled at a rate of 256 times more than the Nyquist rate to achieve the SNR of a 16-bit ADC with a 12-bit ADC. The bloc diagram of the method is shown in Fig. 1.

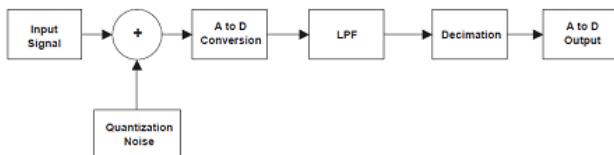


Fig. 1: Diagram of oversampling method

C. Interpolating method

The second method, which is developed in this paper, is to interpolate the signal. The method consists in calculating mathematically intermediate points between original samples. The method is shown in fig. 2. The analog filter is very simple. It allows respecting the initial phase of the signal.



Fig. 2: Diagram of interpolating method

There are a large number of mathematical algorithms to calculate the interpolation of a signal. We can quote: Lagrange, Newton, Spline cubic, Neville, Shannon etc. Shannon algorithm allows to obtain the initial signal $x(t)$ only through n samples [7][8]. If a signal has finite energy, the minimum sampling rate is equal to two samples per period of the highest frequency component of the signal. Specifically, if the highest frequency component of the signal is B hertz, then the signal, $x(t)$, can be recovered from the samples as shown in equation 5.

$$x(t) = \frac{1}{\pi} \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2B}\right) \frac{\sin[\pi(2Bt - n)]}{2Bt - n} \quad (5)$$

The frequency B is also referred to as the signal's bandwidth and, if B is finite, $x(t)$ is said to be bandlimited [11-14].

III. WHY TO CHOOSE SHANNON INTERPOLATION?

To answer this question, we have developed mathematical models in order to compare interpolation algorithms of Lagrange, Newton, Spline cubic, Neville, Hermite and Shannon. Various forms of signals were tested (sinus, saw tooth ...). All algorithms gave a satisfactory answer set apart Neville's algorithm which shows an important error of interpolation as shown in fig. 3.

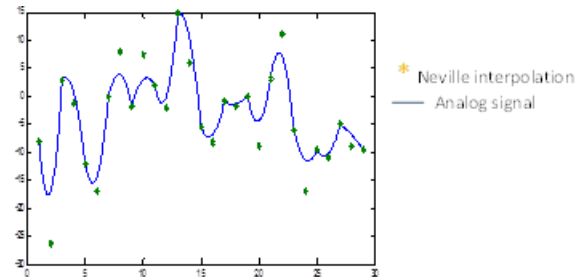


Fig.3: Neville interpolation

Another criterion to choose the best algorithm is the calculation time. In fact, this criterion is important in the case of FPGA integration. The following table 1 shows the calculation time for every algorithm. The results are obtained by software programming on computer.

Algorithms	Calculation time for 12000 samples	Number and type of arithmetic operations
Lagrange	11min and 35,54 seconds	Add: 3, Mult : 6, Div : 3, Subst : 6
Newton	3 min and 50,64 seconds	Add : 2, Mult : 4, Divi : 3, Subst : 8
Spline cubic	12 min and 2,47 seconds	Add : 6, Mult : 10, Div : 4, Subst : 4
Shannon	2 min and 48,54 seconds	Add : 3, Mult : 6, Div : 3, Subst : 3
Neville	5min and 23,12seconds	Add : 3, Mult : 3, Div : 3, Subst : 9
Hermite	5min et 53,45seconds	Add : 4, Mult : 9, Div : 6, Subst : 9

Table 1: Calculation time

According to the table 1, we see that the calculation time depends on the number of multiplication and division. Thus, the Shannon interpolation is the fastest but it is important to take care to *cardinal sine* implementation in FPGA. In fact, *cardinal sine* can be implemented with *Cordic* algorithm but it is cost effective in terms of area and conception time.

IV. FPGA IMPLEMENTATION AND RESULTS

For the implementation, hardware architecture was developed and integrated into *Altera Cyclone V* device. The architecture is based on Multiplier and Accumulator Arithmetic and Logical Unit ALU, barrel Shifter, registers and two Look-Up-Tables LUT as shown in fig. 4. The first LUT was used to implement sine function in order to avoid the use of *Cordic*. The second LUT is used to implement the continuous time. In fact, the continuous time is a random function which works like a white noise. For this design, the FPGA add 40 points between two ADC samples.

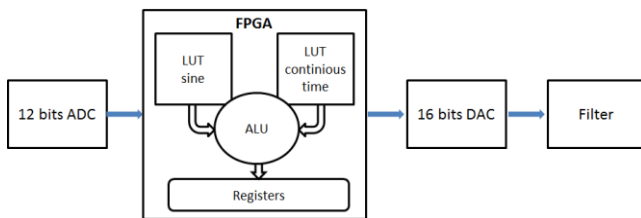
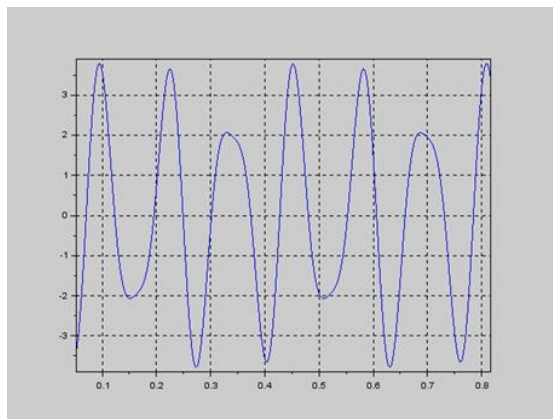
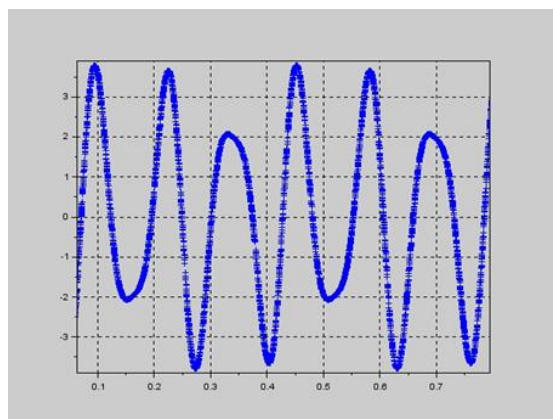


Fig. 4: FPGA implementation

Fig.5 shows respectively (a) the analog signal and (b) the interpolated signal.



(a)

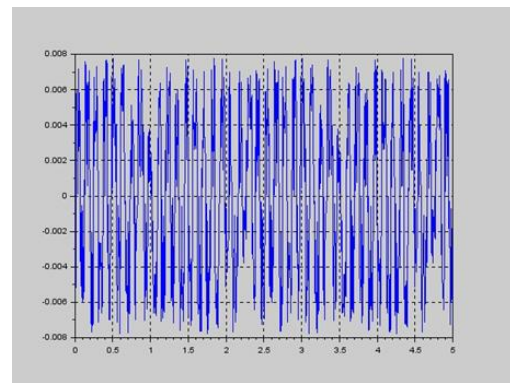


(b)

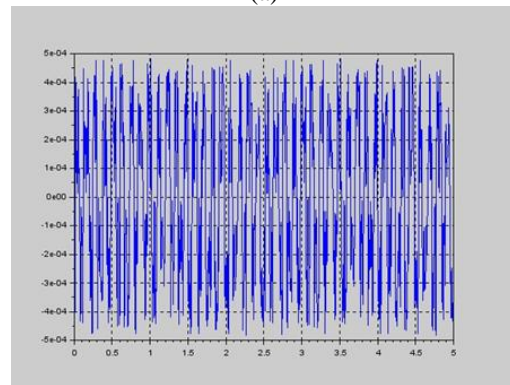
Fig. 5: Analog and Interpolated signals

The error between the analog signal and the ADC is presented in the fig. 6 (a) whereas the fig. 6 (b) shows the error with the 16 bits interpolated signal. As we can see, these results show a reduction of the error with the Shannon interpolation. Indeed, the amplitude of error passed of 0.008 volt for 12 bits in 5.10^{-4} volt for the interpolated signal. So, the resolution of the ADC was improved from 12 bits to 16 bits.

However, this method has some limits. Indeed, the FPGA execution time limits the frequency of the analog signal. Indeed in our case, it was necessary to calculate 40 points between every sample. What implies to have a calculation time 40 times lower than sampling period which, finally, limit the signal bandwidth. Other limit concerns the use of LUT. For the sine function, it is about the step of calculation used.



(a)



(b)

Fig. 6: 12 bits and 16 bits errors

V. CONCLUSION

In this paper, we have presented an FPGA hardware integration of Shannon interpolation algorithm in order to improve 12 bits ADC resolution to 16 bits. A Comparison of different interpolation algorithms shows an adequacy between Shannon algorithm and FPGA integration requirements. In our experiments, an accuracy improvement was seen when the input signal was interpolated by a factor of 40 using a 12-bit ADC and filtered using an analog filter. However, this method has some limits as FPGA execution time which limits the signal bandwidth.

ACKNOWLEDGEMENTS

This work has been conducted in the form of a Master's degree project of A. Ndaw.

REFERENCES

- [1] Microchip technology, "Achieving Higher ADC Resolution Using Oversampling", Microchip datasheet, 2008.
- [2] M. W. Hauser, "Principles of Oversampling A/D Conversion", *J. Audio Engin. Soc.*, vol 39, no 1(2), 1991, pp. 3-26.
- [3] L. Schuchman, "Dither Signals and Their Effect on Quantization Noise", *IEEE Trans. Comm. Techn.*, vol 12, no 4, 1964, pp. 162-165.
- [4] D. Jarman, "A Brief Introduction to Sigma Delta Conversion", Intersil Application Note AN9504, May 1995. Available at :<http://www.intersil.com/data/an/an9504.pdf>
- [5] W. Kester, "ADC Architectures III: Sigma-Delta ADC Basics". Analog Devices, Application note MT22, Available at : <http://www.analog.com/static/imported-files/tutorials/MT-022.pdf>
- [6] L. E. Bengtsson, INTERPOLATION OF MICROCONTROLLER ADC BY SELF-INDUCED DITHERING 1379
- [7] J.W. Goodman, "Introduction to Fourier Optics", Mc Graw-Hill, New York, 1968
- [8] A.J. Jerri, "The Shannon sampling theorem – its various extension and applications: a tutorial review", *Proceeding IEEE*, vol.65, pp.1565-1596, 1977.

- [9] J. M. Madapura, "Achieving Higher ADC Resolution using Oversampling", Microchip Technology Inc., 2008.
- [10] AVR121: Enhancing ADC resolution by oversampling, ATMEL 8-bit AVR Microcontrollers, Application Note.
- [11] D. Slepian "On bandwidth", Proceeding IEEE, vol.64, pp.292-300, 1976
- [12] R. Marks, "Introduction to Shannon sampling and interpolation theory", Book, Springer edition.
- [13] O. BERNAL, « *Conception de Convertisseurs Analogique-Numérique en technologie CMOS basse tension pour chaînes Vidéo CCD Spatiales* », Thèse de doctorat en génie électrique, électronique et télécommunication, L'institut national polytechnique de Toulouse.
- [14] E. Tisserand, J-F. Pautex , P. Schweitzer, « *Analyse et traitement des signaux méthodes et applications au son et à l'image* », édition DUNOD