Compressive sensing in FPGA and microcontroller

S.E. Pinto [#1], L.E. Mendoza [*2], E.G. Flórez [#3]
[#1,2]Department of EEST, [#3]Department of MMI, Pamplona University
[1] luis.mendoza@unipamplona.edu.co
[3] eflorez@unipamplona.edu.co

Abstract—This paper show the hardware implementation of the technique known as Compressive Sensing (CS). CS work develops in a sparse space (dense few). We used to obtain sparse signals Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT). We demostred that speech, electromyography and electrocardiogram signals can be reconstructed from few samples (projection). These projections were physically transmitted to a computer for reconstruction using L1 magic and demostrate that CS projections obtained were correct. The implementation CS showed relevant results, increase in speed and developing new communications. The implementation was performed in a Field Programmable Gate Array (FPGA) SPARTAN 3E and 18F4550 microcontroller. The results showed that is possible the reconstruct signals, breaking Shannon and Nyquist's theory. Furthermore, we concluded that the FPGA implementation is faster and allows higher compression rates than the microcontroller. Finally, we showed using minimization L_1 that DWT, presents better results in sparse conversion.

Keyword- CS, sparse, compressive sensing in hardware, Discrete cosine, Discrete wavelet.

I. INTRODUCTION

Communications and signal processing system has the need to consume the least possible amount of resources. A system is considered efficient and competent if it uses the least amount of resources to obtain the best results. This is why various compression, analysis and information processing techniques are appearing to improve systems performance. CS is one such technique that allows reconstructing a signal with few samples (dense few). Furthermore, the Shannon-Nyquist sampling theorem states that for a perfect reconstruction of a signal it should be sampled using a rate of at least twice the bandwidth [1]. The CS theory explains that, under specific conditions, a sparse signal can be reconstructed using a small number of samples, obtained from a random sub-sampling (then it can be said that, in CS, the number of samples is less than that required for the Shannon-Nyquist theorem) [2].

CS is a very recent field of signal processing. According to the results of Candès, Romberg and Tao [3], and Donoho [4], [5], it is a way to acquire (sparse) compressible signals. In CS, the quality of the reconstruction depends on the signal's compressibility, selecting a reconstruction algorithm, the processing method, and sparse sampling randomness [6]. Currently there are different sparse transformation methods, and these are used depending on the signal's information [7]. Among the most used methods are: discrete cosine transform and discrete wavelet transform and including mixtures. This paper shows the results of CS (compression in FGPA and microconotroller)/reconstruction (in computer) of voice, electromyographic and electrocardiographic signals, the DCT, DFT and DWT sparse conversion methods were used.

There is more to add about the sampling theorem. At times it involves a very high volume of data acquisition and this can lead to a loss of computational cost, because compression is a mandatory step in topics such as storage or data transmission [8]. This explains why more information requires a greater number of samples, and therefore a better compression technique. For example, digital cameras include a growing number of mega pixels (millions of sensors that sample). All acquired data are discarded by the compression algorithm on the inside (in general JPEG or JPEG 2000), which finally produces the picture shown in the camera [9].

To resolve this type of implications, thinking beyond the Shannon-Nyquist theorem is a must. CS can be a useful tool. Proof of this is the work of Richard Baraniuk and Kevin Kelly, from Rice University: the fact of a single-pixel camera that works with CS and reduces the number of samples to acquire an image [7]. On the other hand, as it already has been mentioned, sparse transformations are important. The word "sparse" can mean something poor, limited, short and distributed. But in the context of signals, sparse does not mean limited data. It is a signal for which most values are zero or have small amplitude, and some points or fewer points are of high signal amplitude [10]. If these conditions are met, the signal has the property of compressibility; if the signal has a large number of separate low peaks, it is a high density low signal; and if it has some separated peaks, it is a sign of sparse low density.

Finally, this paper makes an original contribution to CS implementation in hardware, and its validation with voice, electromyographic and ECG signals. The vast majority of papers presented in this area are limited to software implementations, which makes this article a starting point for CS in hardware implementations. L_1 minimization algorithm was used for signal recovery.

II. COMPRESSIVE SENSING

The high compressibility of CS is due to the way the samples are taken. The number of samples, M, is a function of the amount of K-sparse values (non-zero amplitude point) that the signal has, and measuring a signal vector **x** has M samples. In fact, CS is capable of sampling a sparse signal using lower measurements than required by the Nyquist-Shannon theorem. The minimum value of M is twice the number of K-sparse points in the signal, thus M>K<< N. Using M>3K ensures optimal reconstruction. Measurements are made of a sub sampling of the x signal values. To explain this, a vector x ∈ R with length N is defined, which means a discrete sparse x signal, with values zero, and non-zero K values. Mathematically, CS is defined in "(1)", [8], and [11].

$$x = \psi * s \tag{1}$$

Each S_i (i = 1, 2 ... N) value in "(1)" can be obtained from product $\psi_i T_x$. T indicates the transpose, meaning that a sparse signal is rebuildable. Sampling of x is performed without taking N measurements. This is what CS represents in a compressed sampling from few measurements; the result of this operation is stored in a vector with the help of random basis Φ. Measurements are now made by computing the inner product between the x signal and a Φ_i vector arrangement, its result being defined as y_i. Base Φ is also known as encoding base, and it has been designed with the aim of facilitating the reconstruction of a sparse signal. Φ has dimensions N x M, so that the vector y is defined as "(2)":

$$y = \phi * x = \phi * \psi * s = \phi * s \tag{2}$$

In "(2)" is explained how to obtain and use the original vector x or only the K nonzero entries, the $_S$ bases Θ, also with size N x M, reaches the reduction dimension of Ψ, is used with s and has the property of randomness [5], *y* in this work is defined as projections. ***y*** are implicit products, operations/combinations as the x signal K-sparse values. The compression rate during the sampling process is controlled by the user, according to M. The rows of the coding base, testing values of M in different scattered signals, allow finding an optimal balance between a very high compression and a good reconstruction. An image explaining the acquisition of measurements is shown in Fig. 1 [8].

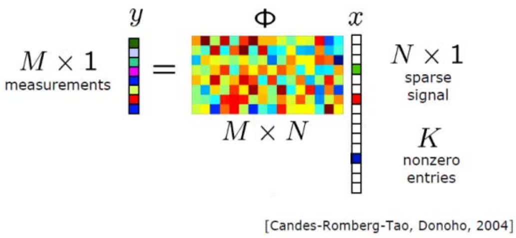

Fig 1. Example of a sparse signal (a). The product of each Φ_i with the column vector x is the value of measurement y_i, so a sampling compression y is stored in the vector. See also that M> 2k.

The reconstruction process is a complex task, which involves the use of the base and the coding vector *y* for a reconstruction of the original signal *x*. If N is the number of measurements able to build a system of equations with only one solution, but as M << N there are many possible solutions (expressed as **x***, which would be the solution vector), now the situation is complex because, again, there is only one solution (the original vector x). A common analogy to explain this is the game Sudoku. Solving a Sudoku is to find an answer through a large field of possibilities, which excludes candidates using some type of property, pattern or norm. In this case the rule is that the number cannot be repeated in any row or column square 3x3. Similarly, in CS it is necessary to use a ruler to discard solutions, satisfying "(3)":

$$y = \phi * x' \tag{3}$$

An effective way is through the norm, a vector property defined by linear algebra as follows [11]:

L_0 norm of vector *x*:
$$\|x\|_0 = \sum_{i=1}^{N} |x_i|^0 \tag{4}$$

L_1 norm of vector *x*:
$$\|x\|_1 = \sum_{i=1}^{N} |x_i|^1 \tag{5}$$

L_2 norm of vector *x*:
$$\|x\|_2 = \sqrt{\sum_{i=1}^{N} |x_i|^2} \tag{6}$$

At present there are many ways to use the rule to discard possible solutions ***x****** and discovering the value and significant position of each point K in vector ***x***, in "(4)" it induces that $0^0 = 0$ and in "(5)" is a way to rebuild the vector values x [7]. The L_1 norm is now a linear optimization problem, which is common in computing, and there are many algorithms to solve it. The objective of this problem is to find x* with a smaller L_1 norm "(5)", since each x* combines K-significant values and it is assumed that an x* satisfies "(5)". With a small norm L_1, the desired solution is of high probability (in each case the value L_1 and vector value L_0 are similar) [7]. Magic L1 and L1 minimization were also used in this paper [13].

III. DEVELOPMENT OF PROJECTIONS IN HARDWARE

In the development of this research, the recording of voice, EMG and ECG signals was performed. The INA 115 instrumentation amplifier was used for the acquisition of ECG and EMG. Then they were converted to sparse and the compression process was performed, as all of the above, at hardware level. The compressed signal or projections were sent to a computer, where the reconstruction process is applied to verify the robustness of each CS signals. CS has made great progress and shown significant results in software. However, this technique does not have even important significance results in hardware level. This was a motivation to perform physical implementation in order to suggest a solution in real time. For the real-time implementation, tests were performed with different programmable logic devices, but satisfactory results were achieved with a Field Programmable Gate Array (FPGA) SPARTAN 3E, and a PIC18F4550 microcontroller. In both cases, the capture of the analog signal and digital conversion was done using the microcontroller's A/D converter, which was also used for sending the compressed signal to the computer by USB port, and thus perform decompression and analysis.

A. Conversion to sparse and restricted isometry property (RIP)

As already mentioned, we used method for sparse conversion such as: Wavelet, Cosine discrete, and Fourier discrete, the signal sparsity was measure using the norm L_1 [10], [11], [11], [16]. The mean value norm L_1, for the wavelet was 8.97, Cosine was 12.3 and Fourier 17.4, in this via we conclude that best sparse representation are wavelet coefficients using 50 cycles. The restricted isometry property (RIP) was utilized for find the optimal matrix in the reconstruction process [17]. The value obtained was of $\delta=0.35$ in speech, $\delta=0.859$ in ECG and $\delta=0.58$ in EMG in the RIP, this result makes evident that the matrix Φ is suitable. These results were obtained in hardware. In previous work we did not have mathematical calculations of the RIP, this show that the results were not very strong. In this paper demonstrated the important of RIP and norm L_1. We in this paper follow the acquisition procedure used in [18]. In this case of implementation using microcontroller obtained M=6, this indicate that optimal value of compression of 40%, and this case FPGA obtained M=25, with value of compression of 50%. The RIP was defined as [1]:

$$(1-\delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\|x\|_2^2 \tag{7}$$

Here *A*, denote the random matrix.

B. With microcontrollers

For the application of CS using microcontroller, it was decided to acquire segments of 40 samples of the input signal, and an output of 24 elements projections. This is, a 40% compression. This decision was made to observe speed in the system's response [17]. Initially, USB port configuration was performed. The implemented random matrix has a size of 24 (rows) * 40 (columns). It is worth mentioning that at software stage, the same random matrix in hardware must be stored, this in order to achieve a successful reconstruction. For the corresponding tests, 24000 samples of the signal (voice, ECG or EMG) were taken, indicating that 600 segments were processed. Let us recall that the computed values are in a sparse space, thus was performed the reverse process which is known as, inverse Fourier transform (IFT), inverse discrete cosine transform (IDCT) and inverse wavelet transform (IWT). And then to get the data in the original space, the reverse process in the pre-conditioning of the signal was performed. In this case, a subtraction was performed, with a constant value of 2.5 volts. Finally, the data concatenation process was performed to obtain the vector of length 24000.

C. With FPGA

The Spartan 3E FPGA card has a lot of features that make it superior to microcontrollers, starting with his memory and speed [19]. The program was implemented in the Xilinx Ise program. Here, 240 segments on 100 samples were taken. That is, 100 possible values of n, and 100 possible values of k, here, n and k are matrix size of implementation in FPGA. Then there is a total of 100*100 = 10.000 possible combinations. To implement the random matrix, a similar 50 x 100 signal was used, here M=50 and N=100 . These values were entered into the program through a table of 10.000 data in a vector3-type signal which is an array of 0 to 10.000, and containing integer values in the range of -10,000 to 10,000. The whole process is done in a single duty cycle of the FPGA card. This gives a great advantage to this device, which will surely be one of the best alternatives for the application of this novel technique in hardware. The process synthesizing in FPGA lasted about six hours. This can be considered a problem, and indicates that there was a large consumption of FPGA hardware resources.

Let's recall that the compressed data are sent via USB to the computer to be reconstructed with the Magic L_1 method and compared to the original signal.

IV. RESULTS

The figure 2, shows the results of MSE in reconstruction process using as sparse conversion DCT. Note how the best results were obtained using the EMG signal with MSE=10.14 in FPGA, for a compression percentage of 90%. However the optimal value M, in this case is equivalent a 50% of value N. In this via the value MSE for compression of signals using CS is 8.13 in ECG, 10.56 in speech and 11.87 EMG in microcontroller and 5.71 in ECG, 4.03 in speech and 4.22 EMG in FPGA.

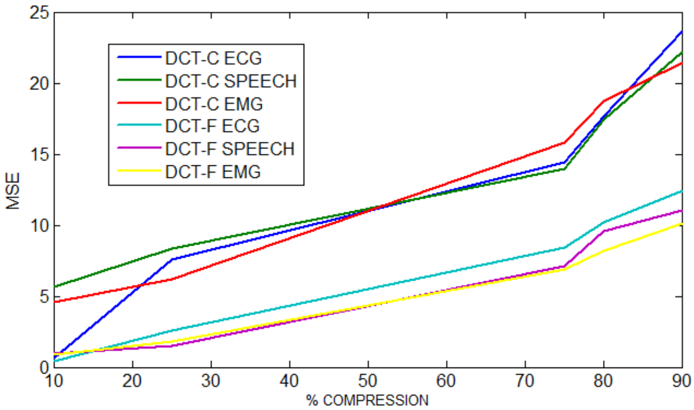

Fig 2. MSE in reconstruction: here DCT-M is DCT in microcontroller and DCT-F is DCT in FPGA.

Likewise, the figure 2 shows the equivalent results of CS implementation in FPGA and microcontroller, using as sparse conversion wavelet and Fourier.

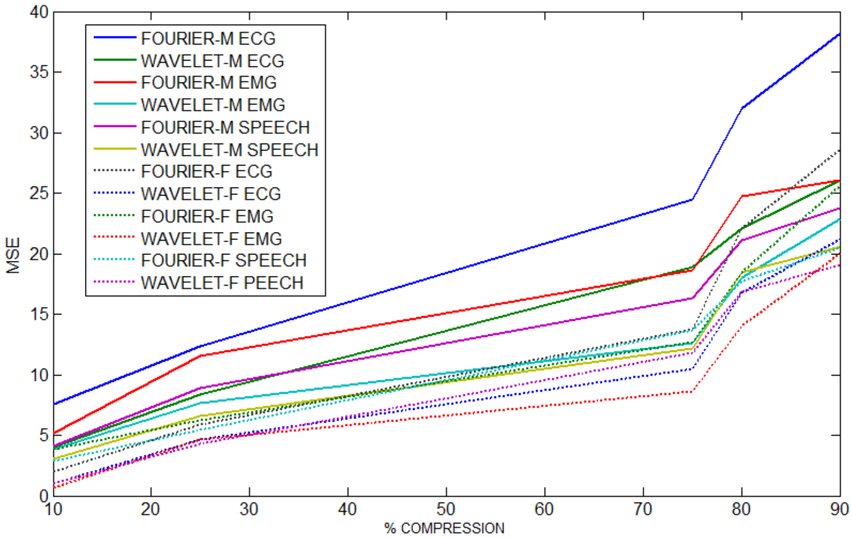

Fig 2. MSE in reconstruction: in microcontroller and FPGA. here the letter M is development in microcontroller and letter F in development in FPGA. Dotted line is development in FPGA with fourier and wavelet.

Note in the figure 2, that results more important was obtained using FPGA and wavelet, why show values de MSE lower that using microcontroller. Using the value M optimal we find that el MSE=4.67 using FPGA in EMG signals for compression factor of 50%, because, if we use more than 50, RIP conditions are not maintained. In this via, we demonstrated that Fourier in microntroller not is advisable. Addition, that execution time in this case is 87.3 second for 24000 samples, while in FPGA using EMG signals with 24.000 samples was obtained a time of 4.01 seconds for EMG, 5.6 seconds for ECG and 3.27 seconds in speech signals. Is relevant mention that the RIP, is

V. CONCLUSION

CS is a technique that allows signal reconstruction without meeting the Shannon and Nyquist theorem. Thus, in the future, it is certain that this technique will revolutionize one of the areas that benefit most from data compression, such as telecommunications. This work showed that CS functions in the field of physiological signals such as speech, ECG and EMG, depending on what is sought in the application. On the other hand, for the proper functioning of CS, selecting a suitable technique to transform the data to sparse is a must. This work has shown that the general best technique for physiological data (speech, EMG and ECG) is the wavelet, because it creates a sparse space ideal for CS requirements. Although the other techniques do work, their responsiveness does not allow reconstructing a signal response as similar to the original signal. It is the opposite case when using wavelet, which got a SME of 5.28 for compression of 50% in EMG signals using an FPGA. As for the speed of the system to apply CS and information shipping, the system that best made this work was FPGA, as it consumed less time and the results were more appropriate. The conclusion is that it is best to take large length threads to apply CS. Furthermore; speech signals can be reconstructed properly after being compressed to a maximum compression rate of 50% in FPGA, and 40% in a microcontroller using RIP. ECG signals can be reconstructed correctly if compressed to a maximum sampling rate of 60.30% in FPGA, and 44.65% in microcontrollers. Similarly, the maximum compression rate that can be applied in EMG signals using FPGA and microcontrollers is 50% and 40% respectively. We demonstrated in this work that it is possible to develop the method of CS compression in hardware.

REFERENCES

[1] E. Candés and M. Walkin, (2008). An introduction to compressive sampling, IEEE signal proccesing magazine. pp. 21-30.
[2] Salman A. M. y Romberg J., (2009). Dynamic Updating for l1 Minimization.
[3] E. Candès, J. Romberg, and T. Tao., (2006). Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information. IEEE Transactions on Information Theory, vol. 52, no. 2.
[4] J. Romberg., (2008). Imaging via compressive sampling. IEEE signal proccesing magazine. pp. 14-20.
[5] D. Donoho., (2006). Compressed Sensing. IEEE Transactions on Information Theory, vol. 52, no. 4.
[6] Boufounus T. P. y Baraniuk G. R.,(2008). 1-Bit Compressive Sensing.
[7] Mackenzie D., (2009). Compressive Sensing Makes Every Pixel Count. What´s Happening in theMathematical Sciences.
[8] Barakat G. y Saliba R., (2008). Compressive Sensing for Multimedia Communications in Wireless Sensor Network.
[9] Hayes B. (2009), The Best Bits. Computing Science, American Scientist, Volume 97, 2009.
[10] Meriño L. M y L.E. Mendoza, (2009). Estudio y Aplicaciones de la Técnica Compressive Sensing (CS) en Señales Sparse.
[11] Baraniuk G. R., (2009). Compressive Sensing.
[12] Boufounus P., Romberg J. y Baraniuk R., (2008). Compressive Sensig: Theory and Apllications
[13] L-1 Magic, (2009). www.l1magic.org.
[14] Aldirmaz, S. Durak Ata. (2011). Compressive Sensing of Linear Frequency Modulated Signals in Fractional Fourier Domains. Signal Processing and Communications Applications (SIU), IEEE 19th Conference. pp.742 – 745.
[15] Wang, Qia; Zeng, Wenjun; Tian, Jun (2014). Compressive sensing based secure multiparty privacy preserving framework for collaborative data-mining and signal processing. Multimedia and Expo (ICME) IEEE International Conference. pp.1 – 6.
[16] Xingsong Hou, Lan Zhang, Chen Gong, Lin Xiao, Jinqiang Sun, Xueming Qian. (2014). SAR image Bayesian compressive sensing exploiting the interscale and intrascale dependencies in directional lifting wavelet transform domain. Neurocomputing. Vol. 133., pp. 358–368.
[17] Stanislaus, J. L. V. M. Mohsenin, T. (2013). Low-complexity FPGA implementation of compressive sensing reconstruction. Computing, Networking and Communications (ICNC), International Conference. pp. 671 – 675.
[18] S. Pinto, L. Mendoza, H. Velandia, A Molina, L. Jaimes (2015). Compressive sensing hardware in signals 1D. Revista TECCIENCIA, pp 5-10.
[19] Karakus, K. Ilgin, H.A. (2012). Implementation of image reconstruction algorithm using compressive sensing in FPGA. Signal Processing and Communications Applications Conference (SIU), 20th. pp. 1 – 4.

AUTHOR PROFILE

L.E. Mendoza. Magister in biomedical engineering, electronics engineer, professor at the University of Pamplona, Colombia. Assistant professor. Teaches telecommunications engineering program.

S.E. Pinto. Magister in control industrials, electronics engineer and electronics engineer, professor at the University of Pamplona, Colombia. Assistant professor. Teaches telecommunications engineering program.

E.G. Flórez. Doctor in mechanical engineering, mechanical engineering, professor at the University of Pamplona, Colombia. Associate professor. Teaches mechanical engineering program.