Compressive Sensing Architecture for Gray Scale Images

Gustavo Gonzalez-Garcia, Alfonso Fernandez-Vazquez,
and Rodolfo Romero-Herrera

School of Computer Engineering, ESCOM
National Polytechnic Institute, IPN
Unidad Profesional Adolfo Lopez Mateos
Av. Juan de Dios Batiz s/n
Esq. Miguel Othon de Mendizabal, Col. Lindavista
Mexico City, 07738, Mexico

Abstract. This paper proposes a compressive sensing architecture for 128 × 128 pixels gray scale images. The proposed architecture is implemented in an FPGA platform. Due to speed and area advantages, the random numbers block generator is implemented using Linear Feedback Shift Register (LFSR) technique. The resulting random matrix is stored in a Random Access Memory (RAM) block. In addition, a second RAM is employed to store the sampled image. We also implement an Universal Asynchronous Receiver Transmitter (UART) to receive and transmit data. Besides the previous blocks, we design an Arithmetic/Logic Unit (ALU), which performs the operations in compressive sensing settings. In this way, a Unit Control (UC) based on a Mealy type state machine directs operations in our architecture. The purpose of the UC is threefold. First, the UC uses the UART to receive the sample image and store it in the corresponding RAM block. Second, the UC directs the matrix multiplication operation to the ALU and obtains the compressed image. Finally, the UART transmits the compressed image to a base station. The main characteristics of our architecture are the following: the maximum frequency of operation is 30 MHz, the power consumption is 37 mW, and the average time processing is 4.5 ms.

Keywords: Compressed Sensing, FPGA, Images, Reconfigurable Architecture.

1 Introduction

Traditional data compression relies on Nyquist theory and is composed by three main steps, i.e., sampling, digital processing, and coding. In this setting, well known data compression standards for audio and image compression can be jpg, mp3, mpg, among others. The main drawbacks of this method are the following: 1) the sampling process generally includes big redundant data, which are removed after digital processing and coding 2) in real world, the signals of interests do not possess band limited as is required by the Nyquist theory, and

A. Gelbukh et al. (Eds.): MICAI 2014, Part I, LNAI 8856, pp. 349–355, 2014.

3) a device with physical limitations imposes distortion in the reconstruction of the original signal. To overcome those disadvantages, a new data acquisition paradigm has been proposed, i.e., compressive sensing. This approach is a universal technique to compress sparse signals [2]. Sparse signals are more common in real word than band limited signals [5]. Compressive sensing allows the reconstruction of the desired signal with high accuracy, which are sampled below the Nyquist criterion [1,5].

In this way, recently, the TamaRISC-CS system has been proposed [2,3]. The proposed system is an application-specific instruction-set processor (ASSP) operating operating in the subthreshold regime. This characteristic allows to accomplish 62X speed-up and 11.6X power savings. This implies that the power consumption is 70nWat 1kHz. Another interesting approach is given in [4], where compressing sensing is applied to biosignal acquisition systems to reduce the data rate and realizes ultra-low-power performance. The proposed system achieves compression factors greater than 16X for ECG and EMG signals. Finally, in [6], the authors suggest the use of compressing sensing for MAGNETIC RESONANCE IMAGING (MRI). In addition, some hardware limitations are discussed in order to successfully apply compressive sensing in MRI applications.

The main goal of this paper is to propose a reconfigurable softcore architecture for data acquisition based on compressive sensing. In order to evaluate the proposed architecture, we implement the corresponding building block into VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language). The architecture is designed such that it can handle gray scale images of size 128 × 28 pixels. Furthermore, we carried out the implementation of the reconfigurable softcore in an FPGA (Field Programmable Gate Arrays).

The rest of the paper is organized as follows. Section 2 reviews the basic theory of compressive sensing. Sections 3 deals with the hardware description of our architecture while Section 4 presents the main results of the proposed approach. Finally, Section 5 concludes the paper.

2 Basic Theory

This section reviews the theory of compressive sensing. Consider a discrete vector signal $\mathbf{x}$, which can be expressed by means of a redundant representation or frame, i.e.,

$$\mathbf{x} = \sum_{i=1}^{N} s_i \psi_i, \tag{1}$$

where s_i, for $i = 1, \ldots, N$, are the frame coefficients, ψ_i is the frame i, and N is the number of frames [1,5]. Alternatively, equation (1) can be rewritten as [1,5]

$$\mathbf{x} = \boldsymbol{\Psi}\mathbf{s}, \tag{2}$$

where $\boldsymbol{\Psi}$ is the resulting frame matrix of size $N \times N$, that is, $\boldsymbol{\Psi} = [\psi_1 \psi_2 \cdots \psi_N]$, and $\mathbf{s}$ is a vector of coefficients s_i, i.e., $\mathbf{s} = [s_1 s_2 \cdots s_N]^{\mathrm{T}}$. The signal $\mathbf{x}$ is K-sparse if it is a linear combination of only K vectors ψ_i for $K < N$, that is, only K of the s_i coefficients in (1) are nonzero and $(N - K)$ are zero.

Consequently, in compressive sensing, the compressed signal is obtained from K measurements, which are performed as

$$\mathbf{y} = \mathbf{\Phi x}, \tag{3}$$

where $\mathbf{\Phi}$ is the measurement matrix with size $K \times N$ and $\mathbf{y}$ with size $K \times 1$ is the resulting compressed vector. The compression ration is therefore K/N. Substituting (2) into (3), we have

$$\mathbf{y} = \mathbf{\Gamma s}, \tag{4}$$

where $\mathbf{\Gamma} = \mathbf{\Phi \Psi}$ is called random sensing matrix with size $K \times N$. Usually, each entry for the matrix is a random variable with either uniform or normal distribution and variance $2/K$.

This paper deals with the implementation of the compressive sensing based on equations (3) and (4). The proposed reconfigurable architecture is described in Section 3.

3 Proposed Reconfigurable Architecture

3.1 Building Blocks

The proposed architecture is an specific purpose softcore. The system is based on a RISC architecture of 8 bits. Additionally, an Universal Asynchronous Receiver Transmitter (UART) is implemented to receive and transmit data. The implemented control unit is based on a Mealy state machine, which perform the compressive sensing algorithm. Figure 1 shows the main building blocks of the architecture, that is, arithmetic-logic unit (ALU), image memory, frame memory, control unit, multiplier block, and UART block.

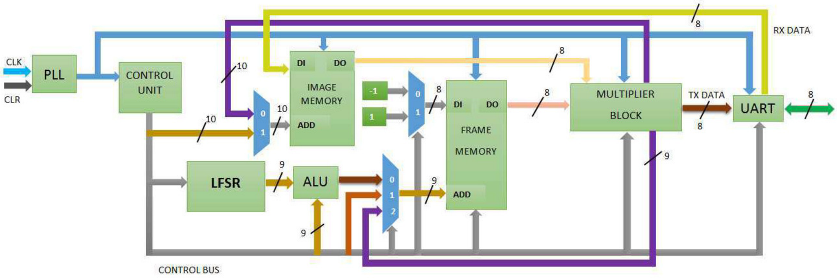

Fig. 1. Main building blocks of the proposed architecture

In the following we discuss in detail the main building blocks of the architecture.

3.2 ALU

The ALU performs basic operations like additions, subtractions, XOR, AND, NADN, OR, and NOR. The considered ALU is synthesized using look-ahead (anticipated carry) in VHDL. This characteristic improves the speed of the architecture.

3.3 Image Memory

The image memory is a $(16K \times 8)$ RAM (Random Access Memory). The main objective of this RAM is to store the raw gray scale image. The size of the image is 128×128 pixels. Each pixel is represented by 8 bits word.

3.4 Frame Memory

The image memory is a $(8K \times 8)$ RAM, which stores the measurement matrix $\mathbf{\Phi}$. In order to simplify the computation of the matrix multiplication in equation (4), we follow the ideas in [2,3], where the entries of the random sensing matrix can be either 1 or -1. To do that, we first set to the value 1 the RAM memory. As a second step, we generate pseudo random numbers, which contain the positions of the RAM memory where the value 1 is substituted by 1. In this setting, the matrix multiplication operation is reduced to summations and subtractions operations.

3.5 Multiplication Block

In this section of the softcore, we implement a multiplication block, which consists of multiplier blocks and adders. The main task of this block is to perform the matrix multiplication $\mathbf{\Psi x}$, i.e., each row of the matrix $\mathbf{\Phi}$ is multiplied by the column vector $\mathbf{x}$. Subsequently, the corresponding compressed coefficients are obtained by adding the resulting vector. The compressed vector is stored in a RAM memory and is transmitted by the UART block.

3.6 UART Block

We also implement an UART protocol to transmit or to receive data from a second device. The data can be the raw data or the compressed data.

3.7 Unit Control

This is the main building block in our softcore. This block is based on a Mealy state machine, which is composed by six states shown in Figure 2. The ASM (Algorithmic State Machine) chart for the implementation of the states is described in Section 3.9.

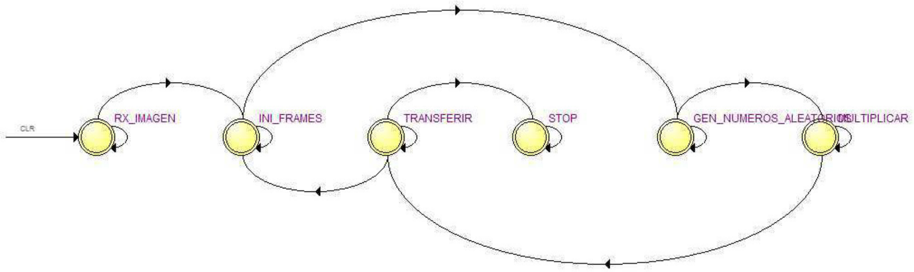

Fig. 2. State diagram

3.8 Random Number Generator

At this stage, the main idea is to maximize the performance of the proposed architecture and to implement a dedicated block to generate the random numbers. Therefore, in order to improve the speed and area consumption, the random numbers block generator is implemented using Linear Feedback Shift Register (LFSR) technique.

3.9 ASM Chart for Compressive Sensing

To develop an algorithm for compressive sensing, we propose an ASM chart, which is illustrated in Figure 3. In the first step, we reset registers and memories. The following step store the random matrix $\mathbf{\Phi}$. The random numbers are generated and are stored in a 64×128 RAM memory as is shown in Figure 2. In this way, the UART block receives the gray scale image $\mathbf{x}$ from a second device using serial protocol. The compressed samples are obtained by means of the product $\mathbf{\Phi x}$ and the compressed vector $\mathbf{y}$ is stored in RAM memory. Finally, the vector $\mathbf{y}$ is transmitted to the second device. In this process, the original images with 2^{14} samples is reduced to 1000 samples. This means, we have a compression ratio of 0.061.

4 Results

We implement the proposed architecture into VHDL. The synthesis of each building block is obtained from Altera Quartus II Web Edition ver. 13.0 and the employed hardware is an FPGA Cyclone IV EEP4CE115F29C7. The simulation is carried out using the environment Altera ModelSim 10.1. Table 1 shows the main results. The first row provides the total combinational functions, which directly impacts the total area. In addition, the remaining parameters are obtained under the following conditions, the voltage source is 1200 mV and the working temperature is 85°C.

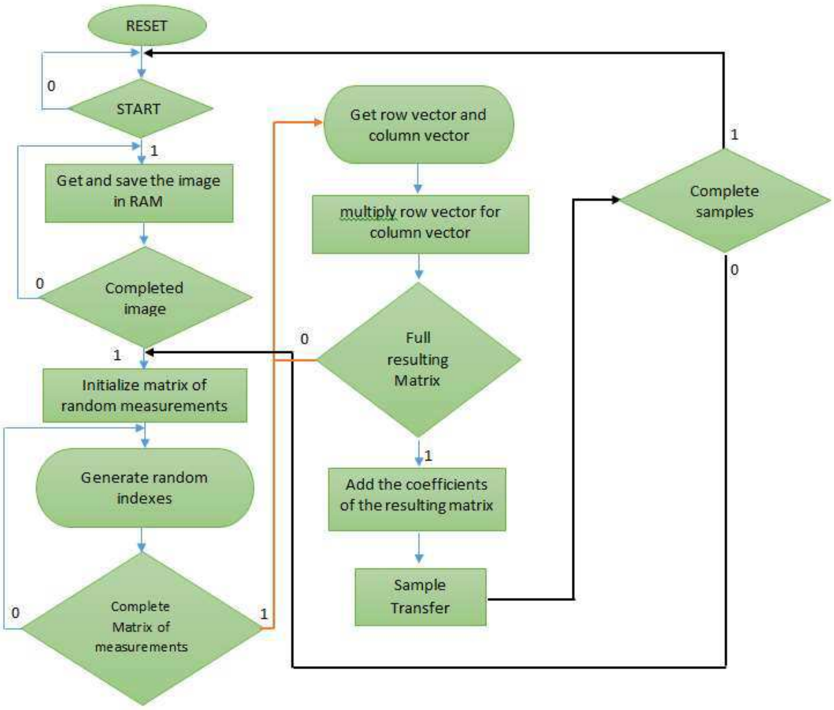
RESET
START
0
1
Get and save the image in RAM
Completed image
0
1
Initialize matrix of random measurements
Generate random indexes
Complete Matrix of measurements
0
1
Get row vector and column vector
multiply row vector for column vector
Full resulting Matrix
0
1
Add the coefficients of the resulting matrix
Sample Transfer
Complete samples
1
0

Fig. 3. ASM chart

Table 1. Synthesis results of the proposed architecture and optimization analysis

Total combinational functions	13355
Dedicated logic registers	12014
Embedded Multiplier elements	0
Maximum Frequency	30 MHz
Core Static Thermal Power Dissipation	102 mW
Thermal Power Dissipation	37 mw
Execution time of the algorithm	4.5 ms

5 Conclusions

This paper presents an specific purpose architecture for compressive sensing and its VHDL implementation. The proposed architecture is an alternative solution to the existing architectures in the literature. As a difference with that solutions, our approach is a simple solution since no silicon synthesis is necessary. We describe the main building blocks along with the proposed algorithm for compressive sensing. In this approach, the number of compressed samples is 1000 for an image of size 128 × 128 pixels. This suggests that our architecture for compressive sensing is a useful tool for data acquisition.

Acknowledgments. This work was supported by SIP-IPN project 20141476.

References

1. Baraniuk, R.G.: Compressive sensing [lecture notes]. IEEE Signal Processing Magazine 24(4), 118–121 (2007)
2. Constantin, J., Dogan, A., Andersson, O., Meinerzhagen, P., Rodrigues, J., Atienza, D., Burg, A.: TamaRISC-CS: An ultra-low-power application-specific processor for compressed sensing. In: 2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC), pp. 159–164 (October 2012)
3. Constantin, J., Dogan, A., Andersson, O., Meinerzhagen, P., Rodrigues, J., Atienza, D., Burg, A.: An ultra-low-power application-specific processor with sub-v_T memories for compressed sensing. In: Burg, A., Coṣkun, A., Guthaus, M., Katkoori, S., Reis, R. (eds.) VLSI-Soc 2012. IFIP Advances in Information and Communication Technology, vol. 418, pp. 88–106. Springer, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-45073-0_5
4. Dixon, A., Allstot, E., Gangopadhyay, D., Allstot, D.: Compressed sensing system considerations for ECG and EMG wireless biosensors. IEEE Transactions on Biomedical Circuits and Systems 6(2), 156–166 (2012)
5. Donoho, D.: Compressed sensing. IEEE Transactions on Information Theory 52(4), 1289–1306 (2006)
6. Lustig, M., Donoho, D., Santos, J., Pauly, J.: Compressed sensing MRI. IEEE Signal Processing Magazine 25(2), 72–82 (2008)