International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Pipelined Image Reconstruction of SAR Radar Based on Orthogonal Matching Pursuit using FPGA Implementation

Eslam Ashraf, Ashraf A. M. Khalaf, Sara M. Hassan

Abstract: ***Synthetic Aperture Radar (SAR) imaging methods is an interesting field in remote sensing. Nowadays, the Orthogonal Matching Pursuit Compressive Sensing (OMP) algorithm is applied on the reconstruction of data that are produced by SAR. The OMP is iterative algorithm that needs high time consumption and processing delay. This issue is considered one of the main problems that face the designers of the SAR systems that how to speed up the performance of the system to be more applicable. This paper provides an applicable pipelined processing technique for the OMP compressive sensing algorithm to speed up the compression and reconstruction of SAR Image Data. Based on the goal of this paper, it is possible to reduce the time processing of the OMP as every clock a new process will be started and it is not required to wait the certain process to be finished. The good-resolution images of the SAR are used for mapping, identification and other applications. This article defines the compressive sensing algorithms and it also discusses the design, analysis of the proposed pipelined processing method for one of the CS algorithms to reduce the consumed time using FPGA implementation. Moreover, the paper includes the implementation of the both normal processing and the proposed pipelined processing for the used algorithm. Finally, a comparison between the two algorithms is presented to evaluate the performance.***

Keywords: SAR; Compression; reconstruction; CS; Algorithm and Pipelined.

I. INTRODUCTION

Sar performance depends on the accuracy of the imaging sensors that are able to multiply imaging modes and resolutions. These options together with the wide swath requirement provide a huge amount of data that should be processed to generate images with high resolution. The low computational resources and the steadily increasing the resolution of SAR so it is not able to process data on board. Hence, it must be stored using a certain storage media or to be transmitted to the ground stations where the processing steps are performed. The big amount of data is constrained by the storage and transmission channel capabilities. To solve this problem, many solutions were made to provide a certain compression algorithm for compressing the raw data image [1, 2].

Image that are provided by SAR are different in nature compared to the optical images. This difference appears in some points like: SAR Image is large in size, it may contain complex pixels with 32 bit and large dimensions, but the optical image is 16 bit real pixel with smaller dimensions. Also, the SAR image entropy is higher than the entropy of optical image; SAR image has information on the low frequency bands and high frequency bands, but the optical image is low-pass with noise in a high frequency band, and finally SAR image is larger of dynamic range than optical. Because of these difference, the traditional techniques of image compression are not applicable to SAR images. Hence , many efforts have been exerted to compress the SAR image data [3].

Many researchers made significant researches to update the data compression algorithms of the SAR systems,

They hope to reach the best solution to decrease the size of the data and increase the performance. In the next paragraphs, we will discuss the most common previous works, the relevant researches and the related works [4].

Discrete Cosine Transform (DCT) is an algorithm that has been used also for the SAR image compression; it is based on transforming the image to another domain that represents the images ideally. This is done by using the uncorrelated coefficients as some terms can be quantized to zero. This technique is not suitable for SAR systems because of the high noise, the high entropy, the high frequency, the dynamic range and the blocking of some artifacts in SAR images [5, 6].

Discrete Wavelet Transform (DWT) is another technique that is relevant. It is based on the idea of using a filter bank connected by a down sampling procedure. DWT is an accurate technique compared to DCT, because it works in the frequency domain and time domain, it provides a less dynamic range and provides a built in despeckling noise [2, 5, 7].

Manuscript published on January 30, 2020.
* Correspondence Author
Eslam Ashraf*, Department of Electronics and Communications, Faculty of Engineering, Minia University, Minia, Egypt.
E-mail: eng.eslamashraff@gmail.com
Ashraf A. M. Khalaf, Department of Electronics and Communications, Faculty of Engineering, Minia University, Minia, Egypt.
E-mail: ashkhalaf@yahoo.com
Sara M. Hassan, Department of Electronics and Communications, Faculty of Engineering, Modern Academy, Cairo, Egypt.
E-mail: sara.hassan@eng.modern-academy.edu.eg

Vector Quantization (VQ) is also is a common algorithm that is used for SAR image compression and one of the first techniques used to SAR radar systems compression due to its simplicity. It is based on the idea of dividing the image into small ones, it leads to reduce the dynamic and makes the pixels use a few number of bits. Accordingly, VQ provides a better performance compared to the scalar quantization [1, 8].

Compressive sensing (CS) is a new algorithm that provides a suitable method for the compressing of SAR raw data image [9]. Nowadays, the compressive sensing is considered one of the newest methods used for the SAR image compression due to its ability to overcome the SAR systems requirements [10]. It can reduce data volume and consumed electricity power. There are three types of the compressive sensing algorithms, Complex Approximate Message Passing (CAMP), Orthogonal Matching Pursuit (OMP) and L_1-regularized Least Square method [11, 12].

Compressive sensing is a fast-growing field so most of researches in the SAR systems data compression focus on the compressive sensing and all the efforts are exerted in the direction of updating CS to be more suitable for the SAR systems due to the high signal to noise ratio in addition to the simplicity of the computational operation [9].

As mentioned before, until now; many researches were made in the compressive sensing; for example, the research of the image reconstruction by Tegan H. Emerson, Timothy Doster and Colin Olson (2018), the sparse subspace clustering by Yanxi Chen, Gen Li and Yuantao Gu in 2018 and finally the new analysis to support the recovery by Haifeng Li and Jinming Wen (2019) [1, 13].

In this paper, we will use the Orthogonal Matching Pursuit (OMP) because it gives a higher performance than the Complex Approximate Message Passing (CAMP) and its implementation is less complex than the l_1-regularized Least Square method [14]. Depending on the compressive sensing for the compression and decompression, we will find that the main problem in the decompression process, because of the compression process is a simple matrices multiplication; but in the decompression, the process is different because it consists of many iterations depending of the numbers of elements of the used matrices [15]. As the numbers of elements increase the amount of iterations increased [9, 16], it leads to the consumption of more time clocks as the numbers of iterations increase. This issue is considered one of the main problems that face the designers of the SAR systems that how to speed up the performance of the system to be more applicable [5, 8, 9].

In the present work, we will concentrate on the previous problem; this paper is considered one of the researches of the SAR image compression, it provides an applicable pipelined method for the OMP compressive sensing algorithm to speed up the compression and decompression of SAR Image Data. Based on the goal of this paper, it is possible to reduce the time clocks of the OMP processing steps. This article also discusses the design, analysis and implementation of OMP compressive sensing pipelined method to reduce the consumed numbers of clocks and speed up the processes.

This paper is organized as follows: section II describes OMP algorithm and its steps, section III shows the methodology we used to reach the paper goal for speeding up the compression and decompression processes by using the OMP pipelined process. Section IV discusses the hardware implementation for both the normal OMP and the pipelined OMP, then the simulation and results are presented in section V. Finally, the conclusion is explained in section VI.

II. THE ORTHOGONAL MATCHING PURSUIT

This section discusses the compression and the OMP reconstruction of the image. The image which is expressed as a vector x ϵ R^N can be reconstructed with a reduced number of samples compared to the Nyquist rate provided that it is a sparse in some basis $f = \psi_x$ where $\psi \epsilon R^{NxN}$ is a matrix whose columns are the basis vector and $x \epsilon R^N$ is a vector with a small number of non-zero component k, where k<<N [2, 16, 17].

Assume the input image pixels coefficients is the x vector and the Φ represents the sensing matrix that has dimensions N x M size; we may get the y matrix that represents the compressed vector, with M dimension [1] as shown in Fig. 1. We can recover the original x vector if 2K<M < N. The concept of the OMP is based on finding the atom that gives the most contribution to the output matrix y then which atom is the next and so on as shown in Fig. 2. This process needs N iteration, where N is the number of atoms in **Φ** [18, 19]. To understand the recovery process, assume there is a sensing matrix which dimensions are 2 x 4, then this matrix has 4 atoms which represent the four columns of this matrix. The OMP reconstruction algorithm will find the atom which has the highest contribution to the output matrix y, then the next atom and so on to reach the final atom [20, 21].

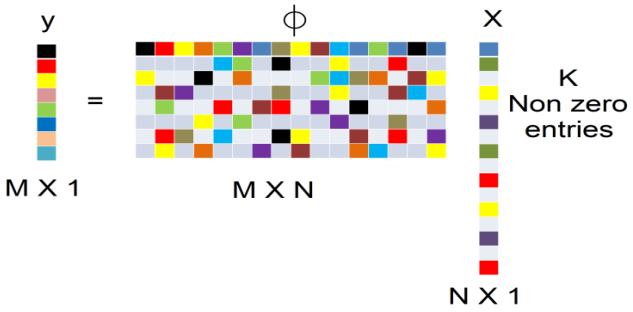

Fig. 1. Example of Sparse Signal

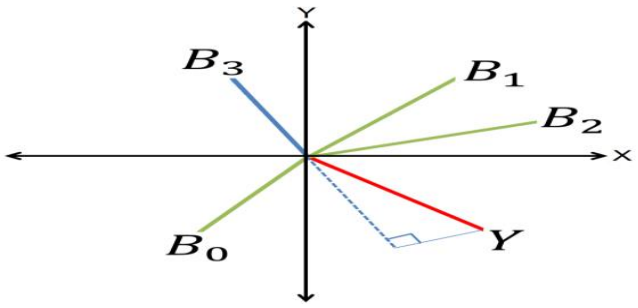

Fig. 2. Interpretation of residue

In Fig. 2, the red line represents the output of the compression process, the green lines and the blue line represent the basis on the sensing matrix, the continuous blue line represents the selected basis, the dashed blue line represents its contribution and finally the perpendicular line on the dashed line represents the residue. The reconstruction will be explained in the next paragraph:

In the first iteration, the residue is equaled to the output, then the measurement column is found as eq (1).

$$B_J = \arg max \left| < y, B_{j.} > \right| for\ B_{j.} \epsilon B \quad (1)$$

B_J represents the basis and y represents the output, in this step, the column that has most of the contribution will be selected by the dot product of the residue and the selected basis. Then, the next step is to find the residue which is calculated using the next eq (2).

$$r = y - X_J y' \quad (2)$$

r represents the residue while y' represents the weight which can be calculated as eq (3).

$$y' = \arg min \left\| y - X_J y' \right\| for\ y' \epsilon R \quad (3)$$

This represents our expectation of s_j in S if S is always 1 or 0, we may enforce that $y' = 1$ Then, we repeat for the t rounds and stop the residual which is small; accordingly, the residue represents the expected matrix X.

In this paper, we used a certain algorithm to compress any 64 x 256 pixel image that is converted to one dimensional matrix with 16384 image coefficients. This matrix is too large to adapt with the number of slices in the available FPGA and to reduce the mathematical operations. Instead of using very huge sensing matrix, we used 2 x 4 matrix which process 4 by 4 elements which will produce 2 elements in the compression and 2 by 2 which will produce 4 elements in the decompression. This technique reduces the mathematical operations and avoids using huge sensing matrix.

The previous algorithm in the normal case consumes 16384 time clock in the compression and decompression processing steps.

In the normal process, the compression starts after the previous compression process ends, the normal processing is shown in Fig. 3. In the compression process, the input is entered serially and the first output is appeared after 4 clocks as shown element_1 and element_2 represent the compressed data for the previous 4 input but in the decompression process, the input_1 and input_2 represent the compressed data and the output1, output2, output3 and output4 represent the data after reconstruction. As explained the compression and the decompression process consume 4 clocks to compress 4 elements and to reconstruct them.

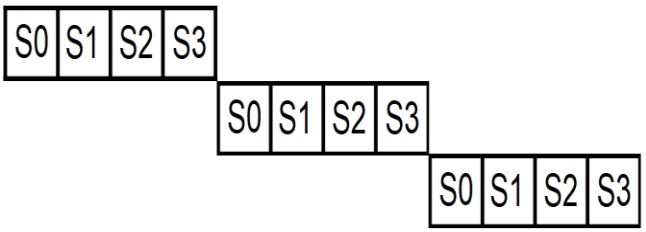

Fig. 3. The Normal processing

This normal algorithm makes the process very slow and consumes large numbers of time clocks that makes the system not suitable for SAR systems. Also, the design must have a huge memory to store the input of the next processes; so it uses more resources and hardware component and leads to a high cost.

III. PROPOSED PIPELINED ALGORITHM

The pipelined processing algorithm is a suitable solution for the normal processing problems because each new clock has a new process that will be started.

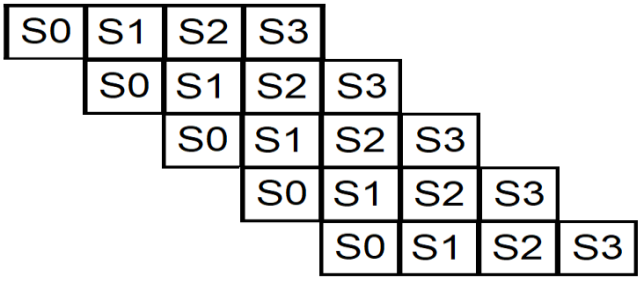

Fig. 4. The pipelined processing

As shown in Fig. 4, the pipelined algorithm is faster than the normal; as every clock anew process will be started, this processing makes it more suitable for SAR Radar applications.

The method that we used in the normal compression and normal decompression will be applied again in the pipelined algorithm. We will use 2 x 4 matrix and the compression will be performed 4 by 4 elements which will produce 2 elements in the compression and 2 x 2 which will produce 4 elements in the decompression [19, 22].

In the pipelined process, the compression starts before the last compression process finished, as shown in Fig. 3. In the compression process, the input is entered serially and the output is appeared after 1 clock as shown element_1 and element_2 represent the compressed data for the previous 4 inputs but in the decompression process, the input_1 and input_2 represent the compressed data and the output1, output2, output3 and output4 represent the data after reconstruction. As seen in Fig. 16 and Fig. 17 the compression and the decompression process consume 1 clock to compress 4 elements and to reconstruct them [5, 17].

The pipelined algorithm reduces the numbers of clock to 4099 so it makes the compression and decompression processing faster and more reliable for the SAR radar and suitable for the other application [4].

To verify the results, we used the Matlab simulation program to simulate the output of the compression and the decompression process. Then, we compared these results to the output of the ISim in addition to the consumed time of the two technique, as shown in Fig.13 which shows the consumed time against the output, the pipelined technique consumed less time than the normal technique and the output of the first appeared before than the normal one [3, 19].

IV. HARDWARE IMPLEMENTATION

In this article, the normal algorithm and the pipelined one in the compression and decompression are implemented to compare between the two algorithms. The Design is performed using Xilinx Spartan 3E FPGA starter kit (XC3S1000E-4C in VQ100 package), by writing a VHDL code on Xilinx ISE14.7 and simulated by using the ISim simulator.

The schematic implementation of the compression in the normal processing is shown in Fig. 5.

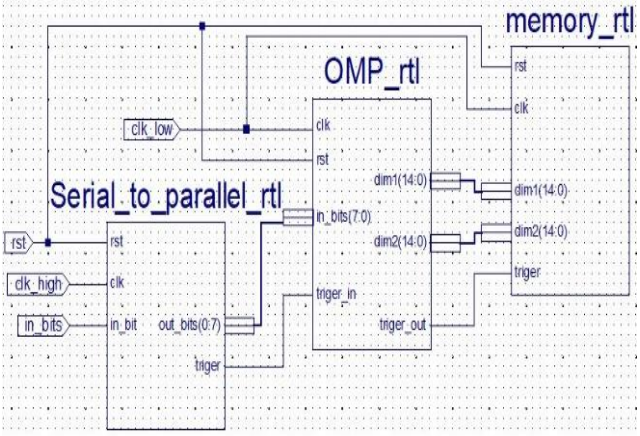

Fig. 5. Schematic implementation of normal compression

Fig 6 shows the schematic implementation of the decompression in normal processing.

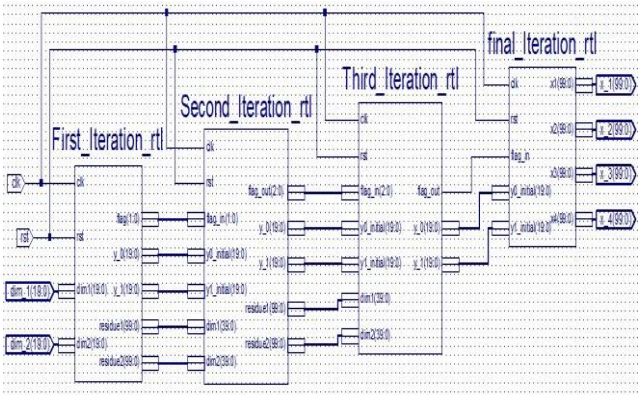

Fig 6. Schematic implementation of normal decompression

In this paper, we facilitate the simulation using the image for the simulation is converted to binary coefficients using Matlab simulation program which will produce a text file that contains 16384 coefficients binary numbers. After that, we simulate the FPGA implementation of the compression, the simulation results will be saved in a text file. Accordingly, we can use this file in the decompression simulation. Finally, the decompression results will be saved also in a text file to view the image that is equivalent to these decompressed coefficients to compare between the original and the original one after the compression and decompression processes. The pipelined algorithm will speed the processing because it consumes only 4099 clocks, the schematic implementation of the pipelined compression is shown in Fig. 7.

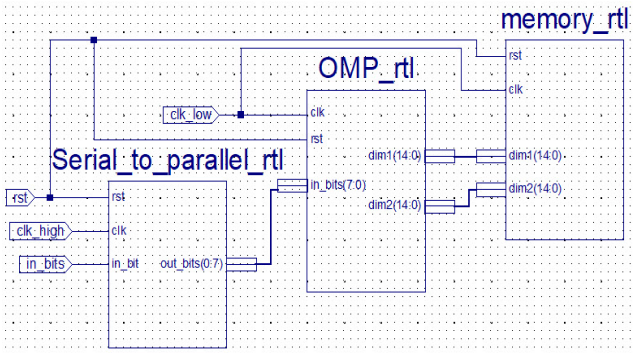

Fig. 7. Schematic implementation of pipelined compression

Finally the schematic implementation of pipelined decompression is shown in Fig. 8.

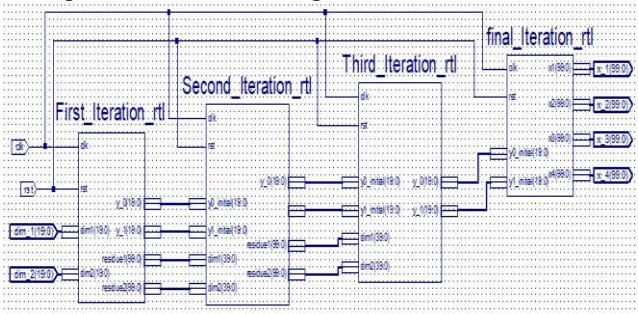

Fig. 8. Schematic implementation of pipelined decompression

The ISim simulation and the results of the pipelined implementation are shown in Fig. 15. and Fig. 16, the improved performance will appeared after the reconstruction and reviewing the image using Matlab simulation program.

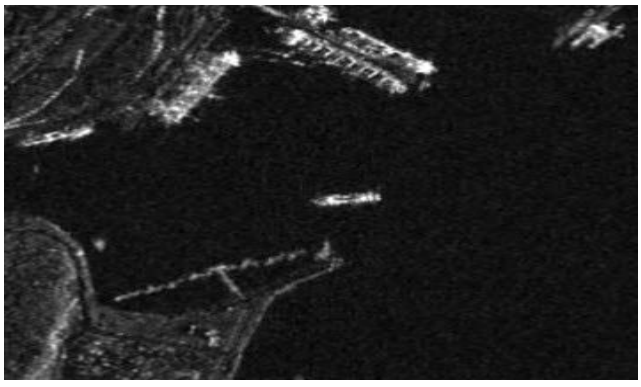

Fig. 9. The original image

Fig. 9 shows the original image that we used to prove out concept which is converted first to a text file using Matlab simulation program and then was compressed using out implementation and Fig. 10 shows the same image after the reconstruction process that was converted from a text file to image using Matlab after being passed by the compression and the decompression using the pipelined technique.

As we explained the pipelined technique reconstructs the image almost as same as the original one. There are small differences in the color of the image because of the estimation of the OMP. Fig. 11 shows the average reconstruction error between the original image and reconstructed one [12, 23].

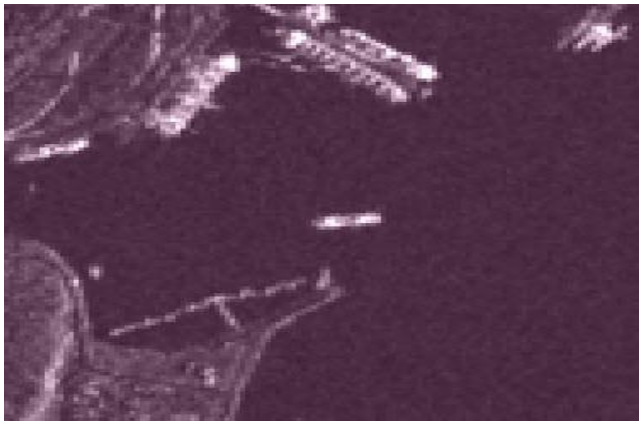

Fig. 10. The reconstructed image

The updated process needs more resources than the normal, as appeared in the device utilization summary for the implementation of both the normal process and the pipelined as shown in Table I and Table II.

The improved performance is shown using the chipscope (Xilinx Chipscope Pro tool provides virtual I/O directly and logic analyzer into the design to check the internal signals or nodes) to check the design after the download on the used FPGA kit to check the behavior. The chipscope signals are shown in Fig.17 and Fig. 18.

V. SIMULATIONS AND RESULTS

Fig. 11 shows the reconstruction rate. The reconstruction rate will decrease as the sparsity level increases because increasing the sparsity samples will require more time for the reconstruction that leads to low construction rate.

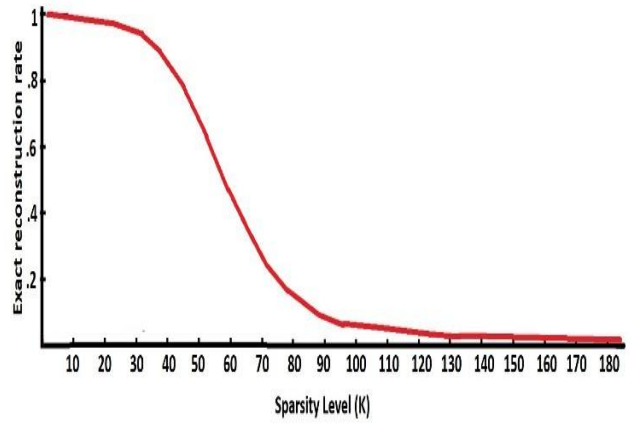

Fig. 11. The reconstruction error between the original and reconstructed image

Fig. 12 shows the Mean Square Error (MSE) of the reconstruction of compressed image. The MSE of the OMP compressive sensing algorithm decreases in every iteration until it reaches a stage it is tolerated which is calculated according to the Restricted Isometry property.

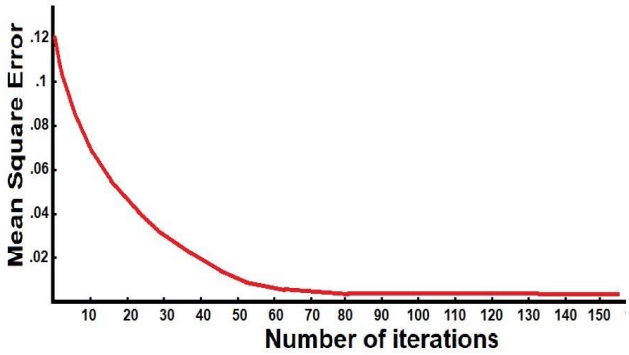

Fig. 12. The Mean Square Error (MSE)

Fig. 13 shows the consumed number of clocks for both the normal and pipelined. As shown, there is a big difference between the usage of the normal processing and the pipelined, the pipelined consume smaller numbers of clocks that enhances the performance speed.

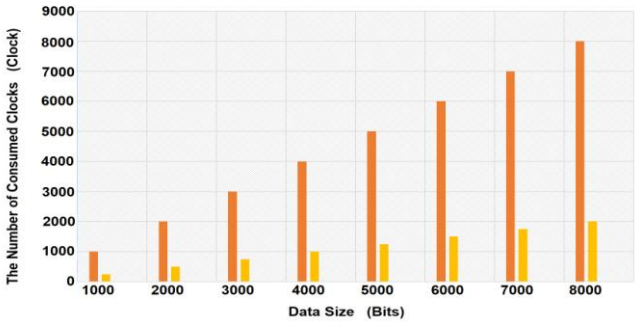

Fig. 13. The consumed number of clocks for both the normal and pipelined

Fig. 14 shows the ISim simulation of the normal compression processing. As shown in Fig. 14, the above yellow triangular shows the number of consumed clocks and the below yellow triangular shows the input and output date at each clock. The two output compressed elements will be calculated after 4 clocks as mentioned before every 4 elements will be compressed into 2 elements. As shown in Fig. 14, the normal compression needs 4 clocks to be finished. The original coefficients consist of four elements and each coefficients consists of 16 bits so the total bits of the process is 64 bits that is equivalent to 8 bytes. But the compressed data consist of two elements and each elements consists of 20 bits so the total bits is 40 bits that is equivalent to 5 bytes after the compression.

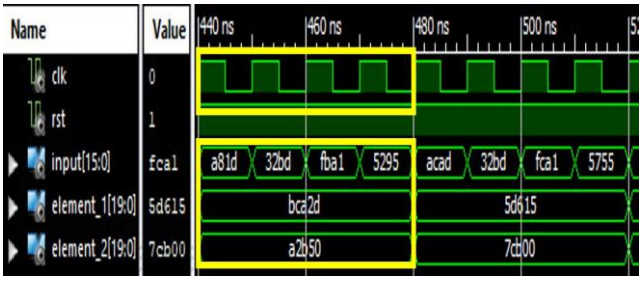

Fig.14. The ISim simulation of the normal compression

Fig. 15 shows the normal reconstruction. The above yellow triangular shows the number of consumed clocks and the below yellow triangular shows the input and output date at each clock. The four output recovered elements will be calculated after 4 clocks as mentioned before every 2 elements will be recovered to 4 elements using 4 iterations. As shown in Fig. 15, the normal reconstruction needs 4 clocks to be finished.

Fig. 15. The ISim simulation of the normal decompression

Fig.16 shows the ISim simulation of the pipelined compression processing. As shown in Fig. 16, the above yellow triangular shows the number of consumed clocks and the below yellow triangular shows the input and output date at each clock. The two output compressed elements will be calculated after 1 clock as mentioned before every 4 elements will be compressed into 2 elements. As shown in Fig. 16, the pipelined compression will start a new compression processing without needing to wait the end of the previous process.

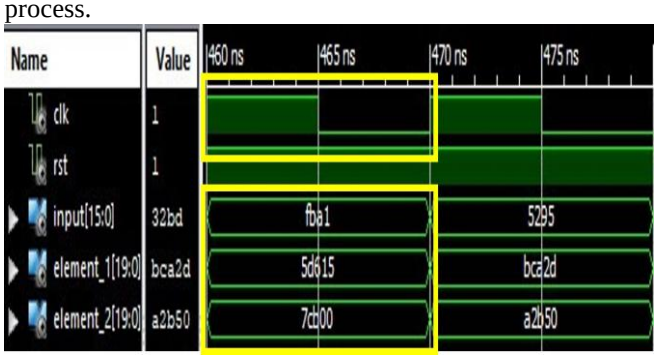

Fig. 16. The ISim simulation of the Pipelined compression

Fig. 17. shows the pipelined reconstruction. The above yellow triangular shows the number of consumed clocks and the below yellow triangular shows the input and output date at each clock. The four output recovered elements will be calculated after 1 clock as mentioned before every 2 elements will be recovered to 4 elements using 4 iterations. As shown in Fig. 17, the pipelined reconstruction will start a new recovering processing without needing to wait the end of the previous process.

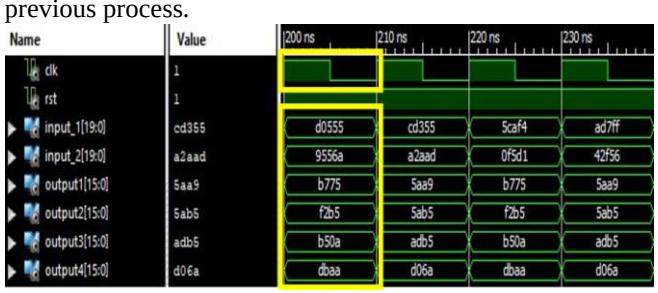

Fig. 17. The ISim simulation of the Pipelined reconstruction

Fig.18 shows two elements after the compression process. This Fig. represent the real data after the downloading of the project on the used kit. These data was sampled form the output pins. As shown in Fig. 18, the compressed data consists of 20 bits.

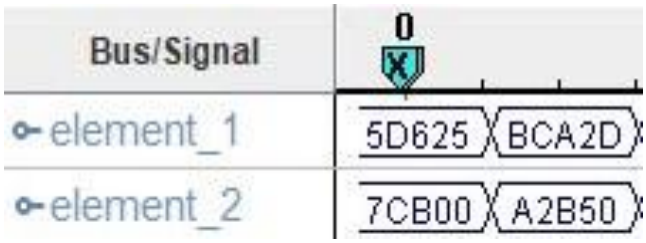

Fig. 18. The ChipScope signals in the compression case

Fig.19 shows four elements after the compression process. This Fig. represent the real data after the downloading of the project on the used kit. These data was sampled form the output pins. As shown in Fig. 19, the reconstructed data consists of 16 bits.

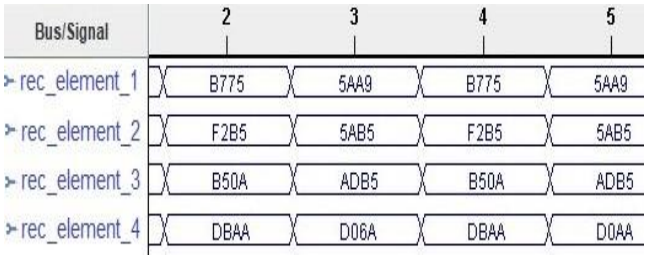

Fig. 19. The ChipScope signals in the reconstruction case

Table I shows the device utilization summary for the normal process and Table II shows the summary for the pipelined process. As shown in table 1 and table 2, the purposed pipelined process needs more resources than the normal.

Table I: the device utilization summary for Normal process

Logic Utilization	Used
Number of Slice Flip Flops	1235
Number of 4 Inputs LUTs	320
Number of Occupied Slices	2123
Number of Slices Containing only Related Logic`	1548
Total Number of 4 Input LUTs	2123
Number of Bonded IOBs	43

Table II: the device utilization summary for PIPELINED process

Logic Utilization	Used
Number of Slice Flip Flops	1546
Number of 4 Inputs LUTs	360
Number of Occupied Slices	2586
Number of Slices Containing only Related Logic`	1958
Total Number of 4 Input LUTs	2586
Number of Bonded IOBs	74

VI. CONCLUSION

This paper explained a new algorithm that can be used to speed the compression and decompression process of image SAR system. The pipelined processing makes the consumed time lesser than the time consumed during the processing of the normal algorithm. The algorithm depends on the concept of making the processes performed at the same time instead of waiting every process to be complete.

The implementation of both, the normal process and the pipelined are being provided to compare between the two algorithms. The improved performance was proved by using a real data, the selected image is converted to coefficients by using Matlab and these coefficients are compressed by using our implementation. Also, the compressed data is reconstructed then converted to image to compare between the original image and the reconstructed one to verify the results. Finally, the simulation for the consumed numbers of clocks against various data with several lengths was performed and another simulation was performed to measure the error that appears in the reconstructed image.

REFERENCES

1. H. M. Makrani, 'Storage and Memory Characterization of Data Intensive Workloads for Bare Metal Cloud', arXiv preprint arXiv:1805.08332, 2018.
2. J. A. Tropp and A. C. Gilbert, 'Signal recovery from random measurements via orthogonal matching pursuit', IEEE Transactions on information theory, 2007, vol. 53, no. 12, pp. 4655-4666.
3. J. Susaki, M. Tsujino, and T. Anahara, 'Fusion of different frequency SAR images for DInSAR-based land subsidence monitoring', in 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017, pp. 945-948.
4. E. J. Candes, 'The restricted isometry property and its implications for compressed sensing', Comptes rendus mathematique, 2008, vol. 346, no. 9-10, pp. 589-592.
5. Y. Arjoune, N. Kaabouch, H. E. Ghazi, and A. Tamtaoui, 'Compressive sensing: Performance comparison of sparse recovery algorithms', in 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), 2017, pp. 1-7.
6. N. Shahbazi, A. Abbasfar, and M. Jabbarian-Jahromi, 'Efficient two-dimensional compressive sensing in MIMO radar', EURASIP Journal on Advances in Signal Processing, March 01 2017, vol. 2017, no. 1, pp. 23.
7. Y. Shi, X. x. Zhu, and R. Bamler, 'Nonlocal Compressive Sensing-Based SAR Tomography', IEEE Transactions on Geoscience and Remote Sensing, 2019, vol. 57, no. 5, pp. 3015-3024.
8. F. Salahdine and H. El Ghazi, 'A real time spectrum scanning technique based on compressive sensing for cognitive radio networks', in 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), 2017, pp. 506-511.
9. H. Sayadi, H. M. Makrani, O. Randive, S. M. PD, S. Rafatirad, and H. Homayoun, 'Customized machine learning-based hardware-assisted malware detection in embedded devices', in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018, pp. 1685-1688.
10. Y. Wang, H. Bai, and Y. Zhao, 'Image Reconstruction from Patch Compressive Sensing Measurements', in 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), 2018, pp. 1-4.
11. G. Z. Karabulut and A. Yongacoglu, 'Sparse channel estimation using orthogonal matching pursuit algorithm', in IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004, 2004, vol. 6, pp. 3880-3884.
12. S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, 'A method for large-scale l1-regularized least squares', IEEE Journal on Selected Topics in Signal Processing, 2007, vol. 1, no. 4, pp. 606-617.
13. J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, 'Striving for simplicity: The all convolutional net', arXiv preprint arXiv:1412.6806, 2014.
14. J. Li, Y. Fu, G. Li, and Z. Liu, 'Remote Sensing Image Compression in Visible/Near-Infrared Range Using Heterogeneous Compressive Sensing', IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2018, vol. 11, no. 12, pp. 4932-4938.
15. S. Kunis and H. Rauhut, 'Random sampling of sparse trigonometric polynomials, ii. orthogonal matching pursuit versus basis pursuit', Foundations of Computational Mathematics, 2008, vol. 8, no. 6, pp. 737-763.
16. K. Neshatpour, H. M. Mokrani, A. Sasan, H. Ghasemzadeh, S. Rafatirad, and H. Homayoun, 'Architectural considerations for FPGA acceleration of Machine Learning Applications in MapReduce', in Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, 2018, pp. 89-96.
17. Q. Wu, L. Ni, and D. He, 'Improved Image Reconstruction Based on Block Compressed Sensing', in 2012 Spring Congress on Engineering and Technology, 2012, pp. 1-4.
18. M. a. Y. M. R. Bhanumurthy, 'SAR Data Processing with Range Cell Migration', International Journal of Engineering Science and Technology, 2011, pp.1-11.
19. H. Yin and L. Wang, 'Effect of Compressive Sensing Star Image on Attitude Determination', in 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2018, vol. 02, pp. 8-11.
20. J. N. Laska and R. G. Baraniuk, 'Regime Change: Bit-Depth Versus Measurement-Rate in Compressive Sensing', IEEE Transactions on Signal Processing, 2012, vol. 60, no. 7, pp. 3496-3505.
21. H. B. Sharanabasaveshwara and S. Herur, 'Designing of Sensing Matrix for Compressive Sensing and Reconstruction', in 2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC), 2018, pp. 1-5.
22. N. Sharma and R. Pandey, 'Compressive Sensing and Recovery of Image Using Uniform Block Sparsity', in 2018 3rd International Conference for Convergence in Technology (I2CT), 2018, pp. 1-5.
23. H. B. L. Ma, M. Zhang, Y. Zhao IEICE Transactions 99-A(11), 'Edge-Based Adaptive Samplingfor Image Block Compressive Sensing', 2016, pp. 2095-2098.

AUTHORS PROFILE

Eslam Ashraf, received the BE degree of Electronics and Communications in 2009. He received master's degree in Electronics and Communications in 2018 from Arab Academy for Science, Technology and Maritime Transport. Currently, He is working towards PhD degree at Minia University at Electronics and Communications Department. E-mail: eng.eslamashraff@gmail.com

Ashraf A. M. Khalaf, (PhD) received his B.Sc. and M.Sc. degrees in electrical engineering from Minia University, Egypt, in 1989 and 1994 respectively. He received his Ph.D. in electrical engineering from Graduate School of Natural Science and Technology, Kanazawa University, Japan, in March 2000. He is currently associate professor at electronics and communications engineering Department, Faculty of Eng., Minia University, Egypt. E-mail: ashkhalaf@yahoo.com

Sara M. Hassan, received the B.Sc. degree in Electrical Engineering from Modern academy for engineering and technology, Cairo, Egypt, in 2007, the M. Sc. degree from Ain Shams University, Cairo, Egypt, in 2013, and the Ph.D. from Ain Shams University, Cairo, Egypt, in 2017. She is currently a staff member at Modern academy for engineering and technology, Cairo, Egypt. Her fields of interest include electrical, electronics, design and implementation for communication systems. E-mail: Sara.Hassan@eng.modern-academy.edu.eg