An FPGA Based Implementation of Forward Automatic Order Selection Processor

التنفيذ باستخدام مصفوفة البوابات المبرمجة لمعالج إعادة التوجيه أوتوماتيكي لإحصائية الكشف

Ali Hadi Abdul Wahid
Technical College - Basrah

Waleed Kalid A.
University of Al-Mustansiriya - Baghdad

An FPGA Based Implementation of Forward Automatic Order Selection Processor

التنفيذ باستخدام مصفوفة البوابات المبرمجة لمعالج إعادة التوجيه أوتوماتيكي لإحصائية الكشف

Ali Hadi Abdul Wahid
Technical College - Basrah

Waleed Kalid A.
University of Al-Mustansiriya - Baghdad

Abstract :

In this study, a new Constant False Alarm Rate processor is investigated with proposed process termed as Forward Automatic Order Selection Ordered Statistics Detector (FAOSO), which does not require any prior information about the number of interfering targets. The proposed architecture has been implemented on a Field Programmable Gate Array (FPGA). After simulating the HDL model of processor, it is implemented using ISE Navigator 6.3i software. The target device for implementation is XC2v300e-6fg456 Virtex-II device .The implemented processor using this technique has been tested using Monte Carlo simulation in presence of eight interfering targets. However, the hardware synthesis results and timing analysis are reported and the implementation process and the study shows clearly that how this digital tool is excellent due to its high reliability and flexibility.

Keywords: CFAR, FPGA, Target Detection, Radar

1. INTRODUCTION

The extraction of targets from signals is a complex task due to the uncontrolled and noisy environmental conditions. However, using the adaptive signal processing techniques to remove noise enhance detection targets in many cases. In radar applications, amplitude is used backscattering of the radar signal to detect the target and usually assumed that the high magnitude of the

backscattering radar signal comes from the targets .Since the background is not uniform amplitude backscattering fluctuates due to background noise, required adjustment scheme to extract targets according to the threshold of varying signal and maintain constant false alarm rate [1].

The processor is called detection that can maintain a steady fixed P_{fa} false alarm rate (*CFAR*), and used to extract goals from the background in noisy environments in the areas of application [2].

The *CFAR* detectors have been used widely in radar signal processing applications to detect the targets from noisy background, was proposed by Gini et al. A typical *CFAR* processor is shown in Figure 1. The input signals are set serially in a shift register. The content of the cells surrounding the cell under test (X_0) are processed by a CFAR processor to get the adaptive threshold T . Then (X_0) is compared with T to make the decision. The cell under test is declared as a target if its value exceeds the threshold value [1].

The early *CFAR* detector called cell-averaging *CFAR* (*CA-CFAR*) was proposed by Finn and Johnson [2]. In the detector, the adaptive threshold is the arithmetic mean of its reference cells. From many *CFAR* algorithms developed, Barkat [3] categorized *CFAR* algorithm into four models. First model is defined to describe situations where there is a transition in the clutter power distribution. The greatest-of-selection logic in cell averaging constant false-alarm rate detector (*GO-CFAR*) proposed by Hansen and Sawyers [4] and the smallest-of selection logic in cell averaging constant false-alarm rate detector (*SO-CFAR*) by Trunk [5] falls into this first model.

Second model is defined to describe situations where the clutter background is composed of homogeneous white Gaussian noise plus interfering targets. Some examples in this category are the censored mean level detector (*CMLD*) by Rickard and Dillard [6], the trimmed mean level *CFAR* (*TM-CFAR*) detector

by Gandhi and Kassam [7], and order-statistic CFAR (*OS-CFAR*) by Rohling [8]. When the number of interfering targets is not known a priori, Barkat et al. [9] proposed the generalized censored mean level detector (*GCMLD*), in which the number of interfering targets is determined and their corresponding samples are then sampled.

The third model describes the most general case in which there is not only a transition in the clutter power distribution, but also interfering targets. Himonas and Barkat [10] proposed the generalized two-level censored mean level detector (*GTLCMLD*), which uses an automatic censoring algorithm of the unwanted samples when both interfering targets and extended clutter are present in the reference window of the cell under test. Khalighi and Bastani [11] presented another variation called the *AEXGO-LOG* processor.

The fourth model deal with non-Gaussian clutter distribution. The log-normal, Weibull, gamma, and Kdistribu- tion has been used to represent envelope-detected non-Gaussian clutter distribution. Some works in *CFAR* detection for Weibull clutter have been reported in some literatures. In [12], Ravid et al. developed the maximum-likelihood *CFAR* (*ML-CFAR*) and the optimal Weibull*CFAR* (*OW-CFAR*) is proposed in [13]. Automatic censoring algorithm also has been developed for non-Gaussian clutter distribution [9-10][14].

However, only few hardware implementation of these *CFAR* was researched. Some examples of *CFAR* algorithms hardware implementation have been presented in [15-17] using parallel/pipeline processing for Max, Min, and Cell-Average (*CA*) *CFAR* algorithms. *OS-CFAR* was implemented using parallel structure in [18]. In [19], *CA-CFAR* and *OS-CFAR* are combined and implemented in *FPGA*. *TM-CFAR* was implemented in [20]. All of these implementation were for simple *CFAR* algorithms suitable for Gaussian distribution type of clutter.

An automatic censoring *CFAR* detector called Automatic Censored Cell Averaging (*ACCA*) *CFAR* was designed by Alsuwailem et al. [21] as a simply *IP* core without flexible communication with others radar system peripherals. Winkler et al. [22] used System on Chip (*SoC*) with reconfigurable processor inside for an automotive radar sensor.

System-on-Chip (*SoC*) architecture becomes an attractive solution for development of pipelined and real-time *CFAR* processor. The architecture is known for its flexibility, its less power consumption, and its high reliability. Field Programmable Gate Array (*FPGA*) technology has made *SoC* fabrication faster and easier.

The rest of the paper is organized as follows. Section 1 on the theoretical background of *CFAR*. Section 2 analyses the data for *CFAR* algorithm and the details of the proposed hardware architecture. In section 3Displays the*FPGA* implementation and experimental results, and provides a brief discussion on the improvements in performance . Finally, Section 4 provides conclusions.

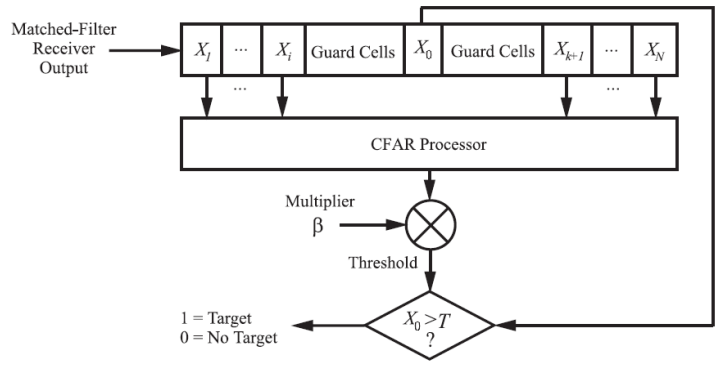

Figure (1) Block diagram of a typical CFAR algorithm

2. THE PROPOSED CFAR ALGORITHM

The block diagram of the proposed Processor is depicted in Figure 2. The radar echo is received and the square law detected range samples are sent to a window register of length $N + 1$, the

statistic *Z* is obtained from the noise power formed by processing the contents of *N* reference cells surrounding the cell under test (*CUT*).The radar outputs $X_i : i = 0,1,.., X_N$ are stored in a tapped delay line. The cell with the subscript 0 is the cell under test, where it contains the signal which should be detected as a target or not. The last *N* surrounding cells are the auxiliary cells used to construct the *CFAR*threshold[24].

In this paper, it is assumed that the random variables of the reference window are independent and identically distributed (*iid*) and are governed by an exponential distribution with probability density function (*pdf*)[23]:

$$f(x) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}} \qquad x \geq 0 \qquad \text{----------(1)}$$

and the distribution function (*df*):

$$F(x) = 1 - e^{-\lambda} \quad x \geq 0 \qquad \text{----------(2)}$$

Under the null hypothesis H_0 of no target in the cell under test, λ is the total background noise power, which is denoted by μ.Under hypothesis H_1 in presence of a target, λ is $\lambda = \mu(1+ S)$, where *S* is the average signal to noise ratio (*SNR*) of the target. We assume that a Swerling*I* mode for the radar returns from the target and Gaussian statistics for the noise. Therefore, for the cell under test, the value of λ is $\lambda = \mu$, under hypothesis H_0 and $\lambda = \mu(1 + S)$, under hypothesis H_1.The assumption of an exponential distribution is justified for the square law detector in the case of normally distributed noise in the video range.

The proposed detector consists of five fundamental steps. These steps are performed dynamically by using a suitable set of ranked cells to estimate the unknown background level and set the adaptive threshold accordingly. This Processor does not require any prior information about the clutter parameters nor the number of interfering targets. The procedure first ranks the outputs of all reference range cells in ascending order according to their magnitudes as follows[24]:

*Step 1 :*Sorting the samples of the reference window;

$X(1) \leq X(2) \leq \ldots \leq X(k) \leq \ldots \leq X(N)$ ---------- (3)

*Step 2 :*Computing the corresponding Information Theoretic Criteria (ITC) for the sorted window ;

*Step 3 :*Estimating the Optimal Order Estimation (K_{opt}) ;

*Step4:*Selecting the corresponding scale factor and computing the threshold level.

$T Z = T X(K_{opt})$---------- (4)

Step 5:The cell under test is compared with the derived threshold and a decision is made according to the tests:

$H_1 : X_0 \geq T X(K_{opt})$ ----------(5)
$H_0 : X_0 < T X(K_{opt})$ ----------(6)

Hypothesis H_1 denotes the presence of target in the test cell, while H_0 denotes there is no target. The threshold multiplier T is defined according to the estimated number of interfering targets.

The proposed Processor requires the estimation of the interfering targets number to set the corresponding threshold. The scale factor is selected with respect to the estimated interfering targets number from a pre-organized lookup table. The proposed algorithm consists of detecting the corrupted reference cells and adjusting the *OS-CFAR* detector parameters to minimize the *CFAR* loss. Both steps are performed dynamically by using a suitable set of ranked cells to estimate the unknown background level and set the adaptive thresholds accordingly. The effectiveness of the (*FAOSO*) algorithm is assessed by computing, using Monte Carlo simulations. The obtained performances are compared to the classical *OS-CFAR* detectors.

The proposed approach proceeds as follow. First, we sort the reference window samples in increasing order. Then, we apply

the information theoretic criteria, *ITC(k),* for each sample. We determine the minimum of the *ITC* and the corresponding sample order. The obtained order represents the estimated number of interfering targets. The *ITC* is defined as [7][25]:

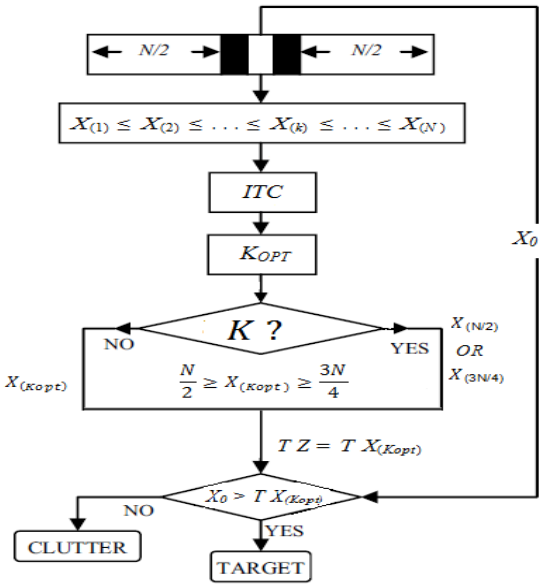

Figure (2) The flowchart for procedures of the CFAR processor.

$$AIC(k) = -2(N - k)N\, ln(\frac{G(\lambda k+1),\ldots\lambda N}{A(\lambda k+1),\ldots\lambda N}) + 2k(2N - k) \quad \text{-----(7)}$$

Where: $\lambda_1 \ge \lambda_2 \ldots \ge \lambda_N$ denote in our case the samples of the reference window, N the number of samples, and *G* and *A* denote respectively, the geometric and the arithmetic means of their arguments. The number of interfering targets is determined as the value for which the *ITC* criterion is minimized. The position of this minimum, K_{opt}, indicates the first interfering target order in the ordered reference window[25].

3. CFAR IMPLEMENTATION AND PERFORMANCE ANALYSIS:

Detector comprises five main units*: I/O* interface, and a shift register, sorting unit, the unit estimate the optimalrrangement,

and comparison, as shown in Figure 2. Given the total length of the shift register as :

$$L=N+G+1 \quad \text{---------- (8)}$$

Where: (N) Reference cells, (G) Guard cells, the Test cell (X_0)

The shift register output data are sent the value of N reference cells in parallel manner to the sorting module. The test cell is delayed from shift Register for a few clocks until the threshold value is calculated from the value. In this work, the sorting circuit is based on a parallel bubble-sort that implemented over an array of compare-swap units[24].After sorting sequentially to these cells N, subject to some sort data to the automatic Oversight mechanism. This mechanism determining the threshold value for determining cell test as a target or not. The decision was by comparing the threshold value with the value of the test cell.

The probabilities of detection of the *OS-CFAR* for $K=12$, and *FAOSO* for $K = K_{opt}$, are obtained by *Monte Carlo* simulations in presence of eight interfering targets. Figure (5) shows the detection probabilities of the *FAOSO*, the OS-CFAR detectors for $P_{fa} =10^{-6}$ and $N = 32$. It is noted that higher values of N yield better detection probabilities and the *FAOSO* Processor.

The corresponding detection probabilities are plotted in Figure 3. It can be seen that the *FAOSO* yields the best detection. This is due to the performance degradation of the *OS-CFAR* Processor because the number of the interfering targets exceeds the upper limit which is four in this case, and on the other hand the *CA-CFAR* Processor presents the worst performance in presence of interfering targets. The *OR-CFAR* Processor remains subject to the performances of the *CA-CFAR* and the *OS-CFAR* Processors separately[24].

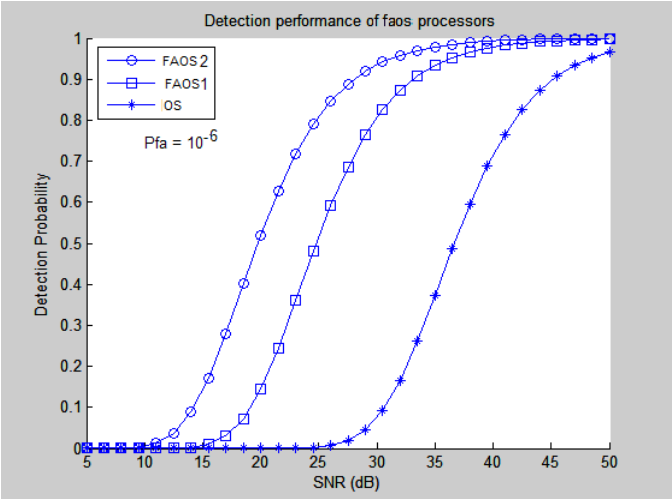

Figure (5) Detection probabilities for $N = 32$ and $P_{fa} = 10^{-6}$.

Table (1) summarizes the statistics for the use of *FPGA* hardware resources and timely performance after synthesis and *FPGA*. For example, at the time of the implementation of the architecture for the performance of *CFAR* processing in radar-based applications is 140 milliseconds on a 4096 × 4096 data samples, using 32 and 8-cell reference and Guard [1].

The proposed architecture was modeled using the *VHDL* Hardware Description Language and synthesized with Xilinx *ISE* targeted for a*XC2v300e-6fg456Virtex-II* device.

Table (1) summarizes the *FPGA* hardware resource utilization and timing performance. The default configuration of the CFAR processor uses 12-bit for data, 32 reference cells and 8 guard cells which is a common configuration used for most radar-based applications with a good performance-accuracy trade-off [1]. The internal temporal data in the accumulator uses 18-bit precision established for the worst case.

Table 1: The statistics for the FPGA hardware resource utilization and timing performance

Synthesis summary for the CFAR processor targeted for (XC2v300e-6fg456) Virtex-II device.	
Number of Slices	3454
Number of Flip Flops	2329
Number of 4-Input LUTs	6290
Clock frequency	165 MHz.

4. CONCLUSIONS

In this work, and an efficient implementation of hardware processors *CFAR* adaptive signal processing and target detection. The new processor does not require any prior information about the number of targets to intervene. Is determined by the number of samples just to minimize interference from *ITC*.

For performance comparison, consider the results presented in [25], where a *CFAR* processor was implemented on a *TMS320C6203 DSP* device. The processing time obtained for this *DSP* implementation was 420 milliseconds to process a sample data set of the same size.

It is important to mention that the implementation in [25] doesnot perform the sortingoperation. The architecture presented in [23] that reports athroughput of 840 *MOPS*, the proposed architecture was synthesized for a *Virtex II* device. The throughput achieved using 32 reference and 4 guard cells is 2,520 *MOPS* which is three times better than the throughput reported in [23] using the same number of cells, the architecture has a maximum operation frequency of 165 MHz .

The architecture performance is over 3 times faster than the required theoretical processing time of 2.5 seconds. This system has the advantages of fast, and flexible. This work has shown is also well suited for modern design, including hardware efficiently for the implementation of target detection .

The proposed architecture was modelled using the *VHDL* Hardware Description Language and synthesized with *XilinxISE* Navigator *6.3i* targeted for a*XC2v300e-6fg456 Virtex-II* device.

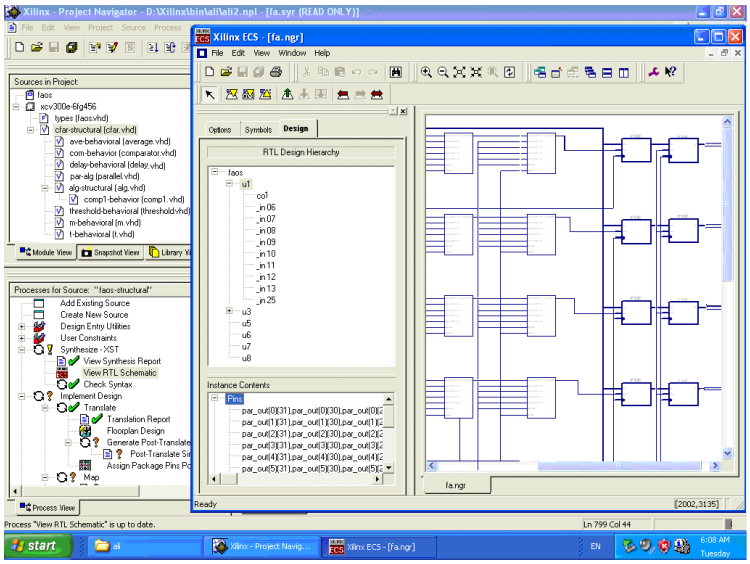

Figure (3) Inner content of FPGA Hierarchical structure of proposed Design

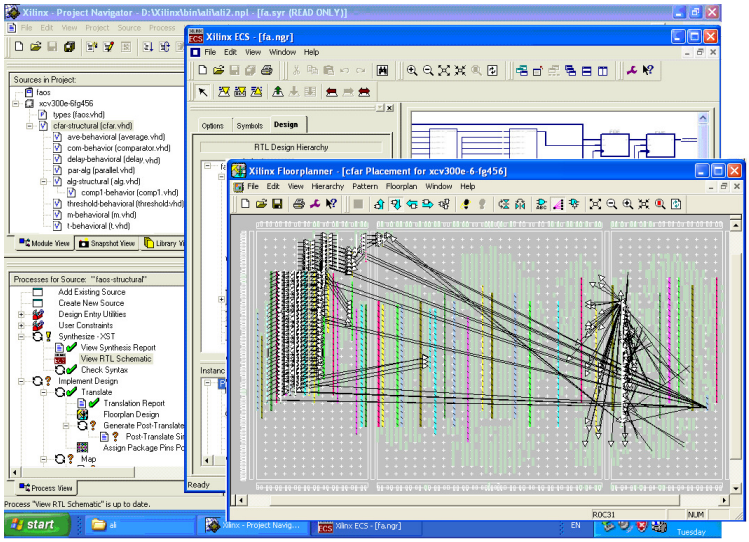

Figure (4) Floor Plant of Proposed Design

الخلاصة:ـ

التنفيذ باستخدام مصفوفة البوابات المبرمجة لمعالج إعادة التوجيه أوتوماتيكي لإحصائية الكشف

في هذه الدراسة، تم بحثنوع جديد من معالج معدل الإنذار الكاذب (CFAR) مع اقتراح إحصائية كشف (FAOSO)، والتي لا يتطلب أي معلومات مسبقة حول عدد الأهداف المتداخلة. النموذج المقترح قد تم تنفيذه باستخدام مصفوفة البوابات المبرمجة (FPGA). بعد أن أنجزت المحاكاة باستخدام لغة HDL ونفذت باستخدام برنامج*ISENavigator 6.3i*, الرقاقة ألمستهدفه بالتنفيذ نوع XC2v300e-6fg456 Virtex-II.أن المعالج المستخدم في تنفيذ هذه التقنية تم اختباره باستخدام محاكاة مونت كارلو مع وجود ثمانية أهداف متداخلة. وتفيد التقارير عن نتائج التركيب التجميعي وتحليل الوقت في النهاية,أن عملية التنفيذ والاستجابة للمعالج عكست إن استخدام هذه الأداة الرقمية ذات درجة عالية من الوثوقية والمرونة

5. REFERENCES

[1] F. Gini, A. Farina, and M. Greco, "Selected list of references on radar signal processing," IEEE Trans. Aerospace and Electronic Systems, vol. AES-37, No. 1, pp. 329-360, Jan. 2001.

[2] H.M. Finn and R.S. Johnson, "Adaptive detection mode with threshold control as a function of spatially sampled clutter-level estimates," RCA Review, vol.29, pp.414-464, Sep. 1968.

[3] M. Barkat, Signal detection and estimation, Norwood, MA:Artech House, 2005, pp. 642-647.

[4] V. G.Hansen and J. H. Sawyers, "Detectability loss due to greatest of selection in a cell-averaging CFAR," IEEE Trans. Aerospace and Electronic Systems, vol. AES-16, pp. 115–118, Jan. 1980.

[5] G. V. Trunk, "Range resolution of targets using automatic detection, "IEEE Trans. Aerospace and Electronic Systems, vol. AES-14, no. 5, pp. 750–755, Sep. 1978.

[6] J. T. Rickard and G. M. Dillard, "Adaptive detection algorithms for multiple target situations," IEEE Trans. Aerospace and Electronic Systems, vol. AES-13, no. 4, pp. 338–343, Jul. 1977.

[7] P. P. Gandhi and S. A. Kassam, "Analysis of CFAR processors in nonhomogeneous background," IEEE Trans. Aerospace and Electronic Systems, vol. AES-24, no. 4, pp. 427–445, Jul. 1988.

[8] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," IEEE Trans. Aerospace and Electronic Systems, vol. AES-19, No. 4, pp. 608–621, Jul. 1983.

[9] M. Barkat, S. D. Himonas, and P. K.Varshney, "CFAR detection for multiple target situations," IEE Proceeding, Part F: Radar and Signal Processing, vol. 136, no. 5, pp. 193–210, Oct. 1989.

[10] S. D. Himonas and M. Barkat, "Automatic censored CFAR detection for nonhomogeneous environments," IEEE Trans. Aerospace and Electronic Systems, vol. AES-28, no. 1, pp. 286–304, Jan. 1992.

[11] M. A. Khalighi and M. H. Bastani, "Adaptive CFAR processor for nonhomogeneous environments," IEEE Trans. Aerospace and Electronic Systems, vol. AES-36, no. 3, pp. 889–897, Jul. 2000.

[12] R. Ravid and N. Levanon, "Maximum-likelihood CFAR for Weibull background," IEE Proceeding, Part F: Radar and Signal Processing, vol. 139, no. 3, pp. 256–264, Jun. 1992.

[13] V. Anastassopoulos and G. Lampropoulos, "Optimal CFAR detection in Weibull clutter," IEEE Trans. Aerospace and Electronic System, vol. 31, no. 1, pp. 52–64, Jan. 1995.

[14] H. Saarnisaari, P. Henttu, and M. Juntti, "Iterative multidimensional impulse detectors for communications based on the classical diagnostic methods," IEEE Trans. Communication, vol. 53, no. 3, pp. 395–398,Mar. 2005.

[15] R. Cumplido, C. Torres, and S. Lopez, "A configurable FPGA-based Hardware Architecture for Adaptive Processing of Noisy Signals for Target Detection Based on Constant False Alarm Rate (CFAR) Algorithms," Global Signal Processing Conference, Santa Clara, CA,September 2004, pp 214-218.

[16] R. Cumplido, C. Torres, and S. Lopez, "On the implementation of an efficient FPGA-based CFAR processor for target detection," 1stInternational Conference on Electrical and Electronics Engineering,Acapulco, México. 24-27 June, 2004. pp. 214-218.

[17] T. R. Saed, J. K. Ali, and Z. T. Yassen, "An FPGA-based implementation of CA-CFAR processor," Asian Journal of Information Technology, vol. 6, no. 4, pp. 511-514, 2007.

[18] B. Magaz and M.L. Bencheikh, "An efficient FPGA implementation of the OS-CFAR processor", International Radar Symposium, Wroclaw, 21-23 May 2008, pp. 1-4.

[19] P. Ramesh Babu and R. Prasanthi, "Analysis of CFAR techniques and FPGA realization for radar detection", Proceedings of International Conference on Intelligent Knowledge Systems, Troy, Turkey, 16-20 August 2004, pp. 1-9.

[20] A. M. Alsuwailem, S. A. Alshebeili, and M. Alamar, "Design and implementation of a configurable real-time FPGA-based TM-CFAR processor for radar target detection", Journal of Active and Passive Electronic Devices, vol. 3, pp. 241-256, 2006.

[21] A. M. Alsuwailem, S. A. Alshebeili, M.H. Alhowaish, and S.M Qasim,"Field programmable gate array-based design and realization of automatic censored cell averaging constant false alarm rate detector based on ordered data variability", IET Circuits, Devices, &Systems,vol. 3, pp. 12-21, 2009.

[22] V. Winkler, J. Detlefsen, U. Siart, J. Buchlert, and M. Wagner, "FPGA-based signal processing of an automotive radar sensor," First European Radar Conference, Amsterdam, 2004, pp. 245-248.

[23] Torres C., Cumplido R., López S., "Design and Implementation of a CFAR Processor for Target Detection," 14th International Conference on Field Programmable Logic, FPL04. Lectures Notes on Computer Science Vol. 3203, pp. 943-947.

[24] B. Magaz, A. Belouchrani, and M. Hamadouche "Automatic Threshold Selection In OS-CFAR Radar Detection Using Information Theoretic Criteria " Progress In Electromagnetics Research B, Vol. 30, 157–175, 2011

[25] El Mashade, M. B., "Analysis of CFAR detection of fluctuating targets," Progress In Electromagnetics Research C, Vol. 2, 65–94,2008.