

Design and Implementation of Real-Time Automatic Censoring System on Chip for Radar Detection

Imron Rosyadi, Ridha A. Djemal, and Saleh A. Alshebeili

Abstract— Design and implementation of a novel B-ACOSD CFAR algorithm is presented in this paper. It is proposed for detecting radar target in log-normal distribution environment. The B-ACOSD detector is capable to detect automatically the number interference target in the reference cells and detect the real target by an adaptive threshold. The detector is implemented as a System on Chip on FPGA Altera Stratix II using parallelism and pipelining technique. For a reference window of length 16 cells, the experimental results showed that the processor works properly with a processing speed up to 115.13MHz and processing time 0.29 μ s, thus meets real-time requirement for a typical radar system.

Keywords— CFAR, FPGA, radar.

I. INTRODUCTION

Radar is an electromagnetic system that detects, locates, and recognizes target objects. Radar transmits electromagnetic signal and then receives echoes from target objects to get their location or other information. The received signal is frequently accompanied by noise and clutter. The disturbances may cause serious performance issues with radar systems by concluding these signals as targets.

To make a right censoring decision, the receiver is desired to achieve constant false alarm rate (CFAR) and maximum probability of target detection. Modern radars usually detect the targets by comparing with adaptive thresholds based on a CFAR processor. In this processor, threshold is determined dynamically based on the local background noise/clutter power.

The CFAR detectors have been widely used in radar signal processing applications to detect the targets from noisy background. Gini et al. noted a list of more than 120 papers about CFAR detection [1]. Though the theoretical aspect of CFAR detection is developed well, the practical hardware applications are not. The computational requirements in radar signal processing should be met by advanced technologies that employ high parallel computational technique.

I. Rosyadi is with the King Saud University, Riyadh, Saudi Arabia 11421, on leave from Jenderal Soedirman University, Purwokerto, Indonesia 53122 (e-mail: irosyadi@ksu.edu.sa or mas_jyom@yahoo.com).

R. A. Djemal, is with King Saud University, Riyadh, Saudi Arabia 11421 (e-mail: rdjemal@ksu.edu.sa).

S. A. Alshebeili is with King Saud University, Riyadh, Saudi Arabia 11421 (e-mail: dsaleh@ksu.edu.sa).

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.

In this paper, an FPGA-based design and realization of an advanced automatic censoring B-ACOSD CFAR detector is presented. B-ACOSD CFAR is a recent CFAR algorithm which performs robustly to detect automatically the target and the number interferences close to the target under log-normal clutter distribution. The SoC architecture of the CFAR detector is implemented on Altera Stratix II EP2S60 FPGA chip.

This paper is organized as follows. Section 2 describes the fundamental information about CFAR theory and the related research done on hardware realization of some types of CFAR algorithms. Section 3 describes the B-ACOSD detection algorithm. The System on Chip architecture for the B-ACOSD CFAR detector is explained in Section 4. Section 5 depicts the simulation and realization of the system. In section 6, the conclusions and future research plans are presented.

II. BACKGROUND THEORY

In a radar system, it is needed to determine the power threshold which any return can be considered from a target. In most radar detectors, the threshold is set in order to reach a required probability of false alarm rate. In natural environment, unwanted clutter and interference sources changes spatially and temporally. In this situation, an adaptive threshold should be employed, where the threshold level is changed to maintain a constant probability of false alarm. This method is known as constant false alarm rate (CFAR) detection.

A typical CFAR processor is shown in Fig. 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.

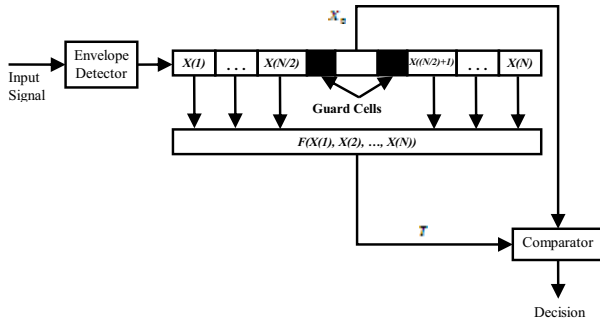


Fig. 1 Block diagram of a typical CFAR algorithm

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 (GTL-CMLD), 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 K-distribution 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) ODV 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 SoC with reconfigurable processor inside for an automotive radar sensor. The processor is responsible for controlling the custom logic and IO tasks. Simple OS-CFAR is incorporated in the system.

A recent automatic censoring called B-ACOSD CFAR have introduced by Almarshad et al. [23]. The algorithm is able to make automatic censoring of unknown number of interfering targets in log-normal clutter. Because of increase in radar resolution, the log-normal distribution becomes more reliable to represent the amplitude of clutter than Rayleigh distribution. Meanwhile, the automatic censoring algorithms developed for Rayleigh clutter as presented in [10] and [21] cannot straightforwardly be extended to the case where clutter samples are drawn from log-normal distribution.

III. B-ACOSD CFAR ALGORITHM

B-ACOSD CFAR has been proposed in [23]. The algorithms consist of two steps: removing the interfering reference cells and the actual detection. 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. This detector does not require any prior information about the clutter parameters nor do they require the number of interfering targets.

In a CFAR processor, the radar outputs $\{X_i; i = 0, 1, \dots, N\}$ are stored in a tapped delay line. The cell with the subscript $i = 0$ 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 procedure.

In censoring step, the procedures first rank the outputs of all reference range cells in ascending order according to their magnitudes to yield

$$X(1) \leq X(2) \leq \dots \leq X(p) \leq \dots \leq X(N). \quad (1)$$

In the algorithm, sample $X(N)$ is compared with the adaptive threshold T_{c0} defined as

$$T_{c0} = X(1)^{1-\alpha_0} X(p)^{\alpha_0}. \quad (2)$$

$X(p)$ is the p^{th} largest sample and α_0 is a constant chosen to achieve the desired probability of false censoring (P_{fc}). It is found that values of $p > N/2$ yield reasonable good

performance in detection [26].

If $X(N) < T_{c0}$, the algorithm decides that $X(N)$ corresponds to a clutter sample without interference, and it terminates. If, on the other hand, $X(N) > T_{c0}$, the algorithm decides that the sample $X(N)$ is a return echo from an interfering target. In this case, $X(N)$ is censored and the algorithm proceeds to compare the sample $X(N - 1)$ with the threshold

$$T_{c1} = X(1)^{1-\alpha_1} X(p)^{\alpha_1} \tag{3}$$

to determine whether it corresponds to an interfering target or a clutter sample without interference.

At the $(k + 1)^{th}$ step, the sample $X(N - k)$ is compared with the threshold T_{ck} and a decision is made according to the test,

$$\begin{matrix} H_1 \\ > \\ X(N - k) > T_{ck} \\ < \\ H_0 \end{matrix} \tag{4}$$

where

$$T_{ck} = X(1)^{1-\alpha_k} X(p)^{\alpha_k} \tag{5}$$

Hypothesis H_1 represents the case where $X(N - k)$, and thus the subsequent samples $X(N - k + 1), X(N - k + 2), \dots, X(N)$ correspond to clutter samples with interference, while H_0 denotes the case where $X(N - k)$ is a clutter sample without interference. The successive tests are repeated as long as the hypothesis H_1 is declared true. The algorithm stops when the cell under investigation is declared homogeneous (i.e., clutter sample only) or, in the extreme case, when all the $N - p$ highest cells are tested; that is, $k = N - p$. Fig. 2 shows the block diagram of the B-ACOSD algorithm.

In detection step, the cell under test X_0 is compared with the threshold T_{ak} to decide whether a target is present or not according to

$$\begin{matrix} H_1 \\ > \\ X_0 > T_{ak} \\ < \\ H_0 \end{matrix} \tag{6}$$

Hypothesis H_1 denotes the presence of target in the test cell, while hypothesis H_0 denotes there is no target.

In B-ACOSD CFAR, the threshold T_{ak} is defined as,

$$T_{ak} = X(1)^{1-\beta_k} X(N - k)^{\beta_k} \tag{7}$$

where the value of β is selected so that the design probability of false alarm (P_{fa}), and k is the number of interfering targets found in censoring step.

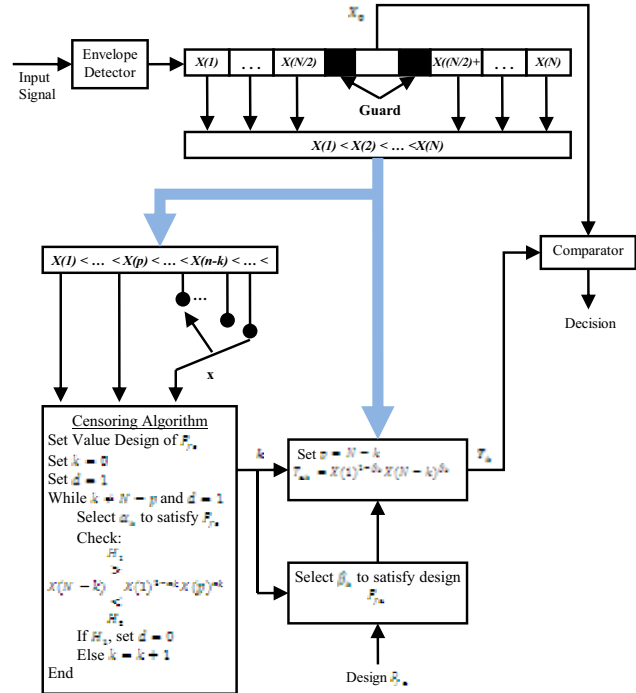


Fig. 2 Block diagram of the B-ACOSD algorithm

The values of α and β for each detected interference are defined by Monte Carlo simulation with 500,000 independent runs by maintaining low value of P_{fa} and P_{fc} respectively. The threshold parameters for $(N,p) = (16,12)$, $P_{fa} = 0.001$, and $P_{fc} = 0.01$ are presented in Table I.

TABLE I
THRESHOLD PARAMETER

	k				
	1	2	3	4	5
α_k	2.60	2.04	1.71	1.44	-
β_k	1.64	1.89	2.12	2.37	2.64

IV. B-ACOSD CFAR SOC ARCHITECTURE

A. System Overview

Design flow for the system architecture development consists of some steps started with Monte Carlo simulation using Matlab®. Monte Carlo simulation will define values of α_k and β_k for some values of N and p . The B-ACOSD CFAR algorithm for specific value of N and p then is executed under Matlab® fixed-point simulation to improve performance and maintain its accuracy when implemented in a digital system. The next step is HDL coding and test bench simulation which be done with ModelSim® from Mentor Graphics.

Integration of all components in a SoC is done using SOPC-Builder®. The software provides useful and easy tools to connect many logic elements ranging from processor, memory, to user-logic elements. Quartus II® from Altera was employed for synthesis, timing simulation and downloading the design

into FPGA chip. Hardware Abstraction Layer needed by the system was coded, compiled, and downloaded using NIOS-II IDE®.

The SoC consists of five main modules: Processor, On-Chip ROM, B-ACOSD CFAR Detector, Input/ROM interface, and Output/RAM interface, as shown in Fig. 3. The SoC utilize Nios II soft-core as the processor. Nios II is a 32-bit embedded-processor architecture designed specifically for the Altera family of FPGAs. Nios II has some key features such as custom instructions and easy custom peripherals management. Nios II/s core type is used because it is suitable to maintain a balance between performance and cost.

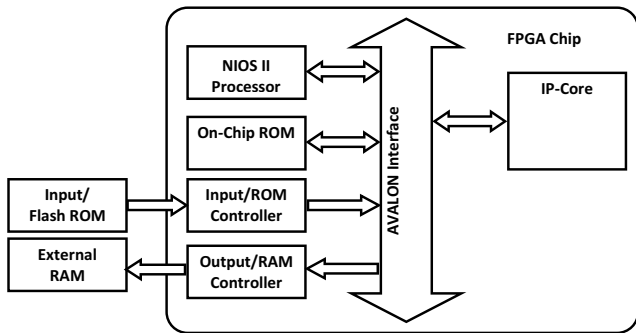


Fig. 3 Block diagram of the System-on-Chip

Input signal from an envelope detector sent directly to the Avalon bus through an input interface or be stored first in a 16-MBytes flash ROM provided by a development board and connected to the system through flash ROM driver. The censoring results are stored in a 2-MBytes external SSRAM controlled by a SSRAM driver.

Since computation of the exponential equation (5) and (7) is hard and high cost, the equations are converted respectively into logarithmic form as follows,

$$\log T_{ck} = (1 - \alpha_k) \cdot \log X(1) + \alpha_k \cdot \log X(p) \quad (8)$$

$$\log T_{ak} = (1 - \beta_k) \cdot \log X(1) + \beta_k \cdot \log X(N - k). \quad (9)$$

In those forms, power computation becomes a simple multiplication, and the multiplication becomes an addition. Because the logarithmic computation in hardware is a complex and slow task, the logarithmic computation is simplified using a look-up table. The look-up table contains range of a lognormal distribution with $\mu = 1$ and $\sigma = 1.1$ as suggested in [23], based on real radar input data measurement. Following the number representation change to logarithmic form, test cell value was also converted accordingly.

The look-up table resides on 32K on-chip ROM inside FPGA. The data distribution resolution in the 32K on-chip ROM is 0.0610. MATLAB fixed-point B-ACOSD CFAR simulation with this resolution gives censoring results as good as its real-number simulation.

B. B-ACOSD Detector Architecture

The CFAR detector comprises five main modules: I/O interface, shift register, sorting module, censoring module, and comparator as shown in Fig. 4. The shift register consists of N reference cells, G guard cells, and the test cell X_0 . The test cell is surrounded symmetrically by its reference cells and guard cells. The total length of the shift register is given as

$$L = N + G + 1. \quad (10)$$

In a register with length L , each datum streamed in input serially will need $(L + 1)/2$ clocks to be a cell test. The shift register output data are sent the value of N reference cells in parallel manner to the sorting module. The test cell from the shift register is delayed for some clocks until the threshold value of its value is computed. After sorting is done sequentially for these N cells, some of the sorted data are subjected to an automatic censoring mechanism. This mechanism will define the threshold value needed to decide the test cell as a target or not. The decision is made by comparing the threshold value with the test cell value.

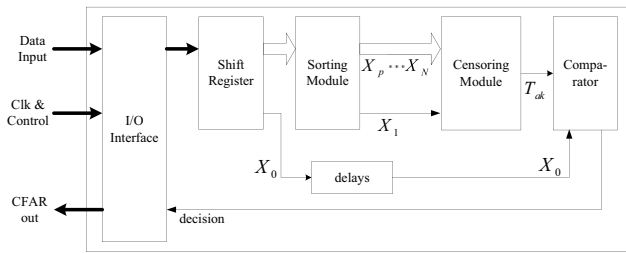


Fig. 4 B-ACOSD CFAR detector architecture

The sorting operation plays a crucial role since it consumes long computation time, and constitutes a bottleneck in the field of real-time signal processing applications. In this work, the sorting circuit is based on a parallel bubble-sort that implemented over an array of compare-swap units as illustrated in Fig. 5 for six inputs.

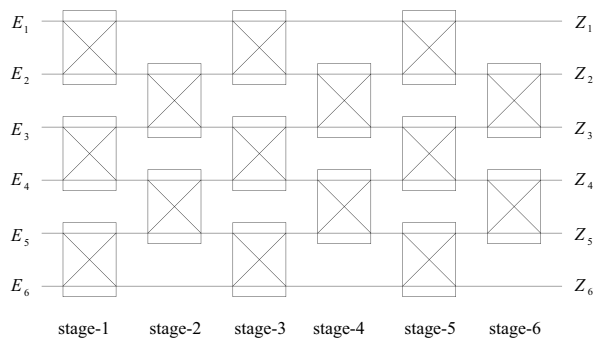


Fig. 5. Sorting module block diagram

The bubble sorting algorithm compares every two elements,

and then decides which one is the greater. The compare-swap circuit task is to compare the two inputs elements and to swap these inputs if the first element is greater than the second. This operation is repeated for each pair of adjacent elements till the end of the entire data array. The number of stages needed N elements is N stages. The advantage of this circuit is its small logic size and that the maximum value can be found with only N stages [24].

The B-ACOSD algorithm was represented as a flow chart as shown in Fig. 6. From the sorted reference cells, the number of interfering cells should be decided first by comparing T_{ck} and $X(N-k)$ for $k < (N-p)$. The number of interfering target will define the threshold value T_{ak} to be compared to the test cell X_0 .

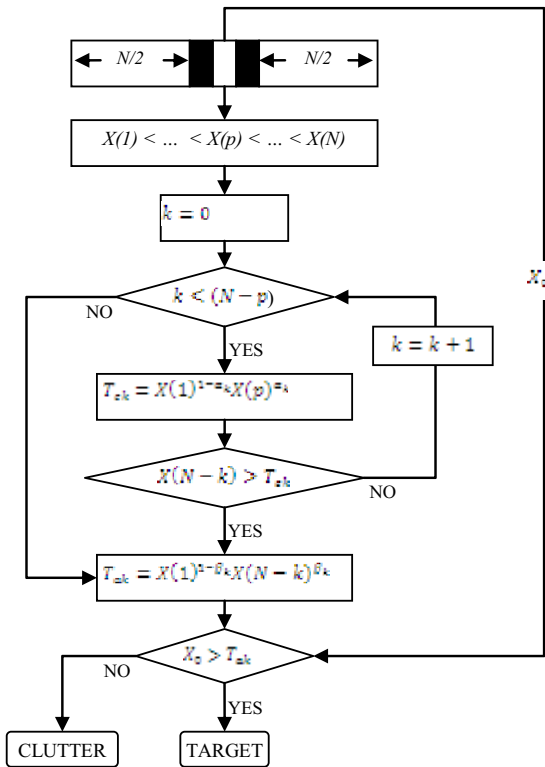


Fig. 6. B-ACOSD Censoring Flow Chart

Since there is no dependency between input/output data of T_{ck} and T_{ak} , both computation are performed simultaneously in parallel way. T_{ck} value depends on $X(1)$, $X(p)$, and α_k values, and T_{ak} depends on $X(1)$, $X(N-k)$, and β_k . This parallelism technique will accelerate the computational time, but affect the logical element required since all values of T_{ck} and T_{ak} should be computed for all possible numbers of interfering target.

T_{ck} and T_{ak} in the censoring module are computed in logarithmic form based on equation (8) and (9). The equations need a multiplication stage and followed by an addition stage. Multiplication and addition for all values of k are performed in

parallel way. For $N = 16$ and $p = 12$, the censoring module needs 23 multipliers and 9 adders in parallel. To maintain pipelining strategy in this parallel process, some delay-blocks were added to the module.

Since T_{ck} for all possible k values are computed, to define number of interfering targets, the censoring module using masking method. All the threshold values T_{ck} were fed to $N-p$ parallel comparators. The output of the comparators is a binary word of length $N-p$. This binary code is applied to a mask circuit, which determines the number of interfering target. For example if $N = 16$ and $p = 12$, the value of T_{c0} , T_{c1} , T_{c2} , and T_{c3} were compared to $X(16)$, $X(15)$, $X(14)$, and $X(13)$ to get binary decision d_0 , d_1 , d_2 , and d_3 , respectively. If $X(N-k)$ was greater than the respective T_{ck} , then d_k is logic 1, else is logic 0. The binary decision d ($d = d_0d_1d_2d_3$) decided which threshold T_{ak} which will be compared with test cell X_0 . The masking look-up table is shown in Table II.

TABLE II
MASKING LOOK-UP TABLE FOR $(N,p)=(16,12)$

Binary Decision				No. of interfering target (k)	Threshold (T_{ak})
d_0	d_1	d_2	d_3		
0	x	x	x	0	T_{a0}
1	0	x	x	1	T_{a1}
1	1	0	x	2	T_{a2}
1	1	1	0	3	T_{a3}
1	1	1	1	4	T_{a4}

The threshold (T_{ak}) resulted in the masking process was finally compared to the test cell X_0 . In this single comparator, if value of X_0 is greater than the threshold T_{ak} , then the CFAR-out is 1, else CFAR-out is 0. CFAR-out equal to 1 means the test cell is a target, else the test cell is not a target.

V. SIMULATION AND REALIZATION

The overall SoC B-ACOSD CFAR SoC consists of NIOS-II processor, on-chip ROM, B-ACOSD CFAR Detector, Input/ROM interface, and Output/RAM interface have been implemented on Stratix II EP2S60F672C3N FPGA chip. On an Altera Development Board, the config.d FPGA chip is then connected to SSRAM, Flash ROM, and other I/O's. Since the SoC design is modular, each module can be test individually.

To verify the design, we performed four type simulations: MATLAB® simulation, MATLAB® fixed-point simulation, Modelsim® functional simulation, and Quartus II® timing simulation. The timing simulation diagram for CFAR system with $N = 16$ and $p = 12$ is shown in Fig. 7. The diagram showed that for each input datum streamed to the System on Chip, the censoring output will be reached after 34 clocks. Table III detailed the clock needed for each module.

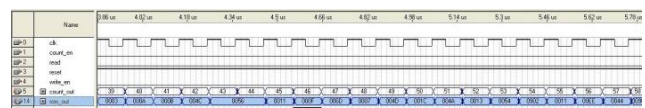


Fig. 7. Timing Diagram

TABLE III
CLOCK NEEDED FOR EACH MODULE

Module	Clock needed	Clock needed for (N,p)=(16,12)
Shift Register	$(L+1)/2$	10
Sorting	N	16
Censoring Module	5	5
TOTAL	$\frac{L+1}{2} + N + 5$	34

An input should take $(L+1)/2$ clocks to be positioned in the center of tapped delay line surrounding by N reference cells and G guard cells. Sorting module needs N clock to sort N reference cells. The clock needed by censoring module is independent to the number of reference cells. The pipeline architecture in censoring module only needs one clock for the parallel multiplier, one clock the parallel adder, one clock for the first comparator, one clock for masking, and one clock for the final comparator.

The FPGA implementation result for SoC with $N = 16$ and $p = 12$ shows that the processor can achieve a maximum operating frequency of 115.13 MHz. Because the clock needed for one CFAR detection is 34 clocks, this implies that the processing time to perform a single run is 0.29 μ s. This processing time is below the real-time requirements 0.5 μ s [23].

The B-ACOSD CFAR SoC also efficiently utilizes the FPGA hardware resources. The SoC only needs 9% combinational ALUT's, 13% dedicated logic registers, 21% of memory elements, and 19% of DSP elements as shown in Table IV.

TABLE IV
HARDWARE IMPLEMENTATION RESULTS

Logical Element	
Combinational ALUT's	4,144/48,352 (9%)
Dedicated Logic Registers	6,366/48,352 (13%)
Total Register	6540
Total pins	114/493 (23%)
Total memory bits	313,600/2,544,192 (12%)
DSP block 9-bit elements	54/288 (19%)
Total PLL's	1/6 (17%)

VI. CONCLUSION

In this work, a novel hardware implementation of a CFAR detector for radar target detection is presented. The proposed architecture is implemented as a System on Chip integrating a Nios II Core processor, on-chip ROM, and B-ACOSD CFAR IP core using Avalon Switch Fabric. The architecture exploits parallelism and pipelining techniques to met the real time requirement of the radar detection.

As a future work, the proposed architecture will be extended to direct implementation in the complete radar system. The other types of CFAR algorithm such as F-ACOSD and ACCA CFAR will be incorporated to the SoC to realize a generic CFAR detector architecture.

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," *1st International 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. Alsuwailam, 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. Alsuwailam, S. A. Alshebeili, M.H. Alhwaish, 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] M. N. Almarshad, M. Barkat, and S. A. Alshebeili, "A Monte Carlo simulation for two novel automatic censoring techniques of radar interfering targets in log-normal clutter," *9th International Symposium on Signal Processing and Its Applications*, 2007, pp 1-4.
- [24] G.M. Blair, "Low cost sorting circuit for VLSI," *IEEE Trans. Circuit System*, vol. 43, no. 6, pp. 515-516, 1996.

Imron Rosyadi was born in Banyumas, Indonesia, on September 24, 1979. He received the B.Sc. degree in electrical engineering from Gadjah Mada University, Indonesia, in 2003. Currently, he is a M.Sc. student in electrical engineering King Saud University, Saudi Arabia. Since March 2009, he was engaged in a research project with Prince Sultan Advanced Technology Research Institute (PSATRI) King Saud University, Saudi Arabia. Since 2004, he has been with Jenderal Soedirman University, Indonesia as an Assistant Lecturer. His main research interests are in the area of statistical signal processing, artificial intelligence application in signal processing, and biomedical engineering.

Ridha Ahmed Djemal received the Ph.D. degree in microelectronics (integrated circuit design) from Grenoble Institute of Technology, France, in 1996. Currently, he is an Associate Professor in the Department of Electrical Engineering at the King Saud University of Saudi Arabia, since September 2007, his research has been in the area of System Design, Image and video processing and encryption and Formal verification for System on Chip (SoC). The main theme of his research works are in the area of Design and Implementation of video application having a standard interfaces and Integrating the formal Verification in the Design flow. This work represents a very successful collaborative effort with the TELNET Firm in Tunisia and TansEda in France. Currently, he is also a member of the Electronic and Microelectronic Laboratory in Monastir – Tunisia and a member of team working on the Hardware Design Implementation of Image Encryption and Radar Recognition (CFAR) Algorithms in the Electrical Engineering Department of the King Saud University. Since October 2005, he is also an associate professor in Electronic Engineering Department, Applied Sciences and Technology Institute of Sousse (ISSAT Sousse), Tunisia.

Saleh Abdullah Alshebeili received the Ph.D degree in electrical engineering from University of Toronto, Ontario, Canada in 1992. Currently, he is a Professor of Electrical Engineering Department, King Saud University. He has more than 15 years of teaching and research experience in the area of communications and signal processing. Much of his research work has been devoted to the development of statistical estimation and modeling techniques. Such techniques play an important role in the analysis and interpretation of empirically gathered data that commonly encounter investigators in practice. He has published more than 49 papers in specialized international conferences and 37 papers in prestigious referred journals.