

# Design and Implementation of a Real Time FPGA Based CFAR Processor for Radar Target Detection Using ML403 FPGA Development Board

Boualem MAGAZ<sup>\*,\*\*</sup>, Toufik MABED<sup>\*\*</sup> and Ali ABBADI<sup>\*\*</sup>

*\*Ecole Nationale Supérieure Polytechnique, Algiers, Algeria*  
**boualem.magaz@enp.edu.dz**

*\*\*Research and Development Center, Algiers, Algeria*  
**md\_toufik16@yahoo.fr**  
**khaledkhald@yahoo.fr**

**Abstract:** The improvement in the development of theoretical aspect of CFAR radar detection is advanced and very promising, yet the practical hardware aspect is still beyond the required high computational signal processing operations. In this paper, a configurable Field Programmable Gate Array (FPGA) based hardware architecture for Ordered Statistics (OS)-CFAR processor for radar target detection is presented.

The proposed architecture is based on an efficient procedure for FPGA implementation of the OS-CFAR detector, based on the  $(N-K+1)$ -th maximum determination. By showing that the determination of the  $K$ -th order out of  $N$  reference cells is equivalent to selecting the  $(N+1-K)$ -th maximum, the detector that uses  $N$  reference cells can be implemented using only  $(N-1)$  comparators and  $(N-1)$  inverters.

The proposed structure, adapted to the ASR-9 radar parameters, is designed, implemented, and evaluated on the ML403 FPGA board. The proposed architecture shows that it can be implemented with the advantages of a parallel structure and allows an important optimization of the required FPGA hardware resources utilization. The system has the advantages of being simple, fast, and flexible with low development cost. For a reference window of 16 range cells, the experimental results carried out using Xilinx development kit showed that the proposed architecture works properly with a processing speed of 50MHz (up to 122 MHz using external clock). The execution time to perform the overall OS-CFAR detection algorithm is 0.819ms for a data set of 1000 range cells. The FPGA implementation results are presented and discussed.

**Key words:** FPGA, Implementation, OS-CFAR processor, Radar.

## INTRODUCTION

Detection of target in a background of clutter is a problem of interest in radar field. In order to improve such detection system, the designer usually prefers a constant false alarm rate. To achieve this purpose, the actual interference power must be estimated from the data in real time, so that the threshold can be adjusted to maintain the desired probability of false alarm (Pfa).

A detection processor that can maintain a constant Pfa is called Constant False Alarm Rate (CFAR). Finn and Johnson [FIN 68] developed a theory based on arithmetic mean of the neighboring resolution cells of the cell under test. This is known as Cell Averaging CFAR detector.

The CA-CFAR detector was shown to be efficient in homogenous environment. In fact, the probability of detection approaches the classical Neyman-Pearson

case where the mean level of clutter is known a priori, provided that these cells do not contain non-homogenous samples. However, the detector based on order statistics (OS-CFAR) proposed by Rohling [ROH 83] provides inherent protection against serious performance degradation in the presence of non-homogenous samples.

The OS-CFAR processor estimates the noise power simply by selecting the  $K$ th largest cell in the reference window; the threshold is obtained from one of the ordered samples of the reference window. The range samples are first ordered according to their magnitudes, and the statistic  $Z$  is taken to be the  $K$ th largest sample.

Although the improvement in the development of the theoretical aspect of CFAR detection is advanced and very promising, yet the practical hardware aspect is still beyond the required high computational signal processing operations. Nowadays, Field Programmable Gate Arrays (FPGAs) have emerged as an attractive

and ideal environment for hardware realizations for high speed algorithms and intensive computation applications.

Compared to general purpose processors, they are not only offer high degree of parallelism, but also can achieve much higher processing speed. Thus, specific parallel structure can be designed to optimize the implementation of this kind of detector.

In this paper, we propose a real time Field Programmable Gate Array, FPGA, implementation of the OS-CFAR detector, based on the (N-K+1) maximum direct determination using the ML403 FPGA development board.

### 1. Direct implementation of the OS-CFAR processor

The sorting algorithm is the operation that puts elements of a list in a certain order. This operation plays a crucial role since it consumes long computation time, and constitutes a bottleneck in the field of real-time signal processing applications. The sorting algorithm can work in descending order (data elements are sorted from the largest to the smallest) or in ascending order (data elements are sorted from the smallest to the largest). Since the processing time is critical, the decision of choosing the highest speed and most efficient method of data sorting is of a great interest.

Recently, few attempts considering hardware implementation of CFAR processors have been reported. In particular, configurable hardware architecture for adaptive processing of noisy signals for target detection based on CFAR algorithms has been presented in [CUM 04b]. The architecture has been designed to deal with parallel/pipeline processing and to be configured for three versions of cell-averaging (CA) CFAR algorithms [CUM 04a]. In [FAR 06] hardware implementation of CA-CFAR processor using conventional discrete components has been presented.

The bubble sorting is one of the best sorting algorithms that combine high speed and simplicity for applications that involve small number of elements. This kind of algorithm compares every two elements, and then decides which one is the greater; this may reduce the implementation resources requirements and ensure real time implementation using dedicated processors.

In order to implement the OS-CFAR processor, we have to rank-order all the reference data cells, as shown in figure 1.

The obtained K-th sample is then multiplied by the scaling factor,  $Tos$ , to get the final threshold level. The scaling factor depends on the reference window size,  $N$ , and the desired false alarm probability,  $Pfa$ .

If the data in the cell under test exceeds the threshold level, the processor declares a target present at the appropriate range.

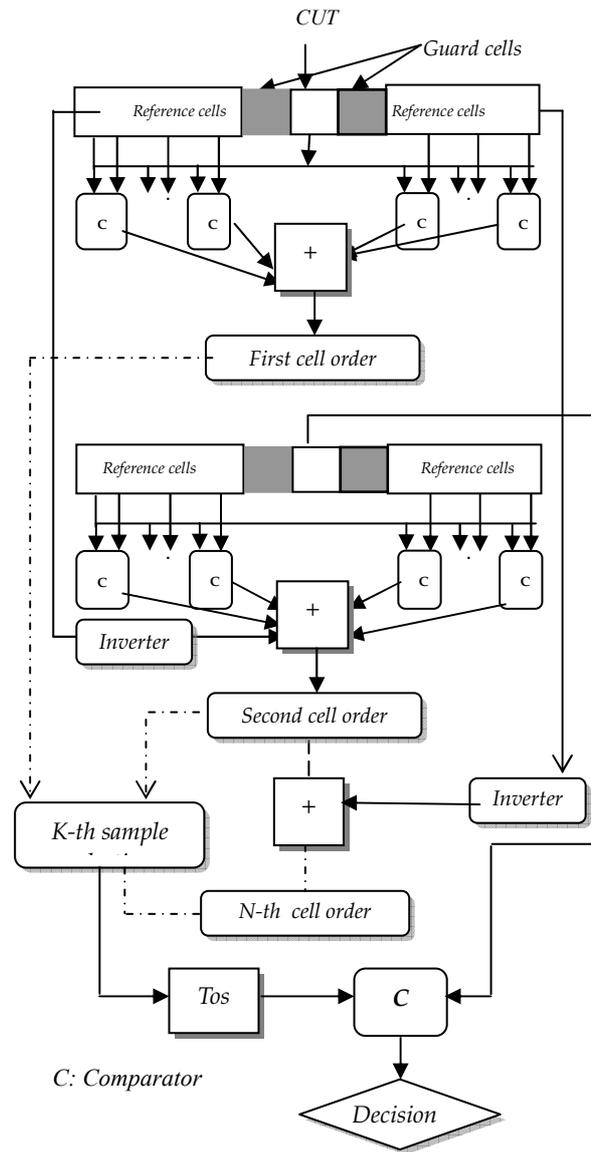


Figure 1. Direct implementation of the OS-CFAR processor architecture

However, it is more difficult to realize the OS-CFAR detection in real time with general purpose microprocessors or digital signal processors because of the high throughput rate required in radar systems. Therefore, it is preferred to design a special structure for the OS-CFAR processing.

The computational complexity of OS CFAR is of a different form since other types of operations has to be performed, mainly comparison and move operations.

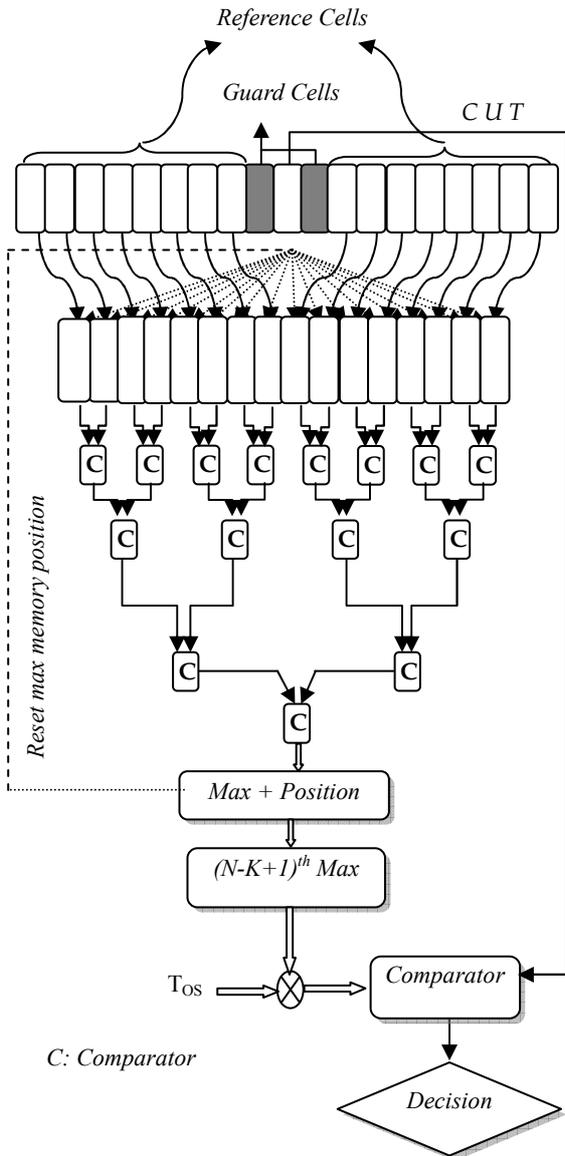
The threshold is obtained by the comparison of each reference cell with all other reference cells.

The implementation of this approach requires  $(N-1) \times (N/2)$  comparators and  $(N-1) \times (N/2)$  inverters for a reference window size  $N$ .

Furthermore, this method makes the required resources depend strongly on  $N^2$ . For example, for  $N=16$ , we need 120 comparators and 120 inverters.

## 2. The proposed approach

The proposed approach for the OS-CFAR processor implementation is shown in figure 2.



**Figure 2.** Block diagram of the parallel OS-CFAR processor FPGA implementation

This architecture with a window size,  $N$ , consists of  $(N-1)$  comparators and  $(N-1)$  inverters and one decision bloc (one multiplier and one comparator).

The proposed approach consists on the following steps:

1. Determine the maximum and its position in the reference window by successive comparisons;
2. Zero setting of the current maximum in the reference window;
3. Repeat steps 1 and 2 up to  $(N-K+1)$ ;
4. The  $(N-K+1)$  maximum represents the  $K$ th order.

For a number of reference cells  $N$ , only  $(N-1)$  comparators are needed to determine the maximum.

In practice,  $K$  is greater than  $N/2$ , the proposed structure reduces the number of comparators by  $(K/N)*100\%$ . This reduction achieves 75% if  $K=3N/4$ .

The obtained  $K$ -th order sample is then multiplied by the scaling factor,  $T_{os}$ , to get the final threshold level. If the data in the cell under test exceeds the threshold level, the processor declares a target present at the appropriate range.

## 3. FPGA implementation

The proposed architecture for the OS-CFAR processor implementation, was designed, synthesized and simulated using VHDL Hardware Description Language and Xilinx ISE targeted for a Virtex-4 M1403 FPGA board (XC4VFX12-10FF668 device).

The Virtex-4 M1403 FPGA board Development Kit provides a complete solution for developing designs and applications based on the Xilinx Virtex-4 FPGA family. The M1403 board utilizes Xilinx Virtex-4 device (XC4VFX12-10FF668 device).

Powered by the 4VFX12 device and supported by industry-standard peripherals, connectors and interfaces, the Virtex™-4 ML403 FX Evaluation Platform provides a great entry-level environment for developing embedded designs based on the Virtex-4 FX FPGA.

The real time considerations are managed so as to satisfy the requirements of classical pulsed radar with a pulse repetition frequency of 1200Hz,  $1\mu s$  pulse width. The period between two sweeps is the critical time for the implementation of the OS-CFAR processor.

Table1 summarizes the FPGA hardware resource utilization and timing performance after synthesis and FPGA fitting.

According to the FPGA synthesis results, the proposed architecture provides a good trade-off between performance and hardware resources utilization. This feature makes it suitable to be embedded in as a radar signal processing module. The default configuration of the OS-CFAR processor uses 12-bits data, 16 reference cells, 2 guard cells and  $K=12$ .

Number of slices	1115 (20%)
Number of flip flop	890 (8%)
Number of 4 input LUTs	1547 (14%)
Number of bonded IOBs	19 (5%)
Number of GCLKs	1 (3%)
Maximum clock frequency (After place and route)	122 MHz

**Table 1.** Synthesis summary for the OS-CFAR processor implementation targeted for a XC4VFX12-10FF668 Virtex-4 device (ML403 FPGA Board)

The proposed system execution time to perform the OS-CFAR detection algorithm is 0.819ms, including

processing latency, for a data set of 1000 range cells.

The obtained results show that the required resources are less than the available FPGA resources comparatively to the direct implementation approach which requires resources that exceeds the maximum available.

The proposed structure operates every clock cycle so that it can speed up the data processing rate and reduce the latency and the required FPGA resources.

We note that the maximum clock frequency, 122MHz, is achieved using external clock. The clock frequency used is about 50 MHz.

#### 4. Conclusion

In this work, a novel hardware implementation of OS-CFAR processor for radar target detection is presented.

This system has the advantages of being simple, fast, and flexible with low development cost. The performance of the prototype hardware setup proved the concept of the proposed CFAR processor.

Therefore, this programmable hardware with its significant improvement in the processing speed will make it possible to increase the resolution of the radar range cells and open new avenues for further useful real time applications.

The proposed FPGA implementation structure of the OS-CFAR processor, based on the direct selection of the  $(N+1-K)$ th maximum in the reference window of size  $N$ , allows the optimization of the required FPGA hardware resources utilization of about by  $(K/N)*100\%$  comparatively to the direct implementation approach.

The proposed architecture has been synthesized with Xilinx ISE targeted for a XC4VFX12-10FF668 Virtex-4 device.

#### REFERENCES

- [BEN 07]: Bencheikh M. L., B. Magaz, M. Hamadouche and A. Belouchrani, "Performance analysis of a new CFAR detector", Proc..International Radar Symposium, pp.361-365, Cologne, Germany, September 2007.
- [BER 00]: Bergeron Janick "*Writing Testbenches, Functionnal Verification of HDL Models*", Kluwer Academic Publishers, 2000.
- [CUM 04a]: Cumplido R., César Torres, Santos Lopez, "On the implementation of an efficient FPGA- based CFAR Processor for target detection", *ICEEE and CIE2004*, Acapulco, Guerrero; Mexico, September 2004.
- [CUM 04b]: Cumplido R., 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, GSPX2004*. Santa Clara CA. September 2004.
- [FAR 06]: El-Faramawy N., E. El-Badawy and A. Salem,

"Hardware Implementation of CA-CFAR processor", *Proceeding of the Int. Conf. On Computer and Communication Engineering ICCCE06*. Vol.1, 9-11May 2006, Kuala Lumpur, Malaysia. pp. 573-578.

- [FIN 68]: Finn H.M. and Johnson R.S., "Adaptive detection mode with threshold control as a function of spatial sampled clutter level estimates", *RCA Rev.* Vol. 29, September 1968 pp. 414-464.
- [MAG 08]: Magaz B., Bencheikh M. L., "DSP Implementation of a Range Azimuth CFAR Processor", *Proc. International Radar Symposium*, pp.317-320, Wroclaw, Poland, May 2008.
- [ROL 83]: Rohling H., "Radar CFAR thresholding in clutter and multiple target situations", *IEEE Transactions on aerospace and electronic systems*, Vol. 19, PP.608-621, July 1983.
- [SKO 00]: Skolnik Merrill Ivan, "*Radar Handbook*", Editorial McGraw-Hill, 2000.
- [UWE 01]: Uwe Meyer-Meyer, "*Digital Signal Processing with Field Programmable Gate Array*", Springer Verlag Berlin Heidelberg, 2001.