A VERSATILE HARDWARE ARCHITECTURE FOR A CFAR DETECTOR BASED ON A LINEAR INSERTION SORTER

Roberto Perez-Andrade, Rene Cumplido, Claudia Feregrino-Uribe, Fernando Martin Del Campo

Department of Computer Science
National Institute for Astrophysics, Optics and Electronics INAOE
Apdo. Postal 51 & 216 Santa Maria Tonantzintla, Puebla, Mexico
email: {j_roberto_pa, rcumplido, cferegrino, fmartin}@inaoep.mx

ABSTRACT

This paper presents a versatile hardware architecture that implements six variant of the CFAR detector based on linear and non-linear operations. Since some implemented CFAR detectors require sorting, a linear sorter based on a First In First Out (FIFO) schema is used. The proposed architecture can be used as a specialized module or co-processor for Software Defined Radar (SDR) applications. The results of implementing the architecture on a Field Programmable Gate Array (FPGA) are presented and discussed.

1. INTRODUCTION

The problem of detecting target signals in background noise of unknown statistics is a common one in sensor systems such radars and sonars. In radar applications, this noise usually comes from thermal noise, clutter, pulse jamming or other undesired echo received by the antenna. Adaptive digital signal processing techniques are often used to remove noise and to enhance the detectability of targets in many situations. An attractive class of schema that can be used to overcome the problem of noise added to the target signal are the *constant false alarm rate* (CFAR) algorithms which set a threshold adaptively based on local information of total noise power. The threshold in a CFAR detector is set on a sample by sample basis using estimated noise power by processing a group of samples surrounding the sample under investigation [1], [2].

There are various CFAR techniques proposed in the radar literature in order to deal with different problems present in radar applications. These techniques require linear operations or nonlinear operations like sorting a set of values and selecting one on a specific position before performing a linear operation. These different techniques have been developed in order to increase the target detection probability under several environment conditions [2].

Although the theoretical aspects of CFAR detectors are very advanced [2], [3], [4], [5], [6], and analog implementation have been used in radar systems for a number of years, recent developments in programmable logic have made practical to explore digital implementations of CFAR and other algorithms to support the SDR paradigm. SDR systems can be implemented using programmable logic to accommodate various radar sensors for different detection conditions. This means they can be changed in run-time either by control of stored software or by downloading new functions [7]. Using a configurable architecture implemented on FPGA is an alternative to avoid the fixed-functional hardware since it allows the modification of certain CFAR detector's parameters. For practical SDR applications, all processing blocks, including the CFAR detector, must support several processing modes and operate with a high computational load in real-time. This work presents a versatile hardware architecture that supports six CFAR detectors. The proposed CFAR hardware architecture can be used as a specialized processing module or co-processor in the receiver´s processing chain of a SDR system.

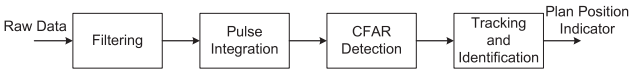

Fig. 1. Traditional Signal Processing Radar Chain.

2. CFAR AND OS-CFAR DETECTORS

A radar transmitter generates an electromagnetic signal that is broadcast to the environment by an antenna. An energy portion of this broadcast signal is reflected by targets. This reflected energy is received by the same antenna and sent to the receiver. In the receptor this energy is digitalized to produce raw data that is then processed to obtain the desired target information. Figure 1 shows a radar receiver processing chain and the position of the CFAR detector.

The CFAR detector (Figure 2) consists of a reference window with *2n* cells which surround the cell under test. Each cell stores an input sample and the values stored in the

978-1-4244-1961-6/08/$25.00 ©2008 IEEE.

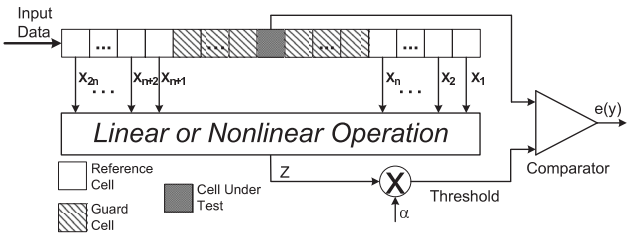

Fig. 2. Generic CFAR Detector.

cells are right shifted when a new sample arrives. Some *m* guard cells are incorporated in order to avoid interference problems in the noise estimation. The reference cells are used to compute the *Z* statistic. This *Z* statistic and a scaling factor α are used to obtain the threshold. This scaling factor depends on the estimation method applied, the false alarm required according to the application and it is related to the noise distribution in the radar environment. The resulting product αZ is directly used as the threshold value that is compared with the cell under test (CUT) to determine if the CUT is declared a target. The target detection problem can be modeled by $H_1 : y = d+g$ and $H_0 : y = g$; where H_1 and H_0 are the target present and target absent hypothesis, respectively; d represents the target signal and g the environmental noise component. If the values of the CUT exceed the αZ, then the target present is declared, i.e. the CFAR processor outputs 1 if a target is present, otherwise outputs 0. The decision criterion is represented by:

$$e(y) = \begin{cases} H_1, & CUT \geq \alpha Z \\ H_0, & CUT < \alpha Z \end{cases} \quad (1)$$

The method to obtain the *Z* statistic from the reference window might be based on linear or nonlinear operations. The most common linear detectors are the cell averaging (CA), greatest of (GO) and smallest of (SO). These detectors calculate the arithmetic mean of the amplitude contained in the Y_1 lagging cells and Y_2 leading cells from the CUT. The equation 2 summarizes these three linear operations for the *Z* statistic:

$$Z = \begin{cases} \frac{1}{2}(Y_1 + Y_2) \\ \max(Y_1, Y_2) \\ \min(Y_1, Y_2) \end{cases} \quad (2)$$

Among the nonlinear detectors are the order statistics cell averaging (OSCA), order statistics greatest of (OSGO) and order statistics smallest of (OSSO). These order statistics detectors need to perform a rank-order operation over the leading and lagging reference cells, i.e. sort the reference cells values and then select the *k-th* sorted value. The OSCA, OSGO and OSSO CFAR detectors perform the selection of the *k-th* ($Y_{(1)}$) and *i-th* ($Y_{(2)}$) sorted value from the leading and lagging cells, respectively. Once selected these two values, the *Z* statistic is calculated in a similar way as the linear detectors as shown in the following equation:

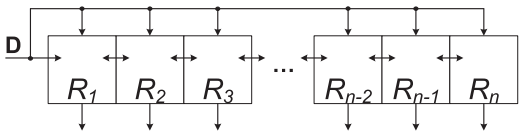

Fig. 3. Sorting Array

$$Z = \begin{cases} \frac{1}{2}(Y_{(1)} + Y_{(2)}) \\ \max(Y_{(1)}, Y_{(2)}) \\ \min(Y_{(1)}, Y_{(2)}) \end{cases} \quad (3)$$

A CFAR detector that can be considered optimal under any environmental circumstances has not been designed yet. Each one of the explained detectors has its advantages and disadvantages, and may be optimal under particular environment conditions. The detection performance is altered by varying the number of references cells, guard cells, the CFAR detector, the *k-th* rank-order sample and the false alarm required (represented by the scaling factor α) [2]. In order to give robustness to the target detection process radar applications, a specialized architecture which supports several of these detectors, and allows to change their parameters such as, selection of scaling factor α and the *k-th* and *i-th* rank-order sample is required. The proposed architecture presented in this work support the six previously explained CFAR detectors.

3. HARDWARE ARCHITECTURE

The proposed architecture uses a linear sorter in order to perform the rank-ordering operation needed in the order statistic detectors. Since keeping the values sorted does not affect the averaging process needed in the linear detectors, the use of a linear sorter is possible. The architecture is parameterizable in terms of its reference cells, guard cells and arithmetic precision.

3.1. Linear Insertion Sorter

The linear sorter used in this architecture implements the insert sort algorithm. It consists of an array (figure 3) of identical processing elements (PE), called Sorting Basic Cell (SBC). In order to fulfill the FIFO sorting functionality, the SBCs must be interconnected in a simple linear structure, called sorting array. This SBC array sorts the values as they are introduced into the sorting array, discarding the oldest value in the sorting array, while maintaining the values sorted in a single clock cycle i.e. in a FIFO schema. The SBC has a register with synchronous load to store the value,

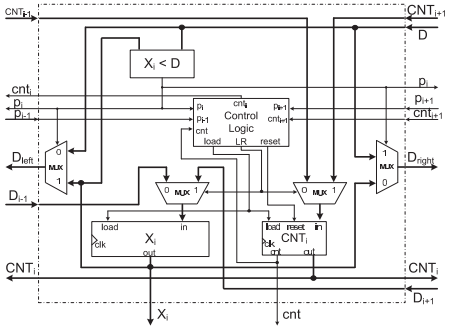

Fig. 4. Architecture of the SBC.

a counter with synchronous reset and load to store the period life of the value, a comparator, four 2-1 multiplexers and control logic (figure 4). This linear insertion sorter is explained in detail in [9].

3.2. CFAR Detector Architecture

The proposed CFAR detector architecture, figure 5, has two SBC's sorting arrays for *2n* the reference cells, *2m+1* shift registers for the guard cells and CUT, which is at the middle of these registers. Also, the architecture has two *n*-1 multiplexers for the lagging and leading sorting arrays respectively. One of each side of these multiplexers performs the rank operation. Given that the reference cells values are ordered, the *k-th* and *i-th* value can be selected by the control signals Sel-k and Sel-i respectively. The result of this selection are the $Y_{(1)}$ and $Y_{(2)}$ values needed in the nonlinear operations.

According to equation 2, it is needed to add all the values stored in the leading and lagging sorting arrays. In order to perform this operation, it is not necessary to add all values each time that one value from the sorting array is inserted and deleted. Only by adding and subtracting the newest and oldest values respectively, the next result is obtained. This whole operation can be performed by the PE Accumulator, which computes the accumulation of the Y_1 and Y_2 values on each sorting array. This PE Accumulator, consists of an adder, a subtracter and a register to store the accumulated Y_n value. The multiplexer control line value is generated by a priority decoder whose input is a data bus formed by the cnt_n signals coming from the SBC. This *cnt* signal indicates when the life period value has expired inside of one SBC's, i.e. the oldest value that must be subtracted and passed to the shift registers. Two 2-1 multiplexers perform the selection between Y_1 and $Y_{(1)}$ from the lagging side and Y_2 and $Y_{(2)}$ from the leading side. An ALU-like module provides the three modalities for computing the *Z* statistic: the average, the maximum and the minimum of the rank-order or the accumulated value. The desired modality is chosen using the control signal SelDet (Select Detector) established either manually by the user or by an automatic control expert system. A multiplier scales up the *Z* statistic with a fixed scaling factor α and a comparator decides whether a target is present or absent as indicated by equation 1.

In this architecture, on each clock cycle, values flow from the lagging sorting array, to the shift registers and to the leading sorting array. In order to begin the target detection processing, a reset signal must be applied. Once data begin to flow as mentioned above, *2n+2m+1* clock cycles of latency are required for having all the values stored in the sorting arrays and in the shift registers. After this latency time, the architecture produces a valid output for each clock cycle allowing for the continuous operation of the target detection process.

4. RESULTS AND DISCUSSION

The architecture was development for a commercial X-band non-coherent radar, which performs a 360 degrees scan in 2.5 seconds (24 rotations per minute, RPM). Incoming raw data from the receiver is sampled to produce a set of more than 16M samples per scan that are processed in a stream fashion. For the purpose of validation, the proposed architecture was modeled using the VHDL Hardware Description Language and synthesized with Xilinx ISE 9.1 targeted for a XtremeDSP Development Kit, with a Xilinx's Virtex-4 XC4VSX35 FPGA device. The default configuration of the CFAR detector 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 area results for this device are 1,364 (8%) slices, 2,637 (8%) flip flops and 690 (2%) LUTs meanwhile the maximum frequency operation is 198 MHz. This architecture requires 84 milliseconds to process a radar data set, which is 30x times faster than the required theoretical processing time of 2.5 seconds needed for this application parameters; thus this same module can be potentially used in radars with much higher resolution or CFAR detectors with larger *n* and *m* values.

A direct comparison between our proposed CFAR architecture and other architectures is not possible, because they do not perform the same functionality. In [8] the proposed CFAR detector use only three linear detectors, besides this work does not implement the sorting functionality. Nevertheless, a parallelism grade can be applied between [8] and our proposed architecture. This grade can be the architecture's throughput measured in millions of operations per second (MOPS). Concurrently our architecture performs *2n+7* arithmetic operations each cycle, while the other architecture performs only 7 arithmetic operations. This means that for a same CFAR detector configuration and the maximum operation frequency, the proposed architecture achieves a throughput of 14,058 (MOPS). The throughput achieved

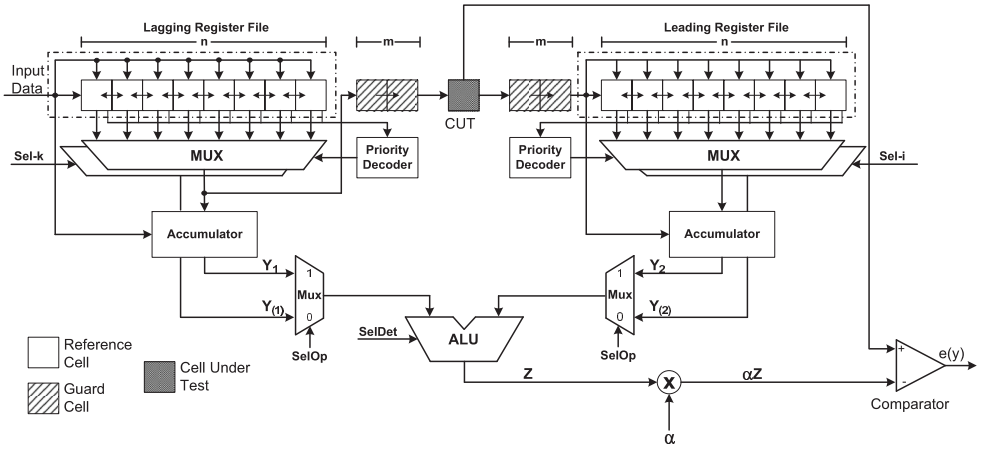

Fig. 5. CFAR Detector Hardware Architecture.

in [8] is 840 MOPS on a XC2V250 Virtex II. For throughput comparison purpose, our architecture was also synthesized for this last FPGA device, getting a throughput of 7,526 MOPS which is nine times more that in [8].

5. CONCLUSION

CFAR detectors are used in signal processing applications to extract targets signals from background noisy. For radar applications, the theoretical aspects of CFAR detectors is advanced, with a number of CFAR algorithms proposed for several environment conditions. As no optimal CFAR detector has been proposed, for practical implementations of software defined radars, a versatile processing architecure that is able to switch among different CFAR detectors and perform in real time is required. The proposed architecture allows to select among six CFAR detectors, scaling factor α and the *k-th* and *i-th* rank-order sample giving robustness to the target detection process. In order to support the non-linear processing, the architecture performs a linear insertion sort based on a FIFO schema. The linear sorting operation is performed with an array of PE called SBC. The architecture exploits the parallel nature of the CFAR signal processing and it can be easily extended to accommodate larger CFAR detectors as required by more demanding applications. Thus, this high performance, yet compact, architecture can be used as a specialized processing module or co-processor in the radar processing chain for conventional or in SDR systems.

6. ACKNOWLEDGMENTS

First author thanks the National Council for Science and Technology from Mexico (CONACyT) for financial support through the scholarship number 204500.

7. REFERENCES

[1] Skolink M. L., "Introduction to RADAR Systems," New York, Mc Graw Hill, Third Edition 2001.

[2] Gandhi P.P., Kassam S.A., "Analysis of CFAR Processors in Nonhomogeneous Background," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 427–445, 1988.

[3] Rohling H., "Radar CFAR Thresholding in Clutter and Multiple Target Situations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 19, no. 4, pp. 608–621, 1983.

[4] Fuste E. García G. M., Reyes Davó E., "Analysis of some modified order statistic CFAR: OSGO and OSSO CFAR," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 1, pp. 197–202, 1990.

[5] You He., "Performance of some Generalised Modifed Order Statistics CFAR Detectors with Automatic Censoring Technique in Multiple Target Situations," *IEE Proceedings in Radar, Sonar and Navigation*, vol. 141, no. 4, pp. 205–212, 1994.

[6] Tsakalides, P.; Trinic, F.; Nikias, C.L., "Performance Assessment of CFAR Processors in Pearson-Distributed Clutter," *IEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 4, pp. 1337–1386, 2000.

[7] Di Wu, Yi-Hsien Li, Eilert, Johan Dake Liu., " Real-Time Space-Time Adaptive Processing on the STI CELL Multiprocessor," *EuRAD European Radar Conference 2007*, pp. 71–74, 2007.

[8] 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 Cumputer Science Vol. 3203, pp. 943-947

[9] Perez-Andrade R., Cumplido R., Martin Del Campo F., Feregrino-Uribe C. "A Versatile Linear Sorter Based on a FIFO Scheme," *IEEE Computer Society Annual Symposium on VLSI* ISVLSI'08, pp. 357-362