Real Time Implementation on TMS320C6711 DSP processor of a new CFAR Radar

H. Mansouri *hmansouri@usthb.dz*
Image Processing and Radar Laboratory, University of Science and Technology Houari Boumediene , Algiers, Algeria.

M. Hamadouche *hamadouche-mhamed@hotmail.com*
LIMOSE Loaboratory, Department of Infotronic, University of Boumerdes, Boumerdes, Algeria.

F. Youcef Ettoumi *fyoucefettoumi@yahoo.fr*
Image Processing and Radar Laboratory, University of Science and Technology Houari Boumediene , Algiers, Algeria.

Abstract

In this paper we present the results of performance analysis in terms of real time implementation of GOWMAX-CFAR detector for radar targets of Swerling I type embedded in white Gaussian noise, in no homogenous clutter. So we expose the optimized algorithm considered for its implementation on DSK 6711 card using a DSP TMS320C6711 processor, and the obtained results compared with those of OS CFAR algorithm.

Keywords: CFAR, DSP, Implementation, Radar.

1. INTRODUCTION

In most DSP applications, the real time processing is very critical. Appropriate chooses must be made for the DSP able to execute the work in real time. [1], [2], [3].

For our application we have chosen to satisfy the requirements of classical pulsed radar with a pulse repetition frequency of 1200 Hz, 1.05µs pulse width. The period of the coherent processing interval is the critical time for the implementation of the proposal system.

The considered radar to be implemented operates according to a Constant False Alarm Rate (CFAR) algorithm. To achieve this, those detectors use an adaptive threshold for deciding for presence or absence of a target [4].

It is derivate from Ordered Statistic OS-CFAR, developed by H.Rohling in [5], and referred as GOWMAX CFAR whose characteristics and performances detection of Swerling 1 type targets have been studied in [6].

One DSP module is used to process the radar signal. This module allows a Texas Instruments TMS320C6711 floating point 32 bits DSP [7].

2. PROBLEM FORMULATION

CFAR detection algorithm is a digital signal processing or process by which a target is declared present or absent in the cell under test, while maintaining a constant false alarm rate. Its principle is based on the decision by comparing the signal in the cell under test with an adaptive threshold called Q in the presence of thermal noise in adjacent cells.

Automatic detection amounts to solving the problem of the presence (decision H1) or absence (H0 decision) of a target by taking cells one by one.

Indeed, in each direction of the space swept by the radar range is divided into several hundred cells and testing for in each cell using a window of N cells, as shown in figure 1. Sliding this window along the reference cells can cover the entire distance radar. Also, we note the existence of guard cells, which are not used in the estimation process, as they may contain a portion of the target power. Finally, the decision is based on a comparison of a sample of the cell under test to an adaptive threshold depending on H0 or H1.

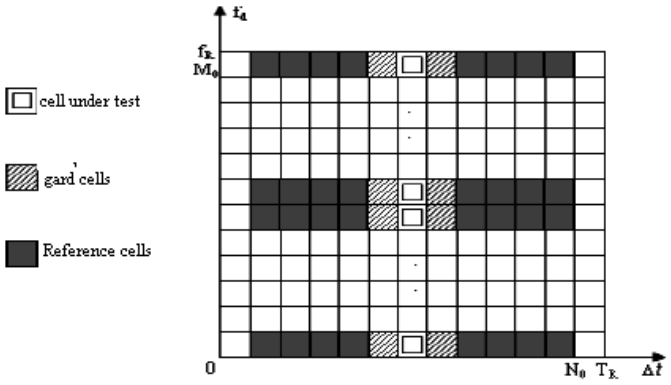

FIGURE 1: Organization of Analysis Windows.

CFAR detectors differ from each other by way of estimating the noise.The bloc diagram of the considered one is given in figure 2. In this scheme the received radar signal, is divided into samples each one containing a target signal or noise. In this diagram, the received radar signal is divided into N cells whose resolution is equal to the duration of the transmitted pulse, and distributed on two half windows of size N / 2 each one. The largest cell energy level is taken from each half window, and then multiplied by a weighting coefficient. The estimated noise Q is then formed by taking the maximum of the two obtained maximal values. Thereafter forming the adaptive threshold T * Q, and the contents of each cell one after the other is compared to the threshold for deciding the presence or absence of a target. T is the threshold multiplier which permits to maintain a constant false alarm rate.

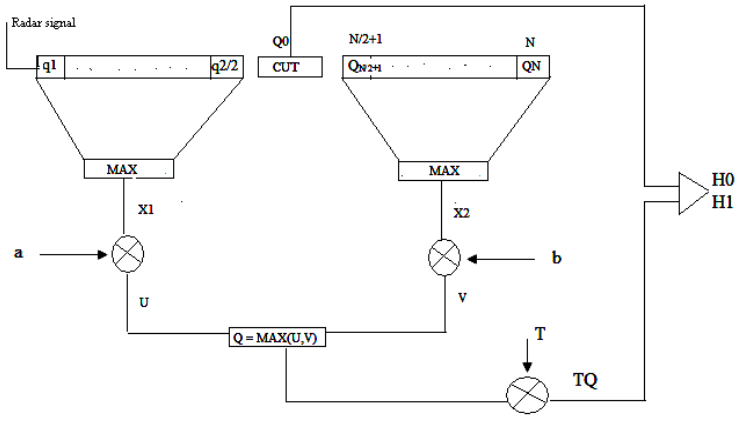

FIGURE 2: Diagram Block of GOWMAX CFAR.

3. CHOICE OF LANGUAGE AND CONSIDERED ALGORITHM.

To write a DSP program we can use an assembler language or a dedicated high-level language.

We have chosen to develop the program on the TMS320C6711 DSP in a high-level language as the C one, who has the advantage of being known by most engineers have to work in the field of digital signal processing . DSP program written in C can be relatively easily understood by many people, without having needed to know precisely the target DSP.

The program will consist of a main one in C, three files in assembly language, a command file, and a standard library file. It corresponds to the algorithm described in figure 3.

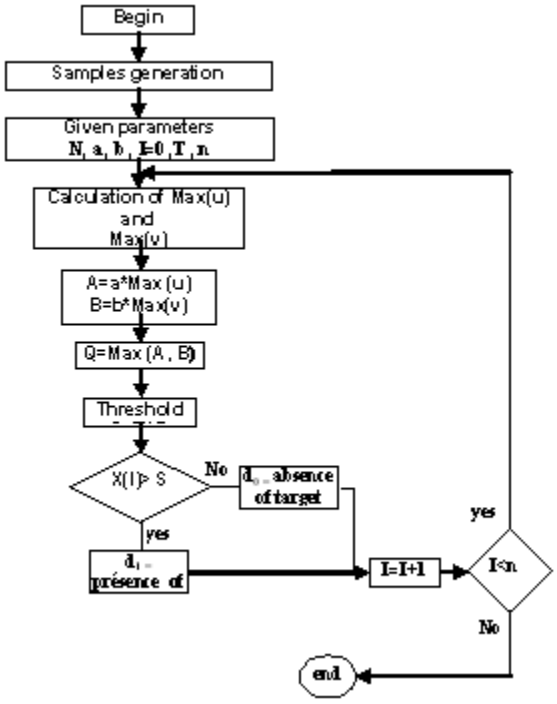

FIGURE 3: Corresponding Algorithm of GOWMAX CFAR Detector.

4. IMPLEMENTATION ARCHITECTURE

The proposed architecture for the GOWMAX CFAR processor implementation is given in figure 4. It consists on determining the maximum in each reference half window by successive comparisons, then multiply it respectively by weighting coefficients a and b. Then we form the estimate of noise by tacking the greatest of. The adaptive threshold is then formed. The result is compared with cell under test for taking the decision.

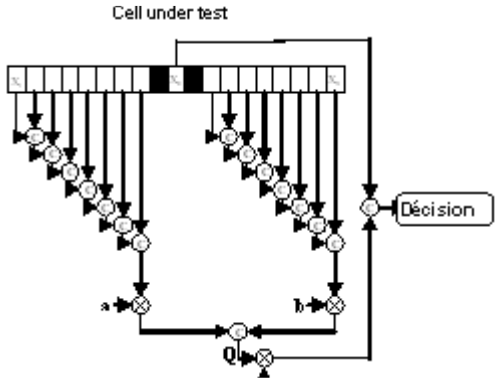

FIGURE 4: Implementation Architecture.

5. RESULTS AND DISCUSSION

5.1 Simulation results

Code composer studio [7] is a fully Integrated DSP Development Environment. Code Composer Studio TM (CCS) Development Tools are a key element of the eXpressDSP Software and Development Tools strategy from Texas Instruments.

CCStudio delivers all of the host tools and runtime software support for TMS320 DSP based real-time embedded application to market faster.

The fully integrated development environment includes real-time analysis capabilities, easy to use debugger, C/C++ Compiler, Assembler, linker, editor, visual project manager, simulators, XDS560 and XDS510 emulation drivers and DSP/BIOS
support.

It allows running or stopping the process, view and modifying registers, memory values and variables in C and assembler view and source C. The characteristics of the CCS are:

- Ability to start and stop the 'C6x.'
- A window that displays CPU register values of the DSP.
- A window that displays the values of variables in the desired format.
- Window display vectors.
- Ability to change data values in any window.
- Ability to set breakpoints software in assembler or C source with a click with the mouse

In our case and through the simulation, we have

- First check the structure and adequacy of program
- On the other hand, it allow us to study the behavior of the detector in various situations clutter

Figure 5 to figure 7 shows some results of implementation of GOWMAX CFAR detector on DSP processor under Composer Studio Code, and for different situations generated by random samples. In those graphs, the first signal represents the received one, the second shows the corresponding adapted threshold, and the third the decision processing.

- Probability of false alarm has been taken equal to 10-4 and for a size of window analysis N=16.
- Weighting coefficient have been fixed respectively to a=0.5 and b=0.01. Those optimal values have been deducted from performances analysis of the radar [6].

Figure 5 has been depicted for homogenous environment. In this situation we can see that all targets have been detected.

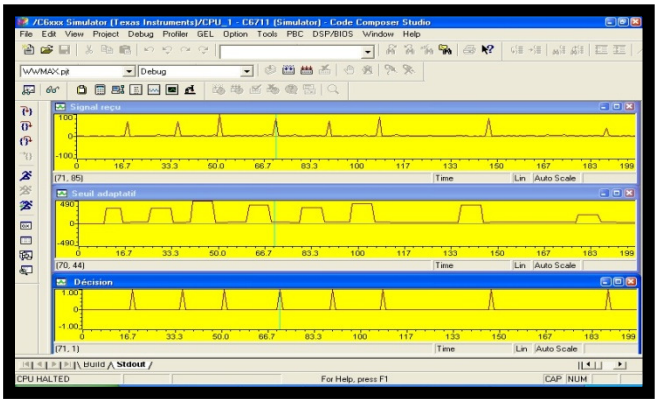

FIGURE 5: Homogenous Environment.

In presence of interfering targets figure 6 shows that the detection depends on position of level of primary and secondary targets. This confirms our deductions in [6].

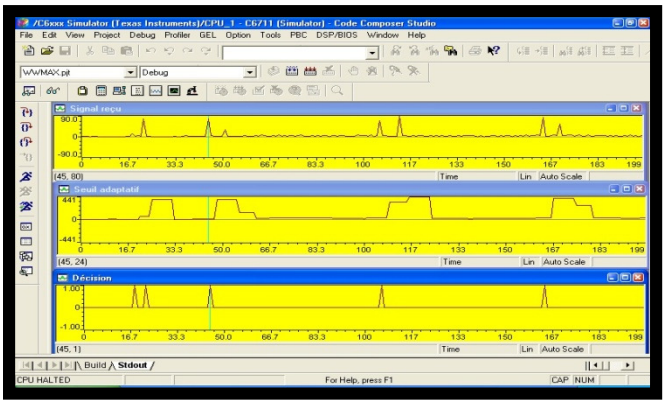

FIGURE 6: In Presence of Interfering Targets.

The same conclusion can be done on figure 7 when environment forms a clutter edge.

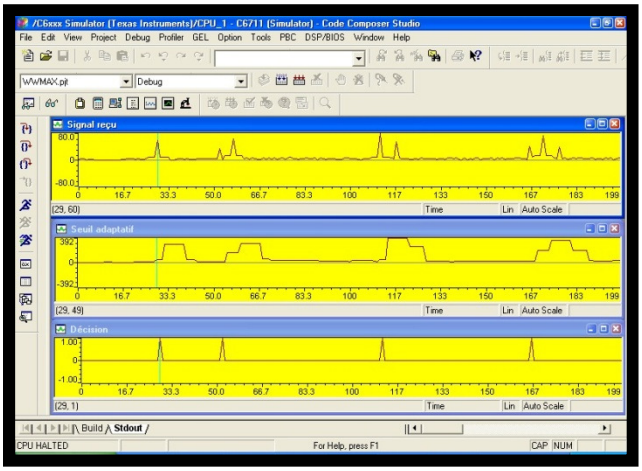

FIGURE 7: In Presence of Clutter Edge.

Since GOWMAX CFAR is derivate from OS CFAR, we have implemented this last to compare performances.

OS CFAR is based on estimating the noise power at the k-th sample taken equal to the samples of which the N reference cells are ordered in an increasing manner, and the comparison of the cell under test with the estimated threshold [5]

This detector is known to be efficient in presence of interfering targets provided that the number of these does not exceed K / N, K being the rank of the selected cell to estimate noise [8], in which case the result is an effect of masking and CFAR losses, as shown in figure 8, for N= 16 and five interfering targets (k=12).

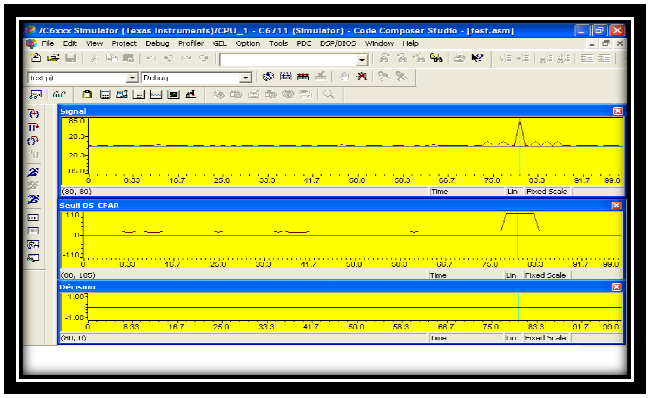

FIGURE 8: Masking effect of targets of OS CFAR detector.

5.2 Implementation results

After simulation, we have processed to implementation of the detector on DSK 6711 card using a DSP TMS320C6711 floating point processor. Its organization is shown in figure 9.[7].

The floating point DSPs are more flexible and easier to program than the fixed-point DSPs. TMS320C6711 DSP manipulates numbers formed with 1 bit sign, 8-bit exponent and a 23bits mantissa (data size in memory: 32-bit single-precision and double-precision 64-bit). Thus, the available dynamic range is very large.

The DSK package from spectrum Digital is a complete DSP system with hard and software support tools. The board includes a Texas Instrument C6713 floating point DSP and a 32-bit codec.

The DSK board includes 16Mb of synchronous dynamic random access memory and 256Kb of flash memory. Four connections on the board provides audio input and output; MIN in, LINE in, LINE out and Headphone. There are four user dip switches that provide users with feedback control. The DSK operates at 225 Mhz. The DSP core is designed for high performance floating point operation. Beyond the DSP core, the C6713 integrates a number of on-chip resources that improve functionality and minimize hardware development complexity

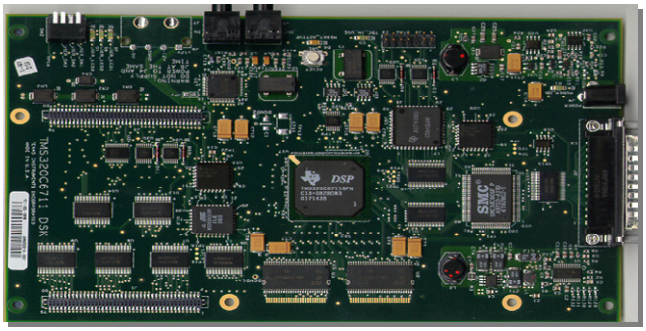

FIGURE 9.a: DSK 6711 Card.

DSP TMS320C6711 processor internal architecture is given in figure 9 [7]. It integrates CPU, peripherals, and external interfaces. The central unit of C6711, which is the heart of the processor is characterized by:

- A clock of 150 Mhz which corresponds to a cycle time of 6.7nS
- Six Arithmetic and Logic Units (ALU) and two multipliers.
-32 bit registers
- An address bus of 32 bits
- 32 bit data formats
- Execution of 900 MFLOPS

With its VLIW architecture (Very Long Instruction Word) CPU can achieve up to 8 simultaneous instructions per cycle time. Thus, this architecture allows:

-A load store architecture
-A set of instructions to reduce code size
-100% conditional instructions for faster execution
-A reduced instruction set (RISC).

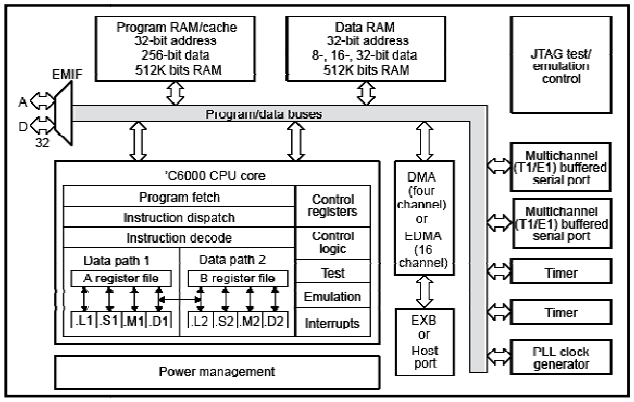

FIGURE 9.b: Internal Architecture of TMS320C6711 DSP Processor.

As said above, the real-time processing constraints are chosen to satisfy the requirements of an ASR-9 radar type[9], which is a surveillance radar whose characteristics are:

-A repetition frequency of 1200 Hz which corresponds to a period τ = 0,833 ms
- A turning speed of 12.5 tr / min
- An average range of 120 km
- It works in the band (2.7-2.9 GHz) with a pulse width of 1.05 µs and a beam angle of 1.3 degrees which corresponds to:

$$(360\ ^\circ / 1.3\ ^\circ) = 277 \text{ azimuthally cells.} \quad (1)$$

Knowing that the radar ASR9 turns in 4.8s, we can deduce the time Tmax required to scan an azimuthally cell:

$$Tmax = (4.8ms/277) = 17ms. \quad (2)$$

And we can process a number of samples:

$$Do = (0.833ms/1.05 \text{ microseconds}) = 794 \text{ per pulse} \quad (3)$$

From any radar or its characteristics, to discuss the feasibility of the implementation of the algorithm "GOWMAX-CFAR" in real time on a TMS320C6711 DSP with a cycle time of:

$$Tcyc= 6.7 \text{ ns} \quad (4)$$

Tcyc must be the processing time of up to 794 samples in real time and does not exceed the scan time of a cell azimuthal Tmax. This corresponds to a maximum number of cycles not exceeding

$$Tmax / Tcyc = 2537313 \text{ cycles.} \quad (5)$$

So if we want to establish a mathematical relationship, and that we can call it the condition of real-time processing, we have to write:

$$Ncyc < Tmax / Tcyc \quad (6)$$

with:

Ncyc = the total number of cycles of treatment.
Tmax = time to scan a target.
Tcyc = cycle time of the DSP used.

As showed in figure 10, in terms of execution time of the GOWMAX CFAR we have obtained 2240027 cycles corresponding to 15ms, value who has been deducted for 794 samples.

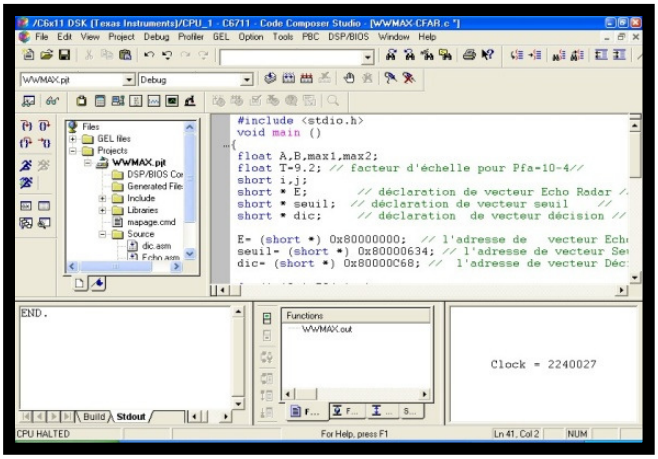

FIGURE 10: Execution Time of Implementation.

For OS CFAR detector, for the same conditions we have obtained an execution time of about 20 ms.

6. CONCLUSION

We have exposed in this paper the various stages of the implementation of the algorithm of a new CFAR detector referred as GOWMAX-CFAR.

We have proposed a simple architecture of algorithm and we have also used all the resources of TMS320C6711 DSP processor in order to implement the algorithm in question under real-time processing constraints.

Our results were acceptable with a processing time (execution time) equal to 15 ms less than the maximum tolerable time not to exceed that of 17ms.

The comparison with OS CFAR lead us to deduce that the new detector offers the best implementaribility for similar performances in term of detection.

The results presented in this study were obtained with random data we generated manually. The work dealt only with the real time processing.

We hope for the future, be able to enter real data, and organize the treatment by adding to map the sampling and analog-digital conversion, all obviously within the constraints of radar operation.

7. REFERENCES

[1]. N. kehtarnavaz, ‘’Real-time digital signal processing based on the TMS320C6000’’, ELSEVIER, 2005.

[2]. R. Chassaing, Digital Signal processing and applications with the C6713 and C6416 DSK, A John Wileyand Sons, Inc., Publications, 2005.

[3]. D. Pal Sharma and J. Singh, ‘’DSP Based Implementation of Scrambler for 56Kbps Modem’’, Signal Processing – An International Journal (SPIJ), Vol. 4, pp 85-96, May 2010.

[4]. H.M. Finn and R.S. Johnson, “Adaptive detection mode with threshold control as a function of spatial sampled clutter level estimates”, RCA Rev. Vol. 29, pp. 414-464,Sep. 1968.

[5].H. Rohling, “Radar CFAR thresholding in clutter and multiple target situations”, IEEE trans; Aerospace Electron. Systems, Vol. AAES-19, pp. 608-621, Jul. 1983.

[6]. H. Mansouri, M. Hamadouche, F. Youcef Ettoumi, “Performance analysis in heterogenous environment and implementation on DSP processor of a GOWMAX CFAR detector,” in Proc. International Radar Symposium, Vilnius, Lituania, June 2010, pp.431-433.

[7]. Texas Instruments, “TMS320C62x/C67x CPU and Instruction Set Reference Guide”, 1998.

[8. M.B. El Mashade, ‘’ Performance analysis of OS structure of CFAR detectors in fluctuating targets Environments ‘’ Progress In Electromagnetics Research C, Vol.2,pp.127-158, 2008.

[9]. C. Wolff, ‘’ Airfield Surveillance Radar ASR-9” Internet:
http://www.radartutorial.eu/19.kartei/karte203.en.html, May 2011.