

Real time design and implementation of MTD-SWVD using FPGA

Original
Article

Ibrahim M. Ali¹, Fathy M. Ahmed¹, Mohamed R. Amen², Mohamed R. Al-Ghonaimi²

¹Military Technical College, ²Faculty of Engineering, Cairo University

Keywords:

Fast Fourier transform, field programmable gate array (FPGA), moving target detector (MTD), Wigner Ville distribution (WVD).

Corresponding Author:

Ibrahim M. Metwalli, Military Technical College, Cairo, Egypt, **Tel:** 01282399906.

Email: ibrahim_metwally77@yahoo.com.

Abstract

Recently, Summed Wigner Ville Distribution-based Moving Target Detection (SWVD-MTD) has been proposed by the authors to replace the traditional Fast Fourier Transform (FFT) based MTD or conventional Wigner Ville Distribution (WVD)-based MTD^[1]. SWVD-MTD performance outperforms other MTD schemes with expected more hardware and speed complexity. In this paper, real time hardware design and implementation of SWVD-MTD using Field Programmable Gate Array (FPGA) is proposed. Comparison of implementation details of SWVD-MTD and other schemes, in terms of FPGA resources, is presented. As expected, the extra hardware complexity and speed of real time implementation is solved by using FPGA.

I. INTRODUCTION

Radar signal processing aims detection and location of targets signal in existence of undesirable signals such as noise, clutter and jamming signals. The power of these undesirable signals is greater than that of target signal. So, Radar signal processing techniques are employed to overcome this problem to achieve reliable detection.

MTD is one of these radar signal processing techniques which are used in filtering out these interfering signals. Traditional MTD applies coherent, linear Doppler filtering, adaptive thresholding and fine ground clutter map as shown in fig.1 (path AA'). It results in an enhanced performance through achieving high probability of detection (Pd) and low probability of false alarm (Pfa)^[2].

The MTD digital signal processor core is a bank of Doppler filters. Ideally, its length equal to the number of the returned pulses during each illumination time.

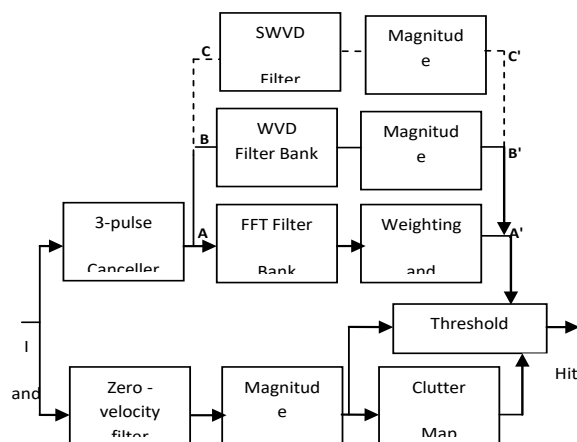


Fig. 1: Block diagram of MTD-I (path AA'), MTD-WVD (path BB'), and the proposed MTD-SWVD (path CC')

Fig. 2 shows the radar operating area that is divided into a number of range-azimuth (RA) cells, where the total range is divided into 'P' range cells and the azimuth into 'M' cells^[3].

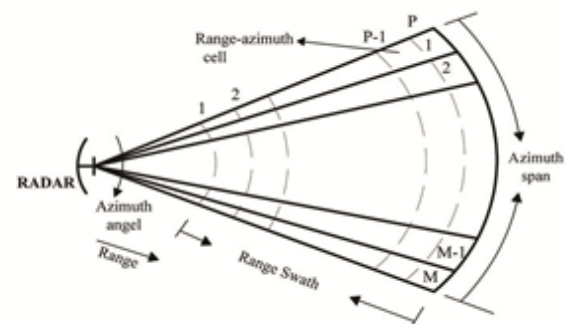


Fig. 2: Radar coverage area.

During each of these 'M' azimuth rays, a fixed number of pulses was reflected from each of these 'P' range cells. The number of reflected pulses, N, in each range azimuth cell can be calculated by^[4]:

$$N = \frac{f_r \theta_b}{\Omega} \quad (1)$$

Where:

f_r is the pulse repetition frequency.
 θ_b is the pulse repetition frequency.
 Ω is the antenna scanning rate.

This group of pulses from each RA cell is called the coherent processing interval (CPI). Pulse Doppler processing of these returned pulses is applied to determine whether its source is the presence of target or clutter and noise alone. The clutter elimination process is affected by the Doppler-filter bank sidelobes, i.e. the lower sidelobes is the better discrimination of the target^[2].

Three MTD schemes for realizing the Doppler filters bank are considered. MTD-I utilizes direct FFT of the received data sequence^[2] (path AA' shown in Fig. 1). MTD-I is characterized by high sidelobes level at the output of the Doppler filter bank which increases the probability of false alarm.

The second scheme is known as MTD-WVD (path BB' shown in Fig. 1). It is based on WVD to realize the Doppler filter bank which produces low sidelobes level without need to additional weighting as MTD-I. MTD-WVD improves the detection performance and enhances the improvement factor. Unfortunately, MTD-WVD suffers from hardware complexity and also cross terms effect, in case of multiple targets, which is caused by the bilinear nature of WVD^[5].

The third scheme, designated as MTD-SWVD (path CC' shown in Fig. 1), realizes the Doppler filter bank based on SWVD. It offers superior performance through presenting lower sidelobe level than the two mentioned schemes and enhances the improvement factor with 9 dB over MTD-WVD and 11dB over MTD-I. Also, it overcomes the problem of cross terms appeared in MTD-WVD. But, MTD-SWVD scheme is associated with the problem of extra hardware complexity^[1].

This paper is organized as follows: After the introduction, section 2 discusses SWVD theory and MTD-SWVD scheme overview. Section 3 gives the detailed design and implementation of MTD-SWVD. Section 4 demonstrates the experimental results. Section 5 presents the hardware complexity evaluation. Finally, section 6 presents the conclusion of this work.

2- SWVD THEORY AND MTD-SWVD

Though the FFT, WVD, and SWVD have been used in the realization process of MTD-I, MTD-WVD, and MTD-SWVD, respectively, only the SWVD theory is discussed in this section, since the theory of FFT^[2,6] and conventional WVD^[7,8] have been well established.

2.1 SWVD THEORY

The summed Wigner-Ville distribution (SWVD) is derived from WVD ; it has been used for realizing a bank of Doppler filters in MTD (MTD-SWVD)^[1]. Mathematical description of SWVD is as follows. The discrete-time version of WVD of the analytic discrete data $s(n)$ is given by^[9]:

$$WVD(n, f) = \sum_{m=-\infty}^{\infty} s\{(n+m/2)\Delta t\} s^*\{(n-m/2)\Delta t\} \times \exp(-j2\pi f m \Delta t) \quad (2)$$

Where Δt is the sampling interval and n, m are integers. The term $s\{(n+m/2)\Delta t\} s^*\{(n-m/2)\Delta t\}$ is referred as the kernel sequence.

Windowed WVD is referred as Pseudo WVD^[5,9]. However, for data sequences with short length (e.g., radar CPI), window process can be neglected and Eq. (2) can represent PWVD with redefined Kernel and then can be implemented by FFT operation as follows^[5]:

$$PWVD(n, m) = \sum_{k=0}^{N-1} x(n, k) \exp(-j2\pi k m / N) \quad (3)$$

Where N is the (CPI-2) length, $PWVD(n, m)$ is the energy distribution of complex data at time index n and frequency index m . for N even, let $Q(n, k) = s(n, k)$ then:^[5]

$$x(n, k) = \begin{cases} Q(n, k). Q^*(n, -k) & k = 0, \dots, \frac{N}{2} - 1 \\ Q(n, k - N). Q^*(n, -k + N) & k = \frac{N}{2}, \dots, N - 1 \end{cases} \quad (4)$$

Kernel calculations for SWVD differs from that of traditional WVD such that; cross-products calculations are carried out for all time index then a summation process of complex kernel values is applied over time to generate summed kernel^[10], instead of calculating the kernel only one time as in WVD^[5]. FFT is then applied on the summed kernel of the signal under test to obtain the frequency spectrum as follows:

$$SWVD(m) = \frac{1}{N} \left| \sum_{k=0}^{N-1} SR(k) \exp(-j2\pi k m / N) \right| \quad (5)$$

$$SR(k) = \begin{cases} \sum_{n=k}^{N-k-1} x(n+k)x^*(n-k), k = 0, \dots, N/2 - 1 \\ 0, k = \frac{N}{2} \text{ for } N \text{ is even} \\ SR^*(N-k-1), k = N/2 + 1, \dots, N - 1 \end{cases} \quad (6)$$

Where, m is normalized frequency index and n is discrete time index. $SR(k)$ is the summed cross products of discrete signal $x(n)$ calculated in equation(4).

2.2 MTD- SWVD

Here, The Doppler filters bank in a MTD system of typical ground based radar is realized based on SWVD instead of direct FFT and conventional WVD as shown in fig.1. For CPI of length M pulses which MTD processes coherently. $N = M - 2$ complex data samples, out of three pulse canceller, are at the input of SWVD filters bank. Moreover, data interpolation by a factor of 2 is applied before summed kernel calculation as shown in fig .3.

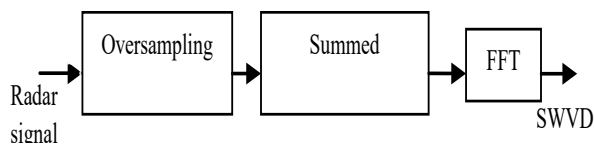


Fig. 3: SWVD computation.

This method overcomes the problem of cross-terms effect of WVD caused by its bilinear nature as shown in fig .4^[1].

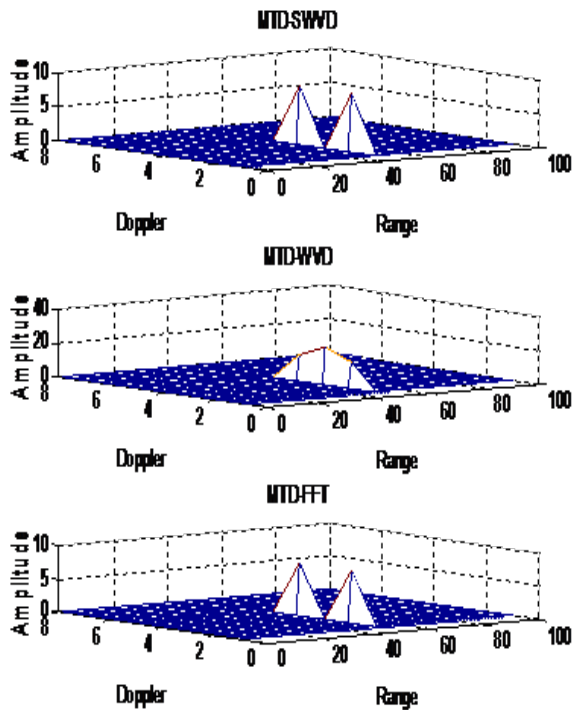


Fig. 4: Range Doppler information for the MTD-I, MTD-WVD, and the proposed MTD-SWVD.

Where returned signals of two target located at the same range cell with diverse normalized Doppler frequency (one at $f_d = 2/8$ and the other at $f_d = 4/8$) are generated and applied to the three MTD schemes and range Doppler information are plotted.

Also, it improves the target detection capabilities by providing higher detection probabilities, as shown in fig. 5 through receiver operating characteristics (ROCs) curve, and an additional gain of 9 and 11 dB in the improvement factor as shown in fig. 6, in the presence of ground and weather clutter, compared to MTD-WVD and MTD-I schemes, respectively^[1].

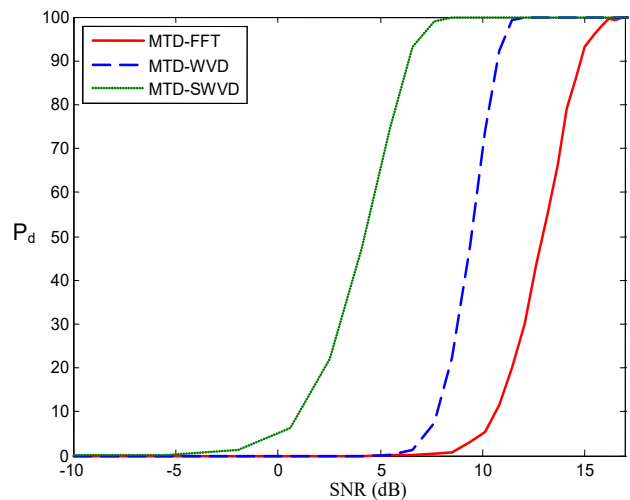


Fig. 5: ROC of MTD-I, MTD-WVD, and MTD-SWVD for $P_{fa} = 10^{-6}$

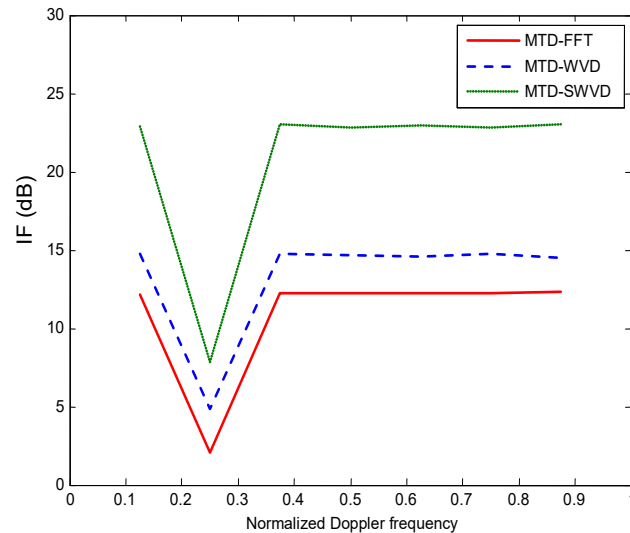


Fig. 6: Improvement factor of MTD-I, MTD-WVD, and MTD-SWVD against normalized Doppler shifts

3. IMPLEMENTATION OF MTD-SWVD

In this section, real time implementation issues of the Doppler filters bank based on SWVD (MTD-SWVD) using FPGA is present. The design is achieved through two parts;

A software part which is Xilinx ISE_13.2 package and Modelsim 6.5 simulator^[11, 12]; and a hardware part which is the Spartan-3 FG900^[13]. The general block diagram of the implemented MTD-SWVD scheme is shown in Fig.7.

Design and implementation of each module of the general block diagram of fig.7 is given in the following subsections. Achieving real time implementation requires that, the time required for all the processing operations applied on the data sequences doesn't exceed the period of these data. Fig. 8 illustrates the way for executing real time processing on the radar CPIs. So, while acquiring the data sequences of the CPI_n, complete processing of the previous CPI_{n-1} is achieved simultaneously.

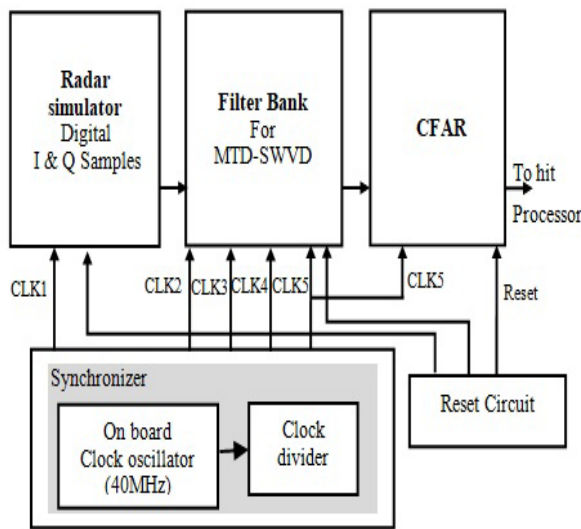


Fig. 7: Simplified block diagram of the hardware implementation of MTD-SWVD scheme

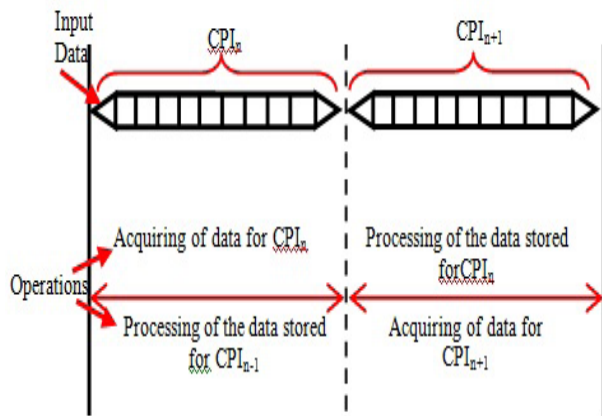


Fig. 8: Sequences of real time data processing

3.1 RADAR SIGNAL SIMULATOR

For the purpose of testing the proposed implementation in real time, a radar signal simulator is implemented. It is assumed that the radar data which shall be processed by the MTD processor is a pulsed radar signal with duration of 5.5 μ sec. So, according to Shannon theory, it is fetched with a rate of 370 KHz.

The received signal after the phase detector is modulated by the Doppler frequency of the moving target. Its frequency is the target Doppler frequency. This signal is contaminated by AWGN, ground clutter, and weather clutter. The output of the phase detector is the in-phase and quadrature components of the received signal. Each of these components is passed to analog to digital converter (ADC). The assumed length of the radar CPI is 8, and the number of range cells is taken to be 32. The output of the ADC is assumed to be of 8-bits word length including a bit for the sign (the Most Significant Bit (MSB)).

A Matlab program is used to get the digital 8-bits, I and Q samples, representing a moving target with Doppler frequency of $0.5 F_r$ contaminated by AWGN of zero mean and unity variance. Weather clutter with 500HZ spectral width of mean Doppler frequency of $0.25 F_r$ and CNR of -30 dB and also Ground clutter with 50HZ spectral width of zero mean Doppler frequency are added to the signal. The value of SCNR is 6 dB. The length of these samples is corresponding to one CPI, i.e for each I and Q components this length is $32 \times 8 = 256$ samples.

The generated I and Q samples are stored in two different Read-Only-Memories (ROMs) in the radar simulator, which is designed and implemented inside the used FPGA, in a suitable way to perform kernel calculations and FFT processing on every eight successive data samples for the same range cell inside FPGA. Fig. 9 shows the simplified schematic diagram of designing this radar simulator. The arranged I and Q samples are fed to the filter bank module with rate CLK1. Such that, every eight successive samples, of the same range cell, should be followed by a hold-on 4-clocks as required for kernel calculation. The address of the radar simulator ROMs is generated by a binary 8 bits counter. The process of counting occurred such that the counter holds-on for 4-clocks after counting 8-clocks, then the process is repeated till the end of the CPI. The transition from hold-on state to counting state is done by a reset signal which is generated by another counter included in the reset circuit inside the synchronizer.

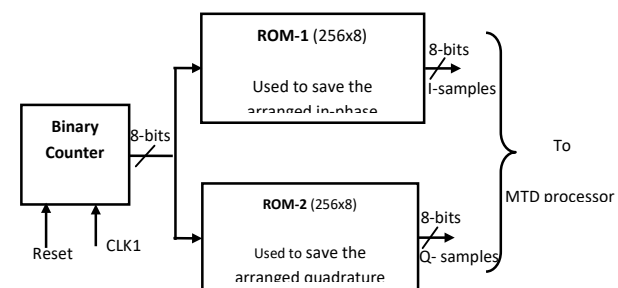


Fig. 9: Simplified schematic diagram of the radar signal simulator

3.2 SYNCHRONIZER

A synchronizer is designed to produce clock signals with different speeds. These clock signals are used through the MTD-SWVD modules according to the design to compensate the time delay results because of execution of complex operations.

The synchronizer uses the 40MHz on board clock oscillator, which is the master clock. Also, it includes a clock divider to produce five clocks CLK1, CLK2, CLK3, CLK4, and CLK5. Fig.10 demonstrates these five clock signals.

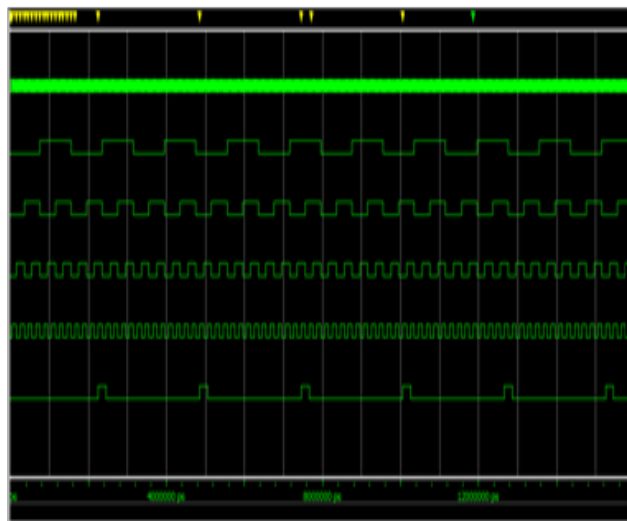


Fig. 10: Different generated clock signals by the synchronizer

The use of different clock signals is as follows:

CLK1, equals to 625 KHz, is used to acquire the data samples from the radar simulator.

CLK2, equals to 1.25 MHz, is used to execute data interpolation, and data shifting processes.

CLK4, equals to 10 MHz, is used to select data for cross-product, to accumulate the data samples of the same time index output from cross-multiplication process, to perform the FFT calculations, and to accomplish the square root in the magnitude process.

CLK3, equals to 5MHz, is used to write the calculated WVD data in memories of data arrangement module. CLK5, equals to 384.61 KHz is used to read the data stored in the memories of data arrangement module before CFAR module, and to perform CFAR process.

3.3 FILTERS BANK OF THE MTD-SWVD

The simplified block diagram of the design of this part is shown in Fig.11. Explanation for each module in this figure is introduced in the following subsections.

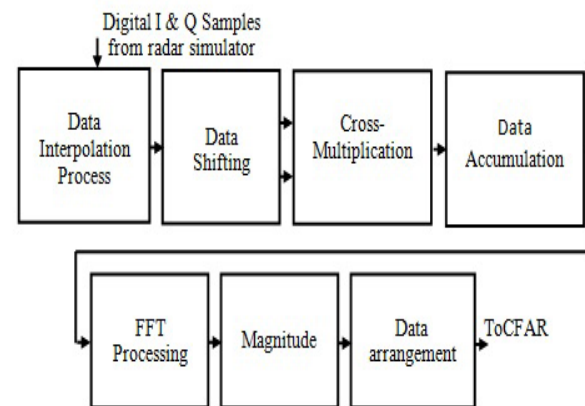


Fig. 11: Simplified block diagram of the designed Doppler filters bank for the MTD-SWVD

3.3.1 DATA INTERPOLATION MODULE

The data interpolation process intends to increase the number of data samples through zero padding between each two successive data samples. The I and Q samples input to the data interpolation module with a rate of CLK1 are arranged in groups of 8-samples. Each group is followed by a hold-on period of 4-clocks filled with zeros. The data interpolation process is accomplished with a clock rate of CLK2 which is twice the rate of the data input CLK1. Realization of data interpolation process on FPGA is done using multiplexer with CLK2 output select and a reset signal. The reset signal is used to reset the multiplexer to be ready for the next eight successive data samples. Then the process is repeated till the end of the CPI.

3.3.2 DATA SHIFTING

Data shifting operation is used to slide through the time index. Such that for each time of data shifting, one data sample is at the midpoint and all the data samples are valid for cross-product process. Fifteen 8-Bit data registers, two 8-to-1 multiplexers, and a 3-bit binary counter are used for data shifting implementation. Fig.12. shows a simplified block diagram of data shifting process.

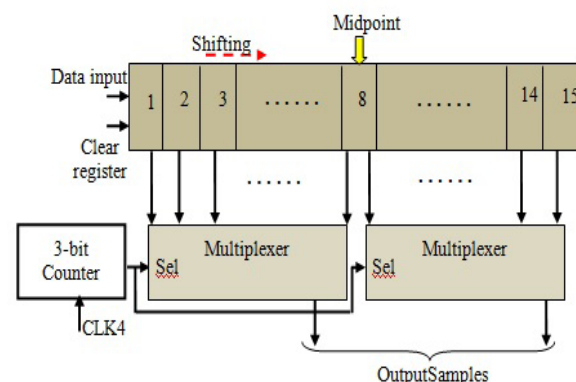


Fig. 12: Simplified block diagram of data shifting

The data is shifted through the shift registers with the same rate of the interpolation process, CLK2. During each time of data shifting, there must be fifteen datasamples valid for cross-multiplication process. A binary counter is used to determine the data output of the two multiplexers. The counter rate is selected to be eight times the rate of data shifting (CLK4); such that during one shift there are fifteen data samples at the output of the multiplexers valid for cross-product. A reset signal from the synchronizer is used to reset all the fifteen-shift register, for every new group of data.

3.3.3 CROSS- MULTIPLICATION PROCESS

Cross-multiplication is done by using Xilinx LogiCORE™ intellectual property IP Complex Multiplier core V3.1. The complex multiplier inputs are 8-bit complex data samples output from the data shifting multiplexers and its output is 17-bits complex data feed to data accumulator. To reduce complexity, only the first eight cross-product points, out of fifteen cross-product points are considered. This is because the 9th cross-product point is set to zero and the last seven points are the complex conjugate of the points from eight-to-two.

3.3.4 DATA ACCUMULATION PROCESS

An accumulator is designed for summing all cross-product points of the same time index to generate eight points out of the 16-points summed kernel at its output. The accumulator generates the remaining eight points of the summed kernel internally where the 9th data sample is equal zero and the data samples from ten-to-sixteen are equal to the complex conjugate of the data samples from eight-to-two. The output data from the accumulator is a serial 16-samples summed kernel, which are then fed to the FFT processor.

3.3.5 FFT PROCESSING

The Xilinx LogiCORE IP Fast Fourier Transform v7.1 is used to realize the block of the 16-point FFT processing. The FFT output appears after the application of the start signal by a constant latency of 5.75µsec. The FFT output is, real and imaginary parts, fed to the magnitude process.

3.3.6 MAGNITUDE PROCESS

The magnitude process is designed to get the absolute value of the FFT output. Performing a CFAR operation for each frequency bin across all range cells requires a data rearrangement before the CFAR operation.

3.3.7 DATA ARRANGEMENT

Data arrangement means that the output data from the magnitude stage is rearranged to be ready for CFAR processing such that, all range cells of the same frequency bin, follows each other in a queue to be passed to a CFAR processor. The data arrangement module consists of two Random-Access-Memories (RAMs), three 2-to1 multiplexers, two addressing counters, and timing unit shown in Fig.13.

This process is fully synchronized with the magnitude output. The concept of simultaneous processing is applied as follows: While RAM-1 is in write mode, it is addressed by writing-counter and clocked by clk_w. During this period, RAM-2 is in read mode and addressed by reading-counter at every clk_r. The reading counter counts in a way such that every 32 successive samples for the same frequency bin during the processing interval can be serially processed by CFAR processor. Output data from RAM2 is then passed through a multiplexer to the CFAR block. This process is reversed, each one completes CPI. Now, data is ready to be processed by the single CFAR circuit.

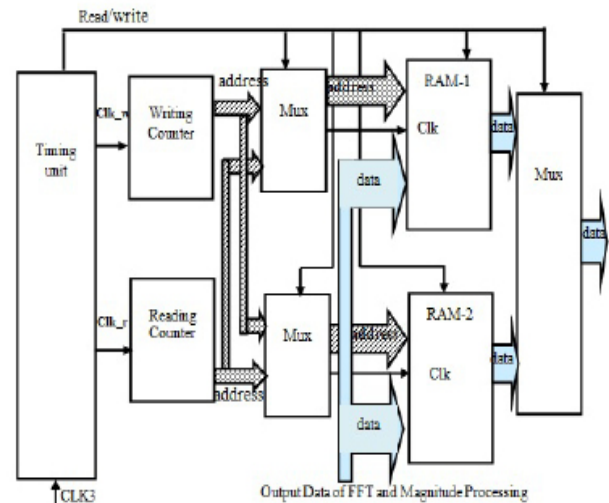


Fig. 13: Simplified block diagram of data arrangement module

3.4 CFAR MODULE

A cell average CFAR processor is designed to establish separate thresholds for each range cell as shown in fig.14. Here, two guard cells around the test cell are introduced such that the summation of leading and lagging windows is not affected by the target returns in the test cell.

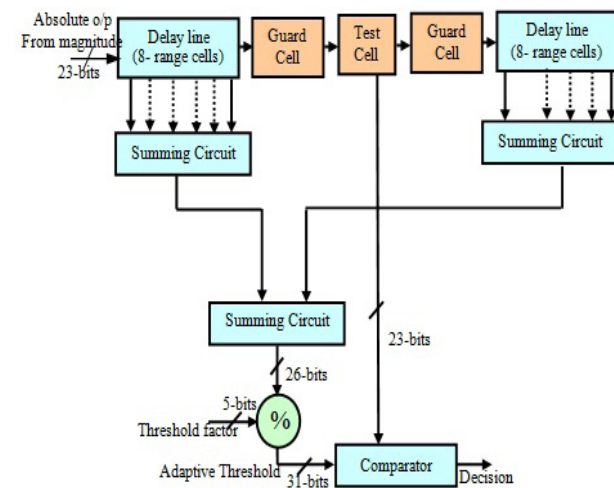


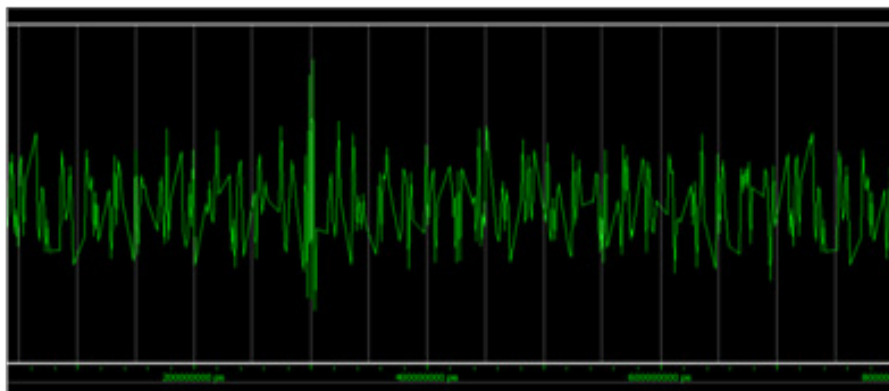
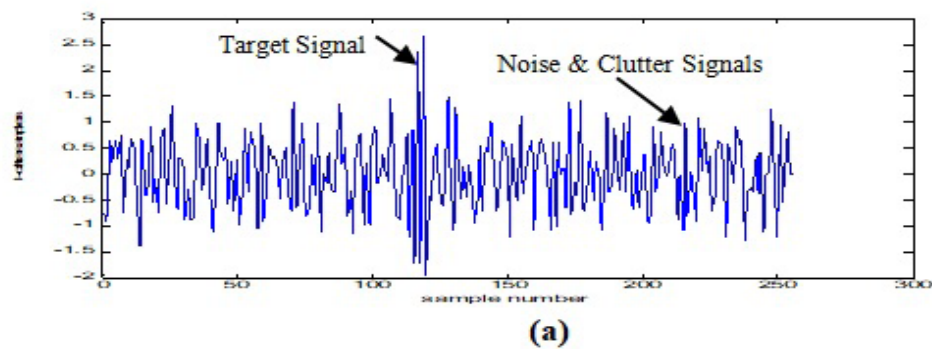
Fig. 14: Simplified block diagram of the implemented CA-CFAR processor

4. EXPERIMENTAL RESULTS

Output signals from the implemented MTD-SWVD processor is tested experimentally for the fetched signal from the radar simulator by monitoring, using the ChipScope software package. Figures 15-19 shows the result of matlab, Modelsim, and ChipScope, respectively, for a simulated radar signal. The signals displayed are: I and Q data samples, FFT output, arranged data –to- CFAR, and the final MTD-SWVD output. It is clear from these figures that simulated and experimental results are identical.

5. HARDWARE COMPLEXITY EVALUATION

Since, Hardware complexity is an important point to be taken into consideration when realizing any hardware system. A comparison between MTD-SWVD, the traditional MTD-WVD and MTD-I has been performed in terms of the used FPGA resources. However, Table 1 shows the number and percentage of the different consumed elements in the internal structure of the used FPGA device for each MTD schemes realized.

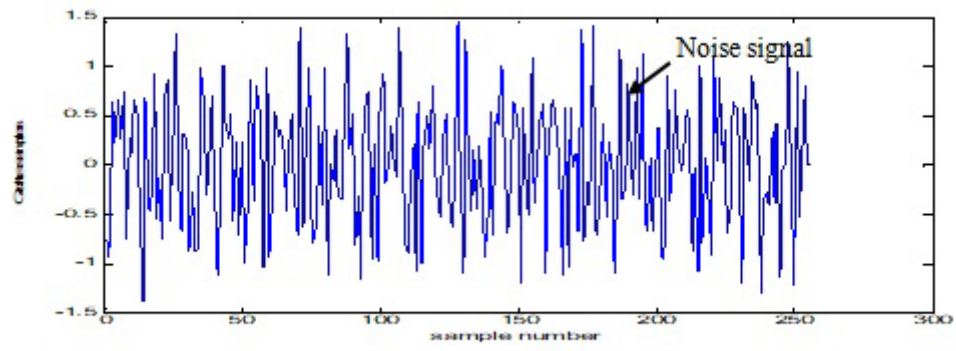


(b)

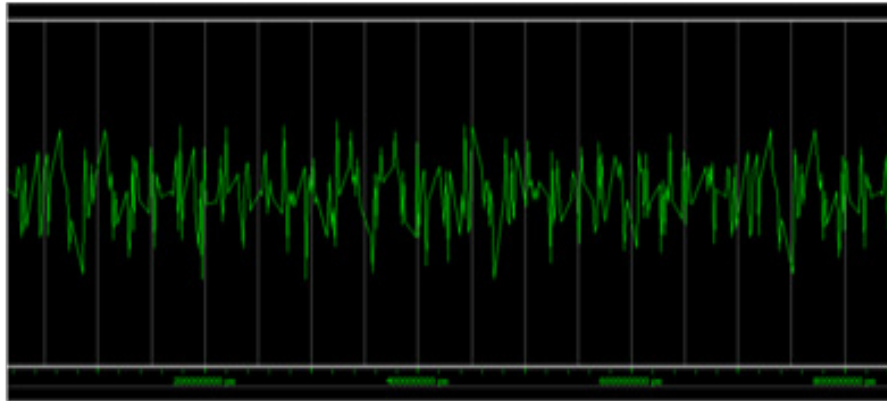


(c)

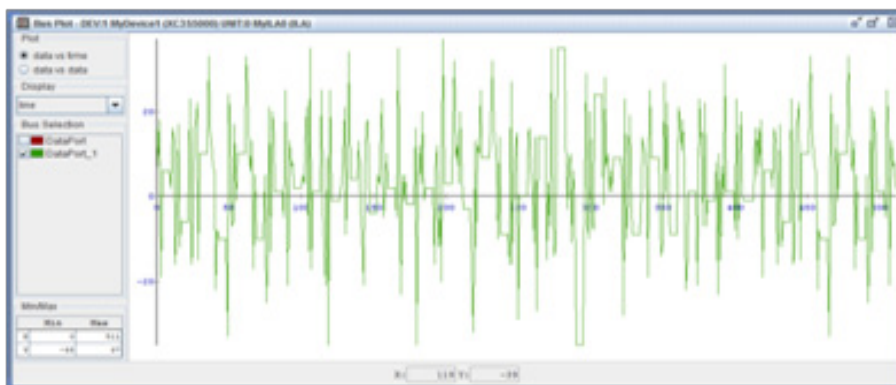
Fig. 15: I data samples; (a) Matlab result, (b) Modelsim result, and (c) Chipscope result



(a)



(b)



(c)

Fig. 16: Q data samples; (a) Matlab result, (b) Modelsim result, and (c) Chipscope result

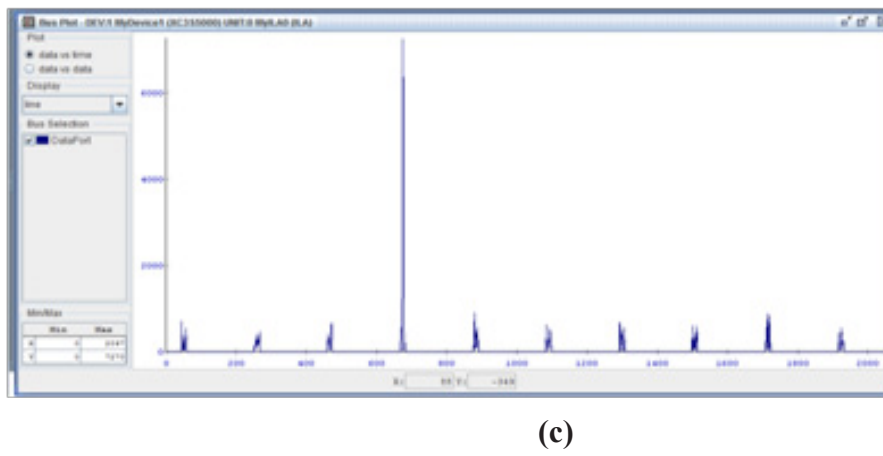
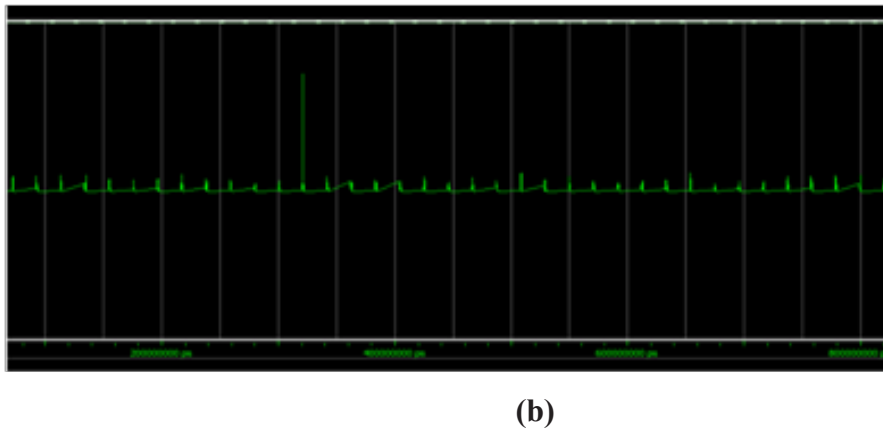
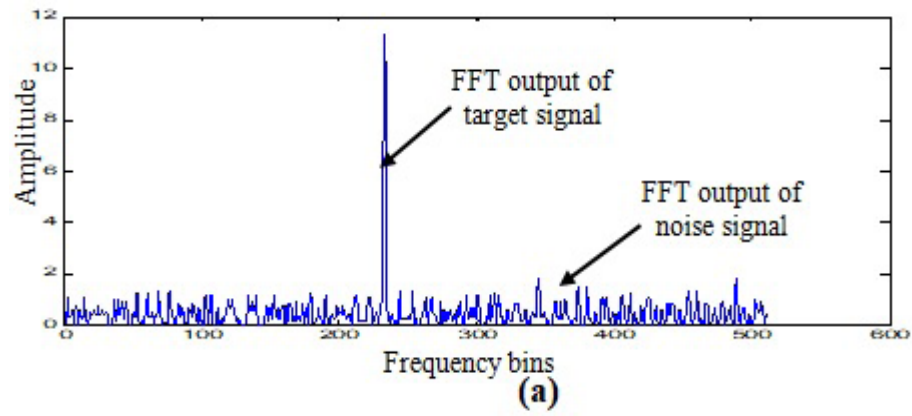


Fig. 17: FFT output; (a) Matlab result, (b) Modelsim result, and (c) Chipscope

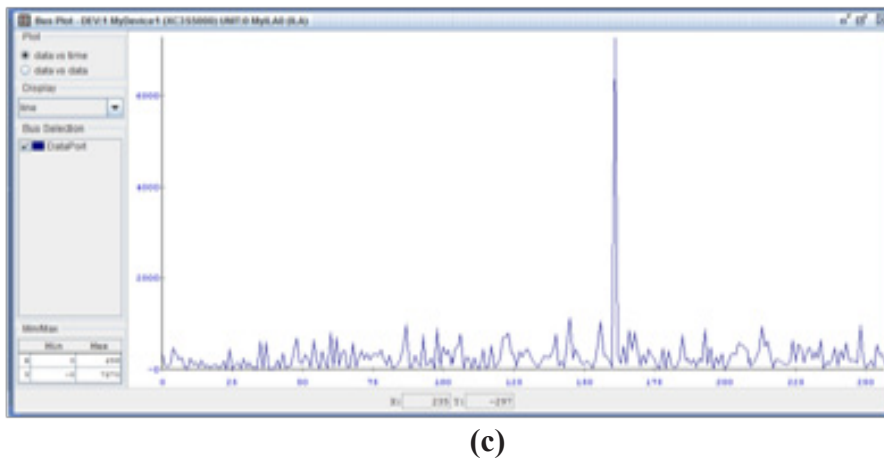
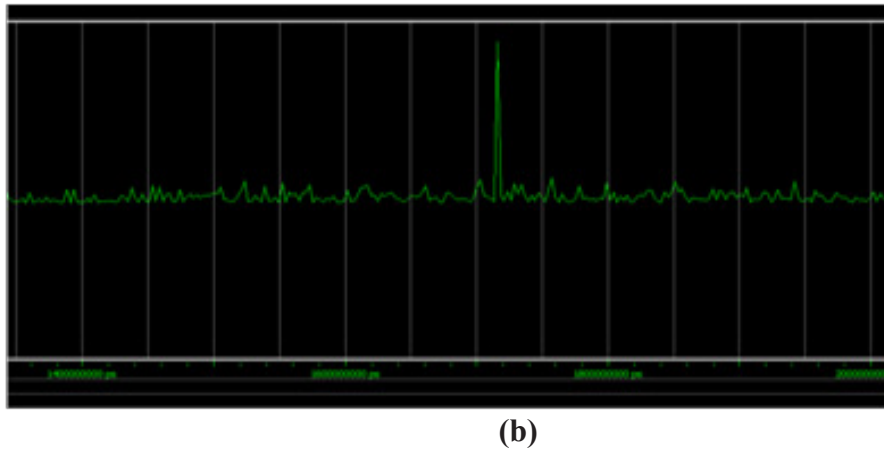
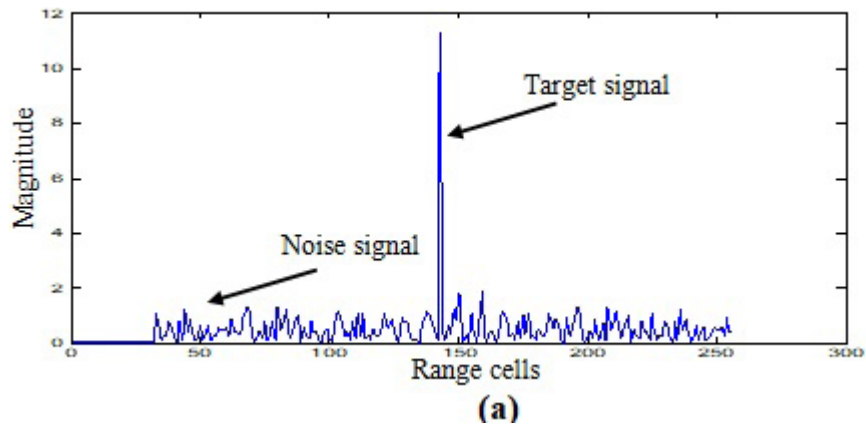
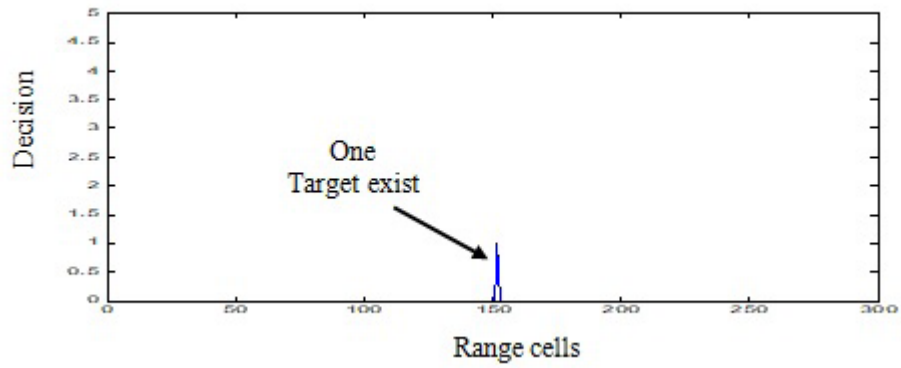
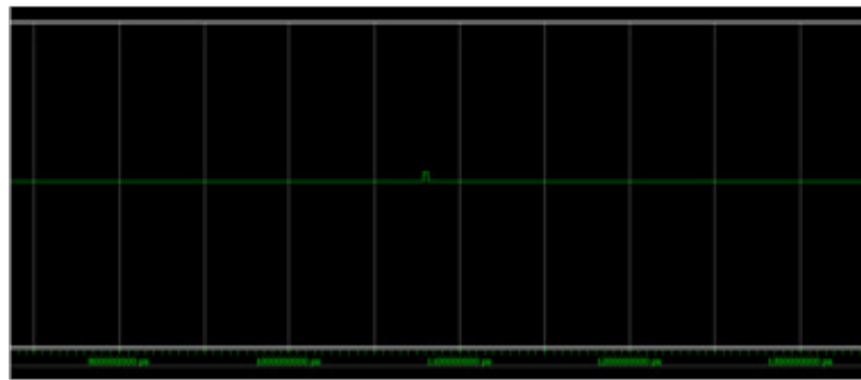


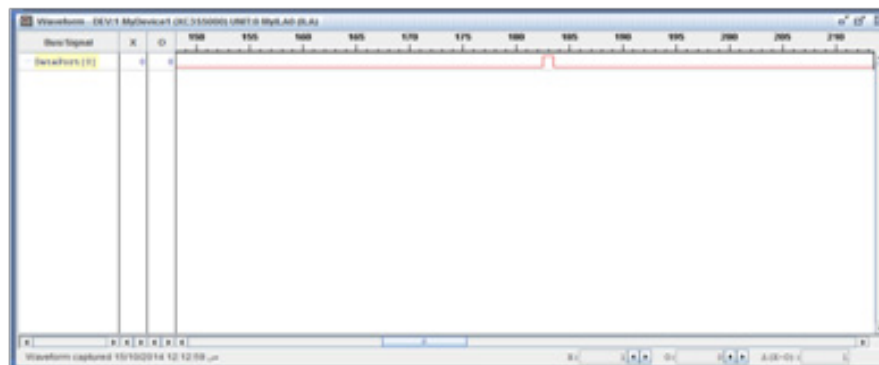
Fig. 18: Arranged data-to- CFAR; (a) Matlab result, (b) Modelsim result, and (c) Chipscope result



(a)



(b)



(c)

Fig. 19: Final output (Decision); (a) Matlab result, (b) Modelsim result, and (c) Chipscope result

Table1: The number and percentage of the different consumed elements in the internal structure of the used FPGA device for the three implemented MTD schemes (MTD-SWVD, MTD-WVD, and MTD-I).

| FPGA resources | Number of Slices | | Number of Flip Flops | | Number of 4 input LUTs | | Number of bonded IOBs | | Number of BRAMs | | Number of MULT18X18s | |
|----------------|------------------|-----|----------------------|----|------------------------|----|-----------------------|----|-----------------|----|----------------------|-----|
| | actual | % | actual | % | actual | % | actual | % | actual | % | actual | % |
| MTD Schemes | 33280 | | 66560 | | 66560 | | 633 | | 104 | | | |
| MTD-SWVD | 3715 | 11% | 5113 | 7% | 5034 | 7% | 3 | 0% | 6 | 5% | 29 | 27% |
| MTD-WVD | 2895 | 8% | 4181 | 6% | 3557 | 5% | 3 | 0% | 4 | 4% | 29 | 27% |
| MTD-I(8) | 881 | 2% | 1198 | 1% | 1243 | 1% | 3 | 0% | 4 | 4% | 5 | 4% |

6. CONCLUSION

In this paper, we have discussed the implementation issues of MTD-SWVD in details. Also, A comparison between MTD-SWVD, MTD-WVD and MTD-I has been introduced. We have found that MTD-SWVD has been implemented with extra hardware resources (11% of the used FPGA) than MTD-WVD (8% of the used FPGA) and MTD-I (2% of the used FPGA). This extra hardware complexity is accepted regarding to the achieved improved performance.

7. REFERENCES

[1] Ibrahim M Ali, Fathy M. Ahmed, "Application of a Wigner Ville Distribution Based method in Moving Target Detection," International Journal of Engineering and Advanced Technology (IJEAT) ISSN:2249-8958, volume-3, Issue-2, December 2013.
 [2] Muehe, C.E., "Moving Target Detector, an Improved signal Processor," Proceedings of International Conference ACARD. 1976, Vol. 195, pp. 14.1-14.10.
 [3] J. Giridhar, K. M. M. Prabhu, "Implementation of MTD-WVD on a TMS320C30 DSP Processor," in Microprocessor and Microsystems 22, 1998, 1-12.

[4] Merrill I. Skolnik, Introduction to radar systems, 3rd ed., McGraw-Hill, 2001.
 [5] P. Krishna Kumar, and K.M.M. Prabhu, "Simulation Studies of Moving Target Detection: a New Approach With Wigner-Ville Distribution," Proc. IEE, Radar, Sonar and Navigation, Vol. 144, No. 5, 1997, pp. 259-265.
 [6] A.V. Oppenheim, R.W. Schaffer, Discrete-Time Signal Processing, Prentice. Hall Inc., Englewood Cliffs, NJ, 1989.
 [7] K.M.M. Prabhu, P. Krishna Kumar, On the classification of radar returns using WVD, Welch and AR methods, Applied Signal Processing 3 (1996) 199-211.
 [8] P. Krishna Kumar, K.M.M. Prabhu, Classification of radar returns using Wigner-Ville distribution, Proceedings of ICASSP-96, May1996, Atlanta, GA, pp. 3105-3108.
 [9] AbeyAsekera, S.S., "Computation of Wigner-Ville distribution for complex data," in Elertvon. Lett. 1990, 26, (16), pp. 1315- 1317.
 [10] Shahida G. Qadir, Yangyu Fan, and Fathy M. Ahmed., "A Wigner Ville Distribution Based Method for Detection of Gaussian Contaminated Sinusoidal Signal in Frequency Domain," in PIERS Proceedings, Marrakesh, MOROCCO, March 20-23, 2011
 [11] Technical Manual, "xilinx ISE 13.2 software manuals and help -pdf collection", Xilinx, Inc.
 [12] Technical Manual, "Xilinx ISE 13.2 In-Depth Tutorial", Xilinx Inc.
 [13] Operational Manual, "Spartan-3 FG900 protoboard", Xilinx Inc.
 [14] Http: // www .xilinx .com / products / devkits / HWSPAR3-SK-UNI-G.htm.
 [15] Mitchell, R.L. Radar Signal Simulation, (Artech House, 1976).