

## Research Article

# Area Efficient Implementation of MTI Processing Module on a Reconfigurable Platform

**Munaza Yousuf, Arshad Aziz, and Riaz Mahmud**

*National University of Sciences and Technology (NUST), Pakistan Navy Engineering College, Karachi 75350, Pakistan*

Correspondence should be addressed to Munaza Yousuf; [munazay@yahoo.com](mailto:munazay@yahoo.com)

Received 21 October 2013; Accepted 10 November 2013; Published 12 January 2014

Academic Editors: Z. Gu, C.-Y. Sun, and L. Tan

Copyright © 2014 Munaza Yousuf et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an area efficient Field Programmable Gate Array (FPGA) based digital design of a processing module for MTI radar. Signal contaminated with noise and clutter is modelled to test the efficacy of the design algorithms. For flexibility of design and to achieve optimized results, we have combined the high-level utility of MATLAB with the flexibility and optimization on FPGA for this implementation. Two- and three-pulse cancellers are chosen for design due to its simplicity in both concept and implementation. The results obtained are efficient in terms of enhanced throughput per Slice (TPA) of 1.146, that is, occupying fewer area resources on hardware while achieving optimized speed. The outcomes show that this design of MTI radar processor has many advantages, such as high processing precision, strong processing ability, real time, and low cost. All these advantages greatly contribute to the design requirements and make it appropriate for the application of high-speed signal processing.

## 1. Introduction

In radar systems, it is often desirable to be able to differentiate between moving targets and those that are stationary. This information can greatly serve operators when the surveillance targets of interest, such as vehicles or aircraft, are located in environments with a high density of traffic, clutter, stationary, and slow moving targets. Fortunately, through the use of digital signal processing, it is possible to extract this information from the system in real time using a Moving Target Indicator (MTI) filter [1]. Modern radar systems have evolving requirements, both in how the systems are designed and how the end user uses the data. This results in design change in electronic systems affecting both the military and commercial design communities, that is, the need for smaller, energy-efficient systems with high processing-power requirements. This makes low power consumption key drivers in most designs of high speed systems. More digital logic also allows designers to make early decisions on actionable intelligence and to meta-tag sensor data earlier for more efficient analysis. These and other emerging techniques will allow for the creation of better radar systems, but each requires additional signal-processing resources. One of these resources is the emerging class of high-performance FPGAs.

Moving Target Indicator (MTI) radars are developed and used extensively to detect and follow specific moving targets and eliminate clutters. Processing system in the radar is massive and complex, since it is required to perform at high processing speed. Improvement in FPGAs component density and performance capability has resulted in more signal-processing functions being incorporated and migrated to the front end of the radar system containing the exciter/receiver [2]. In past systems, conventional digital signal processors were used to perform many of these algorithms. However, Field-Programmable Gate Arrays (FPGA) deliver an order of magnitude higher performance than traditional DSPs. Several DSP functions are needed for radar signal processing in the receiver. Each function should be closely examined to determine whether the application will show substantial speed and performance improvements through implementation in an FPGA. These functions can be efficiently implemented using an FPGA embedded processor, even for highly complex and adaptive operations. Our target will be to combine the capabilities of hardware and software to implement the original MTI receiver to generate a true bit accurate model and validate it using Simulink. The contribution factor of this paper is the design strategy that invests more time in thorough simulation based verification of design concepts

to save time rather than on postsynthesis issues. This focus has helped us simplify the design to more optimal levels on reconfigurable hardware, that is, FPGA. This enables us to minimize the hardware delivery time by translating the concept into a fully functional, implementation ready design optimized for target embedded platform as per requirement.

The rest of the paper is arranged as follows. In Section 2 a detailed literature review has been given that discusses the contribution factor of this work as compared to previous implementations. Section 3 describes the basic characteristics of MTI processing. Section 4 covers design architecture on FPGA. This also includes the integration of software and hardware platforms to achieve optimized results. Section 5 includes results of design implementation. Section 6 pertains to the conclusions that are arrived at, after combining the software and hardware computations.

## 2. A Literature Review

High-speed, real time, and accuracy have always been the significant requirements of radar signal processing in its application area. Using FPGA cannot only meet these needs, but also improve reliability of system and strengthen universality of unit, cut cycle, and cost of the system development, and so forth. This paper has completed the design of an MTI radar module in which a FPGA is the core, primarily elucidating the principles of radar applications, filtering amplification of radar signal processing,  $A/D$  sampling, algorithm and FPGA realization. The literature survey does not reveal any application that discusses a completely FPGA based implementation of an MTI processor. However, some implementations present FPGA's work alongside DSPs that is a semiparallel architecture to perform MTI processing, implemented by Ali et al. [3]. An architecture utilizing a DSP-FPGA combination for clutter processing has been proposed by Xu et al. [4], which involves the FPGA writing the successive returns into an SRAM chip and DSP performing the computations. Different FPGA implementations for FIR filters have been proposed in [5, 6]; both of these implementations focus primarily on area reduction while trying to achieve reasonable speeds. Stapleton et al. [7] have examined the possibilities of enhancing radar signal processing algorithms using FPGAs. Li and Wang [8] have given radar signal processor design on FPGA that focuses on filtering amplification,  $A/D$  sampling algorithm, and realization of FFT. Reference [9] has proposed a timeshared approach requiring about 1.2% slices of an XC2V6000 for implementation of a dynamically reconfigurable Pulse Doppler radar in a mixed system comprising a DSP and FPGA to perform linear frequency modulation. Reference [10] has given implementation of an IIR filter based moving target indication (MTI) processor but there is a downside to IIR filters in that they do not guarantee a linear phase response. Moreover, stability issues have to be looked after. Bin Khalid et al. [11] have proposed an FPGA based real-time signal processor for Pulse-Doppler radar. Hence it is observed that the radar processors have been implemented either entirely on hardware or entirely on software. The results of this research work done contribute to integrate the individual capabilities of both hardware and

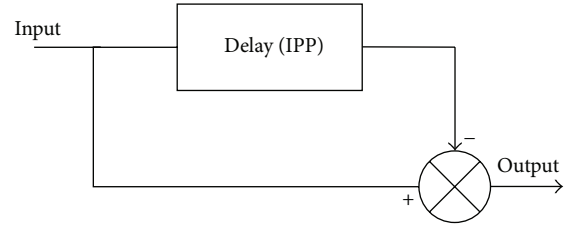


FIGURE 1: MTI pulse canceller.

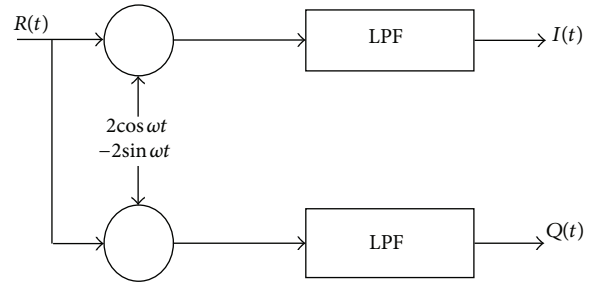


FIGURE 2: Synchronous detection of Doppler-shifted pulse return.

software to achieve optimized results. The conclusions show that this design is advantageous in terms of speed and efficient device utilization.

## 3. MTI Processing

Doppler processing can be divided into two major classes: Pulse Doppler processing and Moving Target Indication (MTI). When only target detection is of concern, the MTI filter is usually adequate. MTI filters even a first or second order FIR high-pass filter such as 2- or 3-pulse MTI cancellers can be used to filter out the stationary clutter. Figure 1 shows a typical MTI Pulse canceller. Higher order filter types are used as they only provide modest performance improvements over the pulse cancellers [12]. The synchronous detection in Figure 2 shows the complex envelope during the coherent local oscillator demodulation stage in the receiver. Quadrature demodulation avoids loss of signal due to blind phase effects of a single demodulator as well as resolving the phase function of the complex envelope. Once the synchronous detection of the complex envelope is complete, it is processed in two stages: high-pass FIR filters with sampling at the pulse repetition rate and a matched correlator for return code (target) detection with video  $A/D$  sampling at the bit duration interval.

The resulting complex signal will be filtered in both the in-phase and quadrature-phase components as defined in:

$$\begin{aligned}
 IpI(n) &= \cos(2\pi f_d \cdot nt_{pri}) \\
 IpQ(n) &= \sin(2\pi f_d \cdot nt_{pri}) \\
 OpI(n) &= \sin(2\pi f_d \cdot nt_{pri}) \\
 OpQ(n) &= \cos(2\pi f_d \cdot nt_{pri}).
 \end{aligned} \tag{1}$$

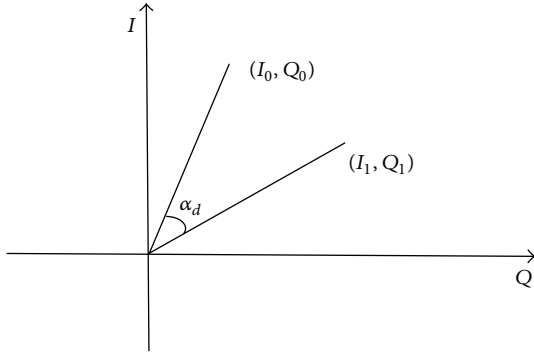


FIGURE 3: Quadrature Doppler phase angle measurement.

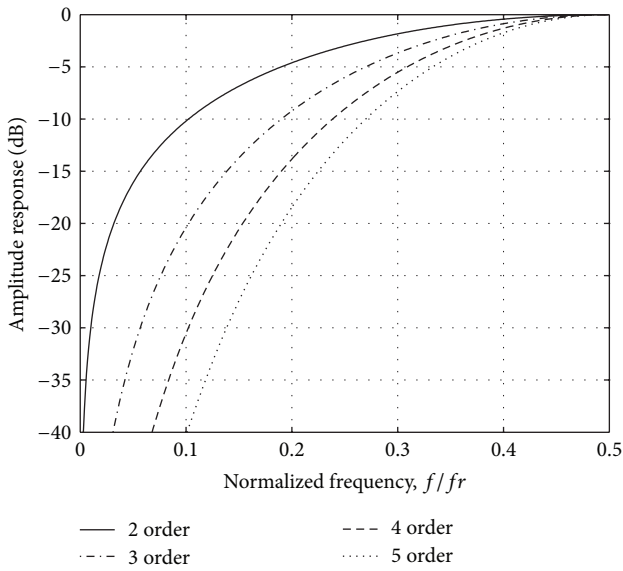


FIGURE 4: MTI high-pass filter response versus normalized frequency.

A steadily moving target moving at  $V_d$  will Doppler shift by  $\omega_d$  the carrier frequency  $\omega_c$ , thereby altering the complex envelope of the radar waveform. This is defined in (2), where  $c$  is the speed of light. The Doppler frequency is simply the time derivative of the complex envelope's phase function, thus emphasizing constant receiver phase offset. This Doppler shift is found by sampling the quadrature phase angle at the pulse repetition frequency. It is this angle  $\theta$ , defined in (2) and illustrated in Figure 3 that is measured using consecutive  $IQ$  samples at the pulse repetition interval:

$$f_d = \frac{2F_t}{c} V \cos \theta. \quad (2)$$

The set of relations as defined in [1] best describe the phenomena. The reference signal and the target echo signal are mixed in the mixer stage of the receiver. Only the difference frequency voltage component from the mixer is of interest:

$$v_d = V_d \sin \left[ \omega_d t - \frac{\omega_0 R_0}{c} \right]. \quad (3)$$

The fundamental MTI filter component is the high-pass FIR feed-forward pulse canceller. These filters are guaranteed to be stable, that is, all zeroes and no poles in the transfer function. They are easy to implement in direct form using a tapped delay line. A two-pulse canceller as shown in Figure 1 is used if the clutter component (assuming DC only) remains constant in a given range bin and can be eliminated by subtracting the output from two successive pulses. The transfer function of two-pulse canceller is equal to  $1 - 1/z$  and is equivalent to FIR digital filter with magnitude response  $\sin(\omega/2)$ . In practice, the clutter has a power spectrum that covers frequencies above DC. The two-pulse canceller will attenuate low frequency components but may not totally reject clutter. A three-pulse canceller with its transform function equivalent to FIR filter is given in (7). This decreases further the components near DC [13]. Figure 4 shows the unity gain frequency response of the filter:

$$h(t) = \delta(t) - 2\delta(t \cdot T) + \delta(t \cdot 2T). \quad (4)$$

Signal-to-Clutter Ratio (SCR) instead of Signal-to-Noise Ratio (SNR) determines the RADAR's capability to detect targets in a high clutter background. Normally, clutter signal level is much higher than the receiver noise level because clutter echoes are random and have thermal noise-like characteristics. Grazing angle, surface roughness, and the RADAR wavelengths mainly affect the amount of clutter in the RADAR operation [14]. Differentiating target returns from clutter echoes is based on the target RCS and the estimated clutter RCS. Clutter RCS can be defined as the equivalent radar cross section endorsed to reflections from a clutter area,  $A_c$ . The average clutter is given by

$$\sigma^c = \sigma^0 A_c, \quad (5)$$

where  $\sigma^0$  (dB/m<sup>2</sup>) is the clutter scattering coefficient.

The clutter area  $A_c$  and SCR are defined as

$$A_c = \left( R \theta_{3\text{ dB}} \frac{c\tau}{2} \sec \Psi_g \right), \quad (6)$$

$$\text{SCR} = \frac{2\sigma_t \cos \Psi_g}{\sigma^0 \theta_{3\text{ dB}} R_{c\tau}}. \quad (7)$$

Clutter-to-Noise (CNR) is defined as

$$\text{CNR} = \frac{P_t G^2 \lambda^2 \sigma_c}{(4\pi)^3 R^4 K T_0 B F L}, \quad (8)$$

where  $P_t$  is the peak transmitted power,  $G$  is the antenna gain,  $\lambda$  is the wavelength,  $K$  is Boltzmann's constant,  $T_0$  is the effective noise temperature,  $B$  is the radar operating bandwidth,  $F$  is the receiver noise figure, and  $L$  is the total radar losses.

For Gaussian clutter, the clutter and noise can be combined and RADAR measurement can be derived from the signal-to-clutter + noise ratio as given by

$$\text{SIR} = \frac{1}{(1/\text{SNR} + 1/\text{SCR})}. \quad (9)$$

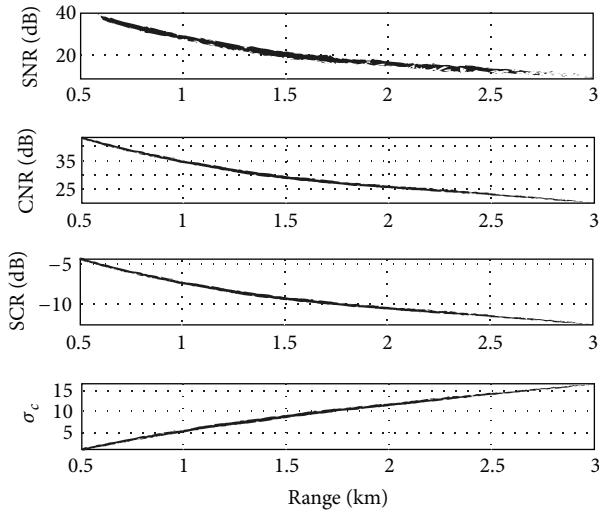


FIGURE 5: Variation of SCR, SNR, and CNR with distance (km).

MATLAB simulations considering clutter have been carried out and the results are shown in the Figure 5. The simulation parameters are written below

$$\begin{aligned} \sigma_t &= 0 \text{ dBsm}, \\ \sigma^0 &= -30 \text{ dBsm}, \\ \text{Antenna Sidelobe Level} &= -30 \text{ dBsm}, \\ \text{Antenna and Target Height} &= 10 \text{ m}. \end{aligned} \quad (10)$$

The optimal goal is to concentrate on clutter rejection; hence, the desired SNR as shown in the top of the graph is quite high. It is also clear that the SCR is well below the SNR level.

#### 4. Design Implementation

The workflow demonstrated by Figure 6 starts by reading the target information from an array stored in the workspace which in this case is the speed of the target. A MATLAB-based mathematical model of the system has been developed and investigated. Part of the data, received as a result from these investigations, has been used for the simulation of the circuit [15] and synthesized in FPGA to verify if the system meets the real-time constraints. The analysis of the results shows their complete coincidence with those received from the mathematical simulation. The gateways are used to do conversion from floating point data type to fixed point data type and vice versa.

The main advantage of proposed structure, that is, the hybrid of software and FPGA hardware, is the capacity whereupon we can easily change the operating factors of any block of the system. If it is required in some applications to increase the precision of the overall system, then the increase in number of bits can be realized in each block hence giving us the flexibility to adjust the parameters of the system and therefore better solutions. The parameters of the design have been chosen carefully so that the radar system is operating

TABLE 1: Process parameters.

Parameter	Value
Operating band	L (1.0-2.0 GHz)
Operating frequency range (MHz)	1215-1400
Nominal operating frequency $f_0$	1250 MHz
Range resolution (m)	150
Receiver bandwidth (MHz)	1
Pulse width $\tau$ ( $\mu\text{s}$ )	1
Unambiguous range (nm)	100
PRF (Hz)	800
PRF Mode	Single
PRI (msec)	1.25
Noise figure (dB)	10
Antenna gain (dB)	35
Unambiguous range (nm)	100
$P_d$	0.9
$P_{fa}$	$1e-6$

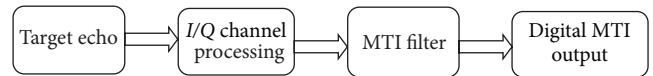


FIGURE 6: Flow chart of design algorithm.

in the  $L$  band of the electromagnetic spectrum. Table 1 shows process parameter assessed and refined according to design approach.

The core of this design relies on the fact that MTI systems contain coherent pulse train signals. The output of the phase detector for moving target varies in amplitude from pulse to pulse as the phase changes between the transmitted and received signals. However there is no change in phase for fixed objects. By making use of this approach signal processing technique like delay line cancellers are used to implement MTI systems. A stationary target at the range of 60 km has been placed to simulate the effect of clutter rejection filter. Also, additive white Gaussian noise (AWGN) has been added to account for receiver noise introduced in actual systems. The simulated radar parameters yield an unambiguous range of approximately 100 Nm and detection of targets with maximum radial velocity of no less than 310 m/s. The computational block for the radar signal processor is implemented on Xilinx FPGA Spartan-III E Xc3s500e using Verilog HDL (Hardware Descriptive Language). Fixed point implementation has been done. Each memory location was 16 bit wide; lower 8 bits to store the imaginary part and upper 8 bits to store the real part. The block diagram of radar system implemented on FPGA is shown in Figure 7. The user defined precision has been set for the addition blocks with output type as signed 2's complement 16 bit number with 14 binary points. The implementation field is set by selecting fabric as core parameter. The solver used for simulation is fixed step discrete solver and simulation step size is  $1.25e-007$  for this design.

The received echo is fed to the phase detector for  $I/Q$  channel processing. The  $I/Q$  demodulation stage is most

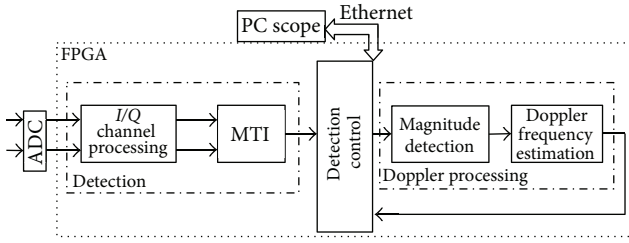


FIGURE 7: Block diagram of hardware implementation.

commonly digitally implemented as either a Hilbert transform or an in-phase and quadrature mixing operation. The mixing operations are simply digital multiplications by sine and cosine [16]. These are further illustrated by pseudocodes in the following listings.

(A) Pseudocode

$$\text{loop}i = 0 \text{ to } P - 1,$$

$$\text{loop}j = 0 \text{ to } 2 * N - 1,$$

$$\text{input}_{\text{re}[i][j]} = \text{input}[i][j] * \sin\left(2 * \text{pi} * f_{\text{IF}} * \frac{j}{f_{\text{ADC}}}\right),$$

$$\text{input}_{\text{im}[i][j]} = \text{input}[i][j] * \cos\left(2 * \text{pi} * f_{\text{IF}} * \frac{j}{f_{\text{ADC}}}\right), \quad (11)$$

where  $P$  is the number of pulses per burst and  $N$  is the number of range bins per pulse.  $f_{\text{IF}}$  and  $f_{\text{ADC}}$  are sampling frequency of intermediate frequency (IF) and ADC respectively.

(B) Pseudocode for Two-Pulse Canceller

$$\text{loop}i = 0 \text{ to } N - 1,$$

$$\text{loop}j = 0 \text{ to } P - 1,$$

$$\text{mti}[i][j] = n[j][i] - \text{temp}, \quad (12)$$

$$\text{temp} = n[j][i].$$

The processing requirements for the two-pulse canceller are thus  $N \times P \times 2$  memory reads, subtractions, and memory writes, while the 3-pulse canceller requires an additional  $N \times P \times 2$  multiplications and additions [16].

(C) Pseudocode for Three-Pulse Canceller

$$\text{loop}i = 0 \text{ to } N - 1,$$

$$\text{loop}j = 0 \text{ to } P - 1,$$

$$\text{mti}[i][j] = n[j][i] - 2 * \text{temp}1 + \text{temp}2, \quad (13)$$

$$\text{temp}2 = \text{temp} \quad \text{temp}1 = n[j][i],$$

where  $n$  corresponds to the present pulse.

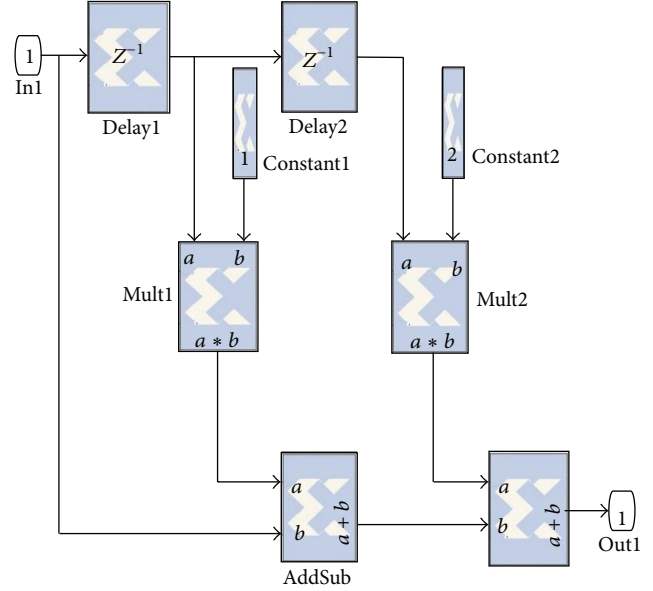


FIGURE 8: Design 1 (3-pulse canceller).

Our design module 1 implements MTI filter using three-pulse canceller realizations as shown in Figure 8. The three-pulse canceller is a popular and simple MTI filter. It is an all-zero FIR filter with filter coefficients 1, -2, and 1. Design module 2 implements MTI filtering using single delay line canceller also known as two-pulse canceller as shown in Figure 9. The system throughput is one sample per clock cycle. According to minimum period of 9.787 ns, the maximum frequency is 144.8 MHz.

### 5. Results

We have successfully implemented our design module on FPGA. The design takes the target information and approximated Doppler speeds within acceptable threshold. On implementation of designs, in Design module 1 the clock rate achieved is higher while Design module 2 turns out to be an efficient approach in terms of achieving effective area utilization on FPGA. Also the results obtained are resourceful in terms of enhanced throughput per slice, that is, occupying fewer area resources on hardware while, achieving optimized speed, TPA of 1.146 for our design module 2 is much better than other implementations [3], which is considered as a significant achievement. Bin Khalid et al.'s [11] implementation on Xilinx Virtex-II Pro (ML300) took less than 90  $\mu\text{s}$  hence resulting in slower performance of overall system. These premises give us a subjective idea of the performance since there are not similar architectures reported. Table 2 gives a comparative analysis of the device utilizations and clock details of our designs with other works [3, 9, 11]. Results indicate that our design gives more area optimization. The filtered signal by the proposed architecture is shown in Figure 10 with clutter being suppressed.

TABLE 2: Implementation results.

Implementation	Device	Clock rate (MHz)	FPGA area (slices)	Throughput/slice (TPA)
Our design 1	Xc3s500e	144.8	1336	0.108
Our design 2	Xc3s500e	122.639	<b>107</b>	<b>1.146</b>
Other work [3]	Virtex-4 SX35	183.55	1075	0.170
Other work [3]	Virtex-4 SX35	177.70	1382	0.128
Other work [11]	Virtex-II Pro (ML-300)	0.11	1468	—
Other work [9]	XC2V6000	102.17	811	—

Bold values emphasize superiority of results.

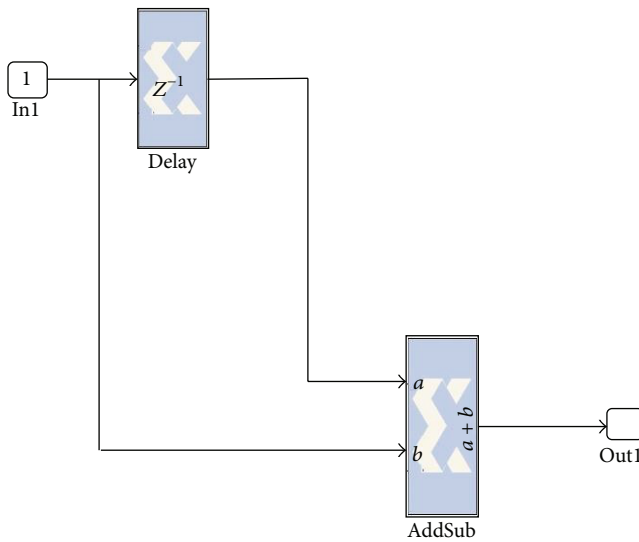


FIGURE 9: Design 2 (2-pulse canceller).

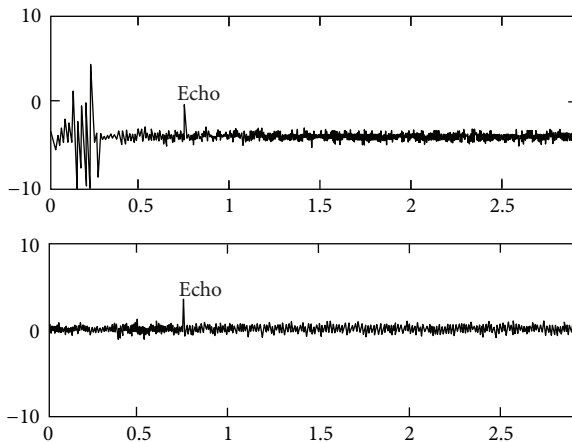


FIGURE 10: MTI filter output.

## 6. Conclusions

Our research emphasizes on designing digitized MTI processing modules that have been efficiently tested and verified on FPGA. This research contribution has integrated the capabilities of both hardware and software to achieve

optimized results both in terms of device utilization and maximum achievable frequency. In the future, this work can be optimized further by carrying out all the phases of design on Verilog HDL. The Verilog HDL code can be optimized to achieve the desired results in lesser time and fewer storage elements. Also, the newer FPGA families being introduced in the market offer better performance and more storage elements to enhance performance and yield better results.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of the paper.

## References

- [1] A. I. Zverev, "Digital MTI radar filters," *IEEE Transactions on Audio and Electro Acoustics*, vol. 16, no. 3, pp. 422–432, 1968.
- [2] FPGA signal processing for radar/sonar applications, Dec 1, 2007, By Ryan Kenny © Penton Media Inc, 2009.
- [3] Z. Ali, A. Arshad, and U. Razzaq, "An FPGA based semi-parallel architecture for higher order Moving Target Indication (MTI) processing," in *Proceedings of the 21st International IEEE Workshop on Rapid System Prototyping (RSP '10)*, pp. 1–7, Fairfax, Va, USA, June 2010.
- [4] S. Xu, M. Li, and J. Suo, "Clutter processing for digital radars based on FPGA and DSP," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '09)*, pp. 1–4, Nanjing, China, November 2009.
- [5] D. S. Dawoud and S. Masupe, "Design and FPGA implementation of digit-serial fir filters," in *Proceedings of the IEEE 7th AFRICON Conference in Africa: Technology Innovation (AFRICON '04)*, pp. 203–209, Gaborone, Botswana, September 2004.
- [6] C. H. Dick and F. Harris, "Implementing narrow-band FIR filters using FPGAs," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '96)*, pp. 289–292, May 1996.
- [7] R. Stapleton, K. Merranko, C. Parris, and J. Alter, "The Use of field programmable gate arrays in high performance radar signal processing applications," in *Proceedings of the IEEE International Radar Conference*, pp. 850–855, May 2000.
- [8] S. Li and Y. Wang, "The design of radar signal processor based on FPGA," in *Future Wireless Networks and Information Systems*, Y. Zhang, Ed., vol. 143 of *LNS in Electrical Engineering*, pp. 283–288, Springer, Berlin, Germany, 2012.

- [9] S. Sumeem, M. Mobien, and M. I. Siddiqi, "A pulse doppler radar using reconfigurable computing," in *Proceedings of the IEEE Proceedings of 7th International Multi Topic Conference (INMIC '03)*, pp. 213–217, 2003.
- [10] A. Arshad, F. Ahsan, Z. Ali, U. Razzaq, and S. Sajid, "An FPGA based architecture for Moving Target Indication (MTI) Processing using IIR Filters," *International Journal of Computer and Electrical Engineering*, vol. 4, no. 5, pp. 679–683, 2012.
- [11] F. Bin Khalid, R. A. Amjad, and M. A. Chohan, "FPGA based real-time signal processor for pulse doppler radar," in *Proceedings of the IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision (ICIEV '12)*, pp. 362–366, Dhaka, Bangladesh, May 2012.
- [12] B. R. Mahafza and A. Z. Elsherbeni, *Matlab Simulations for Radar Systems Design*, Chapman & Hall/CRC CRC Press, LLC, 2004.
- [13] Lab Volt, *Analog MTI Processing, Telecommunications Radar, Students Manual, 29279-00*, Lab Volt, Farmingdale, NJ, USA, 1st edition, 1990.
- [14] D. Kandar, S. N. Sur, D. Bhaskar, A. Guchhait, R. Bera, and C. K. Sarkar, "Implementation of MTI based Pulse compression Radar system using DSRC communication channel," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 379–382, 2010.
- [15] M. Yousuf, R. Mahmud, and A. Aziz, "Simulation design of an efficient MTI processing module for embedded platform," in *Proceedings of the IEEE 2nd Mediterranean Conference on Embedded Computing (MECO '13)*, pp. 75–78, Budva, Montenegro, June 2013.
- [16] R. Broich and H. Grobler, "Analysis of the computational requirements of a pulse-doppler radar signal processor," in *Proceedings of the IEEE Radar Conference*, pp. 08357–084011, Atlanta, Ga, USA, May 2012.

