Small RCS targets detection based on pulse compression radar signal generator and processor

Small RCS targets detection based on pulse compression radar signal generator and processor

A G Zahra, A Youssef, M Talha, F M Ahmed, and W Mehany.
Radar Department, Military Technical College, Cairo, Egypt.
E-mail: alaa.zahra977@mtc.edu.eg

Abstract. In the last two decades, field programmable gate arrays (FPGAs) have become the leading technology for implementing real time systems, specially radars. The most important advantage of FPGA is its parallel nature of executing complicated algorithms. As a result, the received data flow will be processed in a real time manner, which is suitable for radar applications. This paper attempts to show the design and the FPGA implementation of a binary phase-coded pulse compression radar signal generator and processor, that are intended to determine the ranges, and the speeds of targets with small radar cross-section area like drones. The system operating waveform is a bi-phase sequence, that is known to have the best known peak sidelobe level compared to compound Barker codes of the same length. In addition, the design incorporates a range sidelobes cancellation method to remove the time sidelobes in the matched filter output. Also, we are going to represent and investigate the FPGA implementing steps of the radar waveform generator and processor according to the required parameters. Finally, the implementation results of the proposed system will be represented.

1. Introduction

Over the past 15 years, drones have become one of the most critical technologies that had gained a dramatic attention, because of their several usages in various fields as medical, civilian, military and many other sides. Along with the excellent abilities and advantages of drones, they have dangerous usages such that they can be used for drugs and explosives smuggling, terrorism and espionage. That is why, researchers are obliged to find solutions for limiting their illegal usages by developing and enhancing sensors' performance for detecting them accurately [1].

Nowadays, radars are considered as the most fundamental sensor used for detecting targets and estimating their velocities. This is a result of having a superior performance over visual detection sensors in different conditions. Radars are classified into different categories based on the used waveform, the carrier frequency band, the operating field, etc. Regarding the operating waveform, radars are classified to continuous wave radars and pulsed radars. In pulsed radars, it is desirable to obtain a fine resolution in conjunction with transmitting high average power, but both parameters are inversely proportional to each other. The pulse compression had enabled radars to take the advantage of both parameters at the same time. The pulse compression algorithm enables good range resolution levels while transmitting wide pulses intended for long range detection. Phase-coded (PC) signals and linear frequency-modulated (LFM) waveforms are the most commonly used pulse compression radar signals, because of their simple generation and their excellent auto-correlation functions (ACFs) [2], [3], [4].

In [5], a traditional implementation of a pulsed radar processor based on Barker code is presented. In this design, the received signal is fed to a matched filter, then a Hamming window was used to attenuate the sidelobes existed in the ACF. As a result, the ACF main lobe amplitude is attenuated, and gets widened, which degrades the range resolution accordingly. In [6], an enhanced pulsed radar processor is presented based on a nested Barker code, in which a mismatched filter is designed to reduce the range sidelobes based on the least mean square algorithm. The main limitation of the designed processor is that the operating waveform is very sensitive to Doppler frequency, which arises range sidelobes in existence of moving targets, that in turn increases the false alarm rate.

Researchers in [7] have discovered new bi-phase codes of deterministic lengths from 71 to 105 samples, these codes are known to almost have the best known ACF peak sidelobe levels compared to other binary codes of the same lengths. Codes with the optimal peak sidelobe level (OPSL) can be used as the radar operating waveform instead of compound Barker codes, because of their performance is clearly better than Barker signals specially in the existence of moving targets. Therefore, they can be used in radar systems which are intended for detection of small radar cross-section (RCS) targets as drones, and small unmanned vehicles.

In this paper, we are going to show the detailed design and the FPGA implementation of a pulsed radar signal generator and processor based on OPSL code for detecting small RCS targets. This paper has been divided into three parts. The first part deals with the system design methodology, the second part represents the FPGA implementation results, and the third part includes the paper conclusion and the future work.

2. The methodological approach

The design block diagram is shown in figure 1. The system is designed using the ISE design suite from Xilinx, based on the single precision floating point (FP) format with a word length of 32-bit, which allows representation of values between integers and also can represent much greater range of numbers than the fixed point format [8]. The design includes the baseband digital signal generator, and the radar waveform processor. The operating waveform is an OPSL binary phase-coded sequence of length 105 sample, that was chosen because it has a peak sidelobe level (PSL) of -26.444 dB [7], which is more efficient than the 105-sample compound Barker code. For Further assessment, the ambiguity functions (AF) for the 105-sample nested Barker code, and the used 105-sample OPSL code are shown in figure 2. It is clear that the used OPSL code is less sensitive to Doppler frequency than the Barker code, which is considered as a remarkable advantage for the OPSL signal. The design also includes a moving target indicator (MTI) processor followed by a cell-averaging constant false alarm rate (CA-CFAR) to maintain an adaptive threshold. After that, the detected targets information are stored in a first input first output (FIFO) module, such that it can be visualized in a proper way by the data processing techniques. The system design parameters are listed in table 1.

2.1. The waveform generator and the system synchronizer

The signal generator contains the digital signal coefficients represented in the fixed point format, that are created by Matlab. These values are loaded on a read-only-memory (ROM), and addressed by the 15 MHz clock frequency generated by the clock generator (synchronizer). The ROM output flows directly to the digital-to-analog converter (DAC) to be an analog signal, and then it is directed to the transmitter radio frequency (RF) chain. The signal generator module also provides a 120 MHz clock signal beside the main system frequency, that is used for acquiring the received signal. The ROM address bus is generated from a 9-bit counter. In the radar receiver RF chain, the target received echo is amplified and down-converged to the baseband, after which it goes to the analog-to digital converter (ADC). The ADC acquires the signal by a sampling rate of 120 MHz. The resulted data stream is fed to a moving average

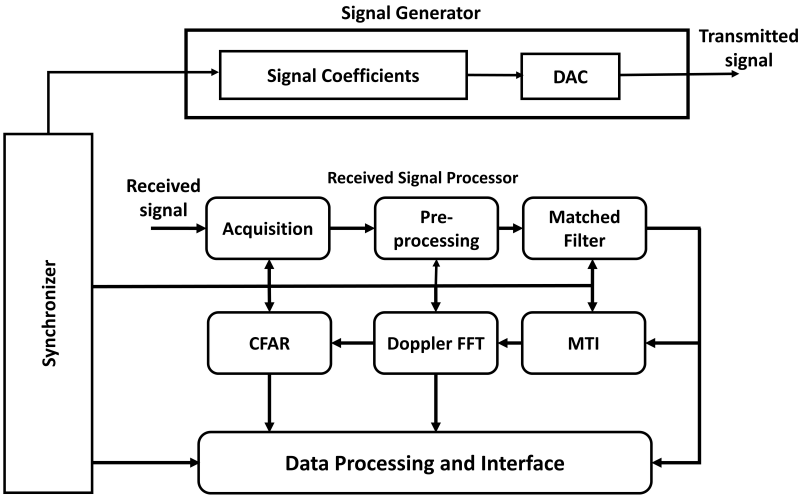

Figure 1. The block diagram of the radar signal generator and processor.

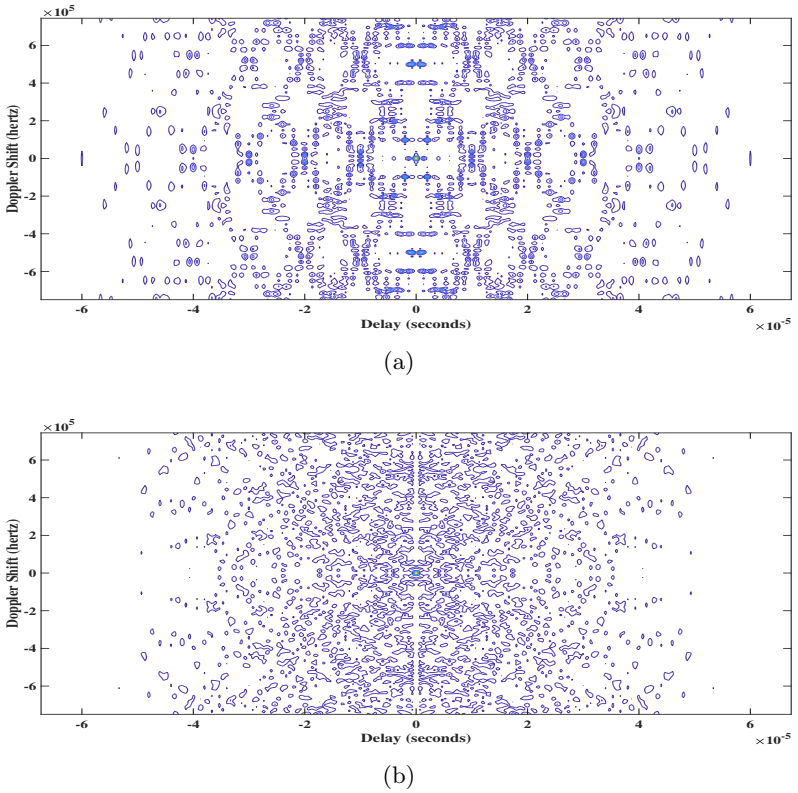

Figure 2. (a) AF of the 105-sample nested Barker code. (b) AF of the used 105-sample OPSL code.

filter for smoothing the received noisy echo, and then it is downsampled to the system clock frequency. After the downsampling process, the smoothed signal is converted to FP format for further processing. The block diagram incorporates the digital waveform generator and the signal acquisition method is shown in figure 3.

Table 1. The design parameters.

Parameter	Value
OPSL code sequence in hexa-decimal	1C6387FF5DA4FA325C895958DC5
Code length	105 *samples*
Waveform modulation type	Binary phase modulation
Range resolution ΔR	10 m
Maximum unambiguous range R_u	5.12 km
Pulse duration τ	7 µ sec
Subpulse duration δ	66.67 n sec
Sampling frequency f_S	15 MHz
Pulse repetition time PRT	34.13 µ sec
Pulse repetition frequency PRF	29.3 kHz
Coherent pulse interval CPI	256 *pulses*
Number of samples per subpulse	1 *sample*
Duty cycle d_t	0.205
Number of samples per PRT	512 *sample*

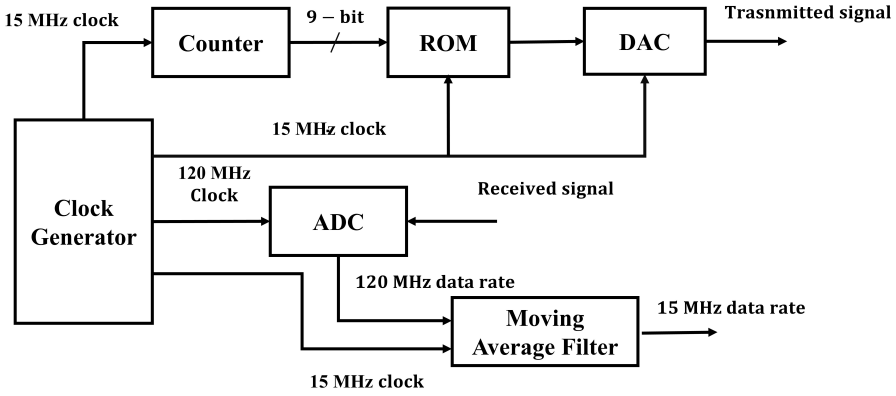

Figure 3. The block diagram of the signal generator and acquisition.

2.2. The matched filter

The matched filter (MF) is an algorithm used for pulse compression waveforms to maximize the signal-to-noise ratio (SNR). The MF is implemented in the frequency domain using the Fast Fourier Transform (FFT) algorithm, as the time domain implementation is not a practical process requiring huge computational resources [4], [6]. The operating OPSL binary code has MF sidelobes, that are removed using the optimum filter (OPF) technique demonstrated in [9]. This method had proven efficiency in suppressing the MF sidelobes of compound Barker codes. Here, it is used for the binary OPSL code of length 105 samples and also has the same efficient effect. The MF is accomplished by calculating the spectrum coefficients of the received signal, that they are multiplied with the stored conjugate spectrum coefficients of the replica signal and the sidelobe cancellation filter.

The optimum filter and the replica signal coefficients are created in frequency domain by Matlab in the FP format. Then, they are multiplied with each other and stored on a ROM addressed by a 9-bit counter synchronized with the received signal FFT module output. The multiplier output is fed to the inverse FFT module to provide the auto-correlation function

(ACF) that is free of sidelobes. Figure 4 illustrates the 105-sample OPSL binary code matched filter implementation with the sidelobe cancellation technique.

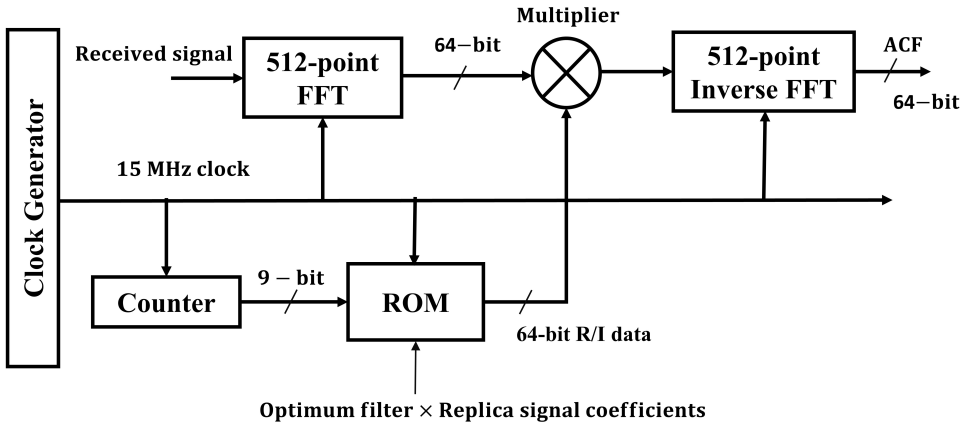

Figure 4. The matched filter implementation with the sidelobe cancellation technique.

2.3. Doppler processing

To specify the target Doppler value and consequently its speed, the real and imaginary (R/I) fast-time data streams coming from inverse FFT module of the MF need to be ordered by a buffer circuit to formulate the slow-time data. The data buffer network is composed of two dual-port random access memory (RAM) modules, and a FFT module. The first port of the first RAM receives the data in its natural order, while the second port is in the read mode. The read operation is not done by the data natural order, but the read address is hopping by a step of 512 samples, such that the output data is the slow-time data.

This operation lasts for a time equals the coherent pulse interval (CPI), time calculation is accomplished by means of a 18-bit counter ($512 \times 256 \times 2$). When the CPI is finished, the operation is reversed in a way that the first port is in the read mode, and the second one in the write mode. The whole addressing and control operation is managed by a circuit named buffer addressing.

The obtained slow time data flows to the MTI circuit for eliminating zero-Doppler frequency targets. The MTI structure is simply composed of a single delay line and a subtractor, its frequency domain response is indicated in Eq. (1) [4].

$$H(z) = 1 - z^{-1} \tag{1}$$

The MTI FPGA implementation incorporates a single flip-flop (FF) of a 32-bit width, and a 32-bit FP subtractor (for each data bus R/I). The MTI R/I output is directed towards the FFT module. If a moving target is existed, the FFT output will be a peak at the Doppler cell representing the target Doppler frequency. Since the CFAR operates on the fast-time data, the FFT output has to be rearranged again in its natural order by another buffering circuit. The data is fed to the second dual-port RAM, in which it is written in the first port by its order for a CPI. In the read operation, the address hops by a size of 256 to formulate the fast-time data at the output port. Figure 5 shows the Doppler processing implementation block diagram.

2.4. CFAR design

Received signals from targets are always contaminated with numerous types of interferences which often have a variable level. Therefore, maintaining a fixed threshold will cause

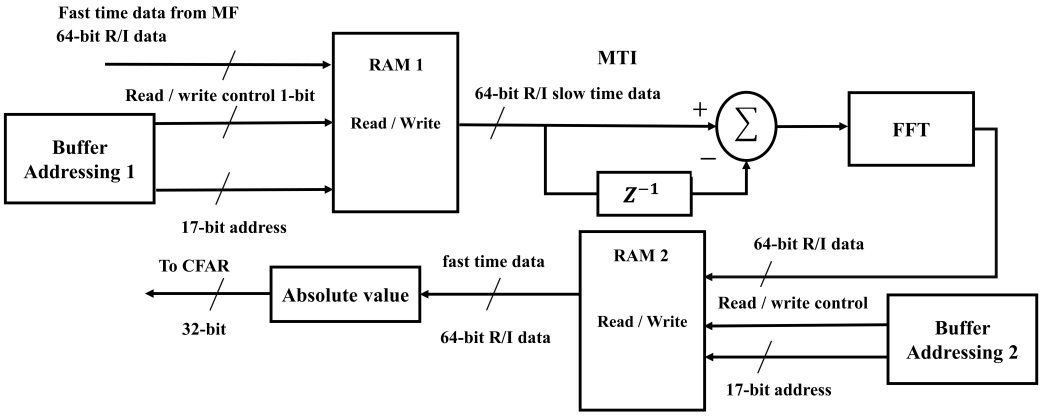

Figure 5. Doppler processing implementation block diagram.

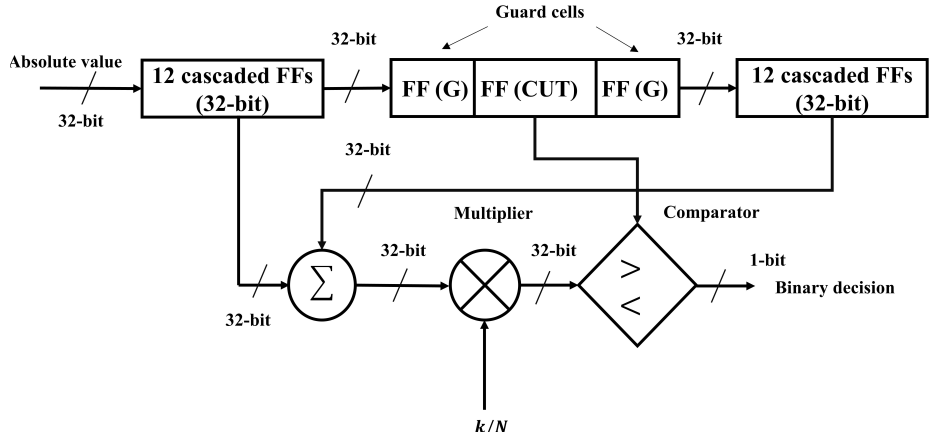

Figure 6. The CFAR implementation block diagram.

misdetection of weak targets because of the low SNR [2]. CFAR is an algorithm that provides an adaptive threshold according to a fixed probability of false alarm P_{fa}. In this design, P_{fa} is chosen to be 10^{-6} with a total number of CFAR cells are 24 cells: 12 on the right window, and the other 12 are on the left window. In the middle, there are 3 cells: the cell-under-test (CUT) in between two guard cells (GC). The CFAR threshold multiplier k is obtained by the following equation.

$$k = N(P_{fa}^{-1/N} - 1) \tag{2}$$

Where N is number of range cells in each CFAR window. The CFAR implementation operates on the absolute value of the Doppler processing data output. It consists of an adder tree for each window to collect the window sum. The two windows sums are then collected and multiplied by k/N using a FP multiplier, then the output is compared to the CUT to make a decision if there is a target or not. Every cell in CFAR is modeled by a FF of 32-bit width, the comparator is implemented by a VHDL code. As seen in figure 6 , the CFAR implementation block diagram is illustrated. The CFAR decision is fed to the FIFO module, from which targets information are sent to the data processing serially. Range and Doppler frequency information of targets are calculated by counting circuits depending on the indexes of the FFT modules in the design.

3. Implementation results and discussion

The pulsed radar signal generation and processor are designed on the ISE 14.7 design suite from Xilinx on a XC7A200TFBG484 FPGA board. The simulation results are investigated using the Modelsim simulator, and the FPGA implementation are visualized on the chipscope. Chipscope is a tool on the ISE suite allows designers to visualize data flowing in the FPGA board, to ensure design accuracy. In figure 7 , the OPSL code is visualized using the chipscope tool. The signal is delayed by a number of samples to simulate a single target range.

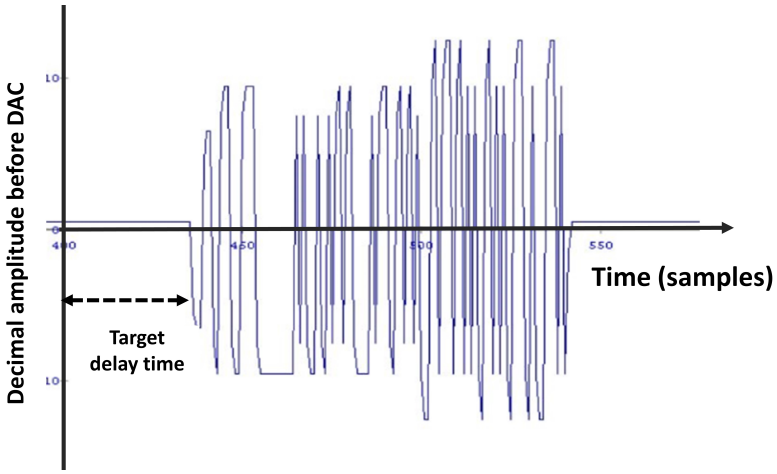

Figure 7. The transmitted OPSL binary phase-coded signal.

The Matlab simulation results of applying the optimum filter technique on OPSL binary codes are shown in figure 8 reflecting its superiority in removing the sidelobes in the matched filter output. The figure also illustrates the difference between the 105-sample OPSL code, and the 105-sample compound Barker code. The OPSL code sidelobe level is much lower than the Barker sequence.

The MF output Modelsim simulation and implementation results are shown in figure 9, and in figure 10, respectively. Figure 9 shows the successive matched filter output for a series of pulses returned from a single target. We can notice the ACFs peaks are equal, because they are returned from a fixed target. Also, it is clear that range sidelobes are completely suppressed. In figure 10, the successive matched filter output for a single target is displayed, we can deduce that the target is moving as the ACFs peaks are not equal due to Doppler modulation, and the range sidelobes are removed by the applied OPF technique.

As explained earlier, the input slow-time data to the MTI is formed by the buffering circuits. Reordering the fast-time data by the first RAM will concatenate the MF outputs in every range cell over the successive 256 pulses (Doppler cells). If the target is stationary, the slow-time data will be a rectangular pulse as shown in figure 11. This kind of targets will be eliminated by the MTI filter.

Also, the Doppler processing module output is illustrated in figure 12. The Doppler processing finds the input slow-time data spectrum. The FFT output is almost a delta-Dirac function located at a certain Doppler bin representing the target Doppler frequency from which the target speed is calculated.

In figure 13, we illustrate the CA-CFAR threshold versus the target absolute value coming from the Doppler processing module. It is clear that the threshold is adaptive based on the interference level and the designed P_{fa}. The CFAR produces a decision every clock. When the target value is greater than the threshold value, CFAR decision is 1, and the opposite is true.

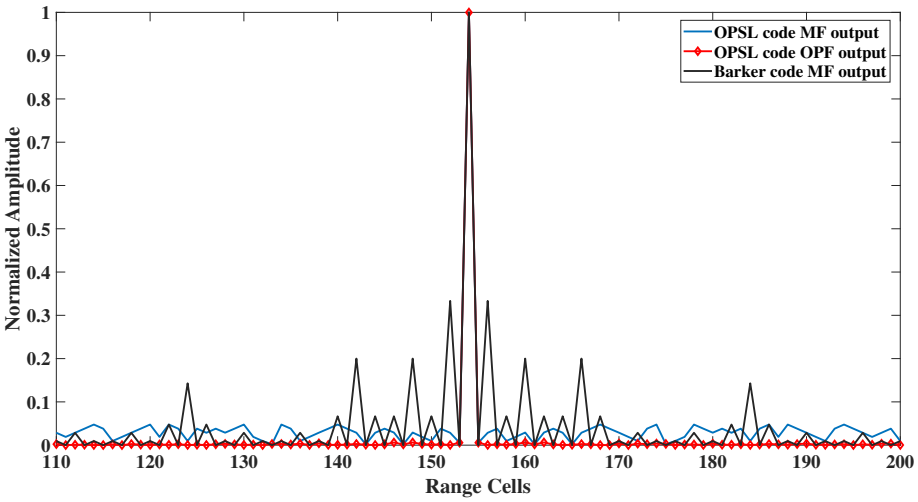

Figure 8. The blue line shows the traditional MF output, the red line with diamond marker indicates the OPF output free of sidelobes, an
sample compound Barker code.

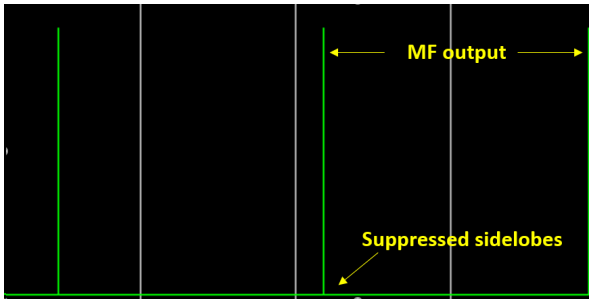

Figure 9. Successive MF output with suppressed sidelobes on the Modelsim simulator for a fixed target.

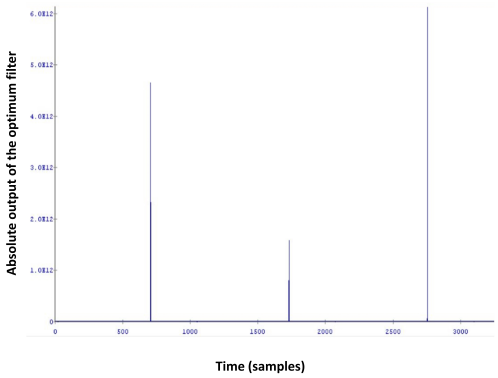

Figure 10. Successive MF output with suppressed sidelobes on the ISE chipscope for a moving target.

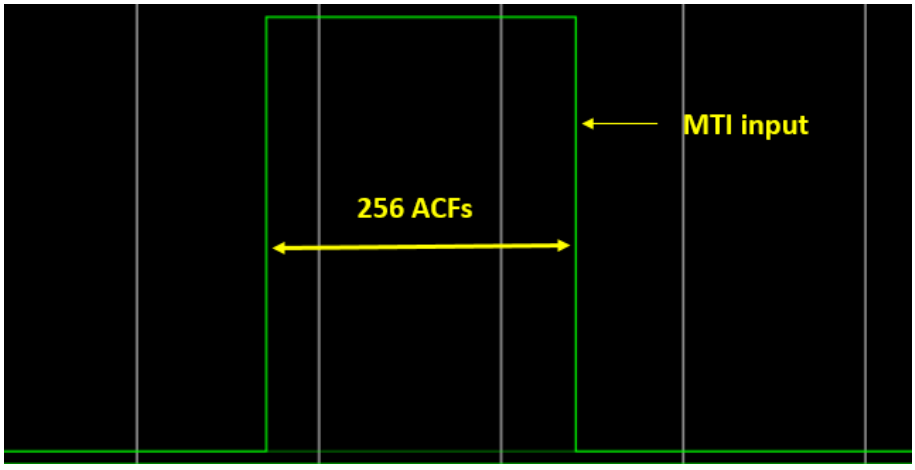

Figure 11. The MTI input for a fixed target.

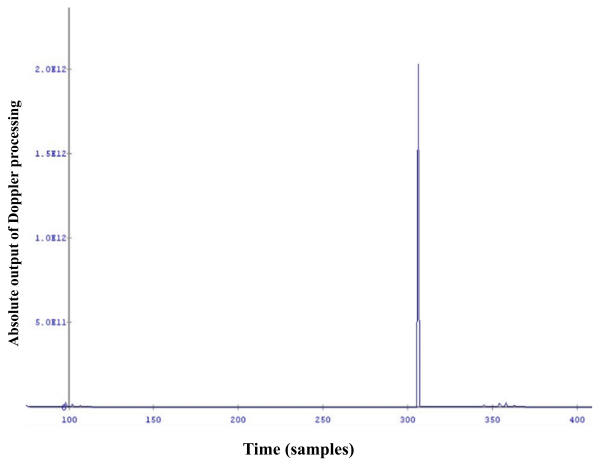

Figure 12. The Doppler processing absolute output for a moving target.

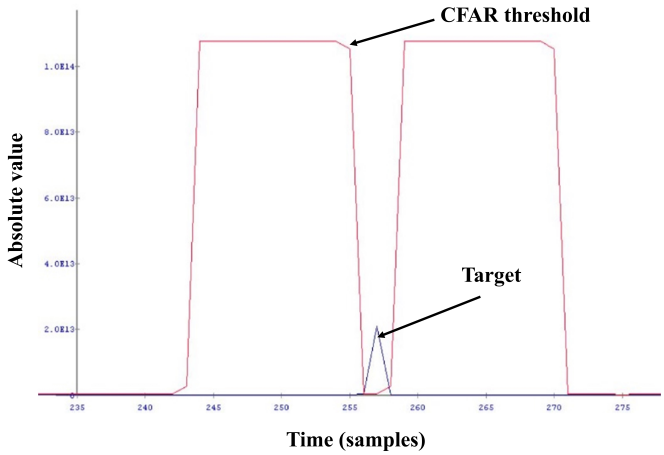

Figure 13. CFAR threshold is compared to the absolute value of the Doppler processing output.

In table 2, we are showing the used FPGA device resources utilization, the presented data are collected from the synthesis report.

Table 2. FPGA device resources utilization.

Resources	Maximum capacity	Utilized
Slice LUTs	134600	42146 (31%)
Slice Registers	269200	49456 (18.4%)
DSP Slices	740	92 (21%)
Block RAMs	13 Mb	4.2 (32%)

4. Conclusion and future work

FPGA had become the backbone of implementing real time applications as radar systems, because of its parallel behavior in executing commands. In this paper, we have demonstrated the detailed FPGA implementation of a pulsed radar signal generator and processor, based on a binary phase coded signal that is known to have the best PSL of -26.444 dB, compared to the 105-sample nested Barker code that has a PSL of -9.54 dB. In addition, we have applied a sidelobe cancellation technique on the matched filter output to enhance the detection process. The FPGA implementation results are shown to ensure the accuracy of the proposed system design. The future work will include evaluating the processor performance under noise and clutter conditions. Also, we will focus on developing a new technique to improve the SNR while keeping the implementation resources unchanged.

5. References

[1] Ochodnicky, J. et al. (2017) 'Drone detection by ku-band Battlefield Radar', *2017 International Conference on Military Technologies (ICMT)*. doi:10.1109/miltechs.2017.7988830.

[2] Skolnik, M.I. (1980) *Introduction to radar systems.* New York: McGraw-Hill.

[3] Youssef, A. et al. (2017) 'On time compression overlap-add technique in linear frequency modulation pulse compression radar systems: Design and Performance Evaluation', *IEEE Access*, 5, pp. 27525–27537. doi:10.1109/access.2017.2771799.

[4] Richards, M.A. and Melvin, W.L. (2022) *Principles of modern radar*. Stevenage: SciTech Publishing.

[5] Khalid, F.B. et al. (2012) 'FPGA based real-time signal processor for pulse Doppler Radar', *2012 International Conference on Informatics, Electronics and amp;amp; Vision (ICIEV)*. doi:10.1109/iciev.2012.6317426.

[6] Abdel Razek, A.M. et al. (2019) 'Design and implementation of an advanced radar signal processor with Waveform Generator and BIST unit on a single FPGA chip', *IOP Conference Series: Materials Science and Engineering*, 610(1), p. 012095. doi:10.1088/1757-899x/610/1/012095.

[7] Nunn, C.J. and Coxson, G.E. (2008) 'Best-known autocorrelation peak sidelobe levels for binary codes of length 71 to 105', *IEEE Transactions on Aerospace and Electronic Systems*, 44(1), pp. 392–395. doi:10.1109/taes.2008.4517015.

[8] 'IEEE standard for floating-point arithmetic' (2019) *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84. doi:10.1109/ieeestd.2019.8766229.

[9] Metwally, I.M. et al. (2019) 'A universal formula for side-lobes removal optimum filter in multi-samples phase coded pulse compression radars', *Advances in Intelligent Systems and Computing*, pp. 568–578. doi:10.1007/978-3-030-31129-2-52.