

PAPER • OPEN ACCESS

Design and Implementation of an Advanced Radar Signal Processor with Waveform Generator and BIST Unit on a Single FPGA Chip

To cite this article: Ahmed M. Abdel Razek *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **610** 012095

View the [article online](#) for updates and enhancements.

A promotional banner for the 240th ECS Meeting. The banner features a colorful striped border at the top. On the left, the ECS logo is displayed in a green circle. To the right of the logo, the text reads: "240th ECS Meeting", "Digital Meeting, Oct 10-14, 2021", "We are going fully digital!", "Attendees register for free!", and "REGISTER NOW" in bold orange letters. On the right side of the banner, there is a photograph of a diverse group of people in a professional setting, with a man in a white shirt and tie clapping and smiling.

ECS **240th ECS Meeting**
Digital Meeting, Oct 10-14, 2021
We are going fully digital!
Attendees register for free!
REGISTER NOW

Design and Implementation of an Advanced Radar Signal Processor with Waveform Generator and BIST Unit on a Single FPGA Chip

Ahmed M. Abdel Razek , Fathy M. Ahmed, Hazem Kamel , Wael Mahfouz, and K. H. Moustafa

Radar Department, Military Technical College, Cairo, Egypt.
fkader@mtc.edu.eg.

Abstract. Field Programmable Gate Arrays (FPGAs) are best suited for signal processing applications that require real-time processing. Therefore, it is preferred over DSP processors in implementing radar receivers that process incoming continuous stream of data. In the past, implementing complex arithmetic operations in floating point representation was a monopoly on DSP processors and the designers had to work around the sequential nature of the DSP processor to make it suitable for real-time applications by using buffered and multi-clocking designs. But nowadays, the great advances in FPGA design technology minimized this design effort since it is capable of performing these complex algorithms in real time. This paper represents the design and implementation of an advanced radar signal processor for binary phase-coded pulsed radar incorporating a time range side lobes suppression technique on a single FPGA chip. The proposed hardware design includes a waveform generator, an advanced signal processing unit, and a built-in self-test (BIST). Design aspects and hardware details for each part is introduced thoroughly.

Keywords: Radar; Matched Filter; Barker code; FPGA; VHDL; MTI; CFAR

1. Introduction

One of the most important needs that faces radar waveform designers is to design a waveform that is able to provide a relatively high average power (P_{av}) as well as a high resolution in range using pulse compression technique. The pulse compression techniques could be implemented either using frequency- or phase-modulation. In phase modulation, the main challenge is to design a phase-coded long pulse that provides high average power in transmission; meanwhile at reception, it could be compressed to provide an acceptable resolution in range. The more the code is long the more the average power is high; but unfortunately, the more the signal processing algorithm is complex which creates a conflict situation that contradicts with real-time signal processing. One of the solutions to deal with the complexity challenge is to use an advanced Field Programmable Gate Array (FPGA) to be able to increase the length of the modulating code, which results in increasing the radar resolution and signal-to-noise ratio (SNR).

Advanced FPGA families, such as the Xilinx Virtex, are characterized by a high-gate count and high-switching speeds in addition to suitable amount of DSP slices and relatively large internal block memory. These hardware features allow the implementation of complex algorithms and mathematical operations that were not possible to be implemented in real time.

This paper is organized as follows; after the introduction, design of the proposed radar signal processor is explained in section 2. Hardware implementation is considered in section 3. Finally, conclusion comes in section 4.



2. Design Methodology

The proposed design, as shown in Fig. 1, consists of a digital waveform generator in base-band and a radar DSP processor. The radar DSP processor consists of a digital coherent demodulator working at a sampling frequency 245.76 MHz. This stage is followed by down-sampling to achieve 125 KHz for Barker 13 and 1.7 MHz for compound Barker 169 sampling frequency. Two channels are provided with and without Moving Target Indicator (MTI) followed by a Constant False Alarm Rate (CFAR) processor as well as a binary integrator.

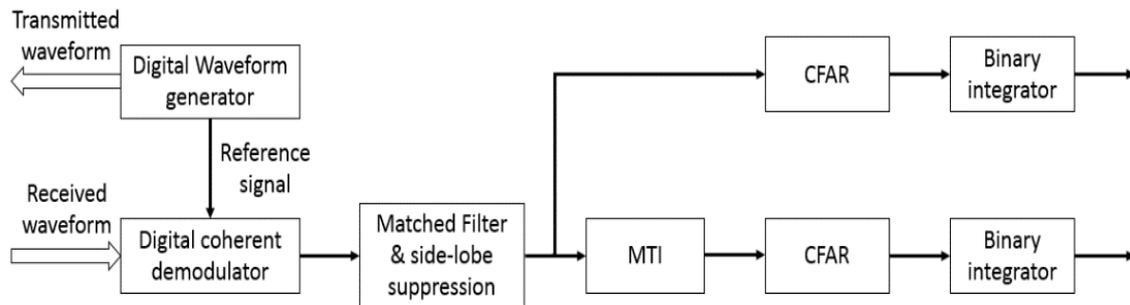


Fig. 1: Block diagram of the Proposed design

2.1 Digital Coherent Demodulator

The coherent demodulator, also called a Phase-Amplitude Demodulator (PAD), demodulates the IF signal to provide two orthogonal signal vector components, $\sin\theta$ and $\cos\theta$ [1]. The digital coherent demodulator consists of a digital multiplier that multiplies the received signal with reference 30MHz sine component for the in-phase (I) channel and 30MHz cosine component for the quadrature-phase (Q) channel. After multiplication, a low-pass filter is applied to remove the undesired high-frequency components and to keep the envelop only. The low-pass filter is designed using window method. A Hamming window is used with 21 coefficients resulting in 47dB stop-band attenuation [2]. Both I- and Q-signals for Barker 13 pulse and compound Barker are shown in Fig. 2 and 3 before low-pass filtering and after applying the designed low-pass filter. It could be noticed that the high-frequency component is completely removed and the Barker code 13 is left.

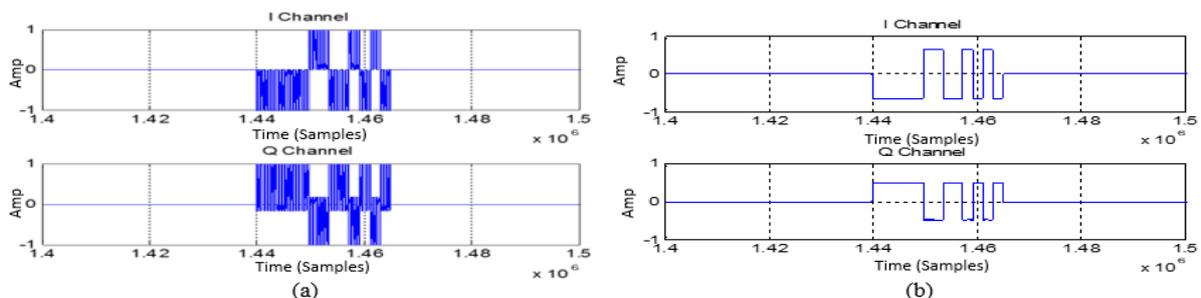


Fig. 2: Barker 13 I/Q signals (a) before low-pass filter and (b) after low-pass filter

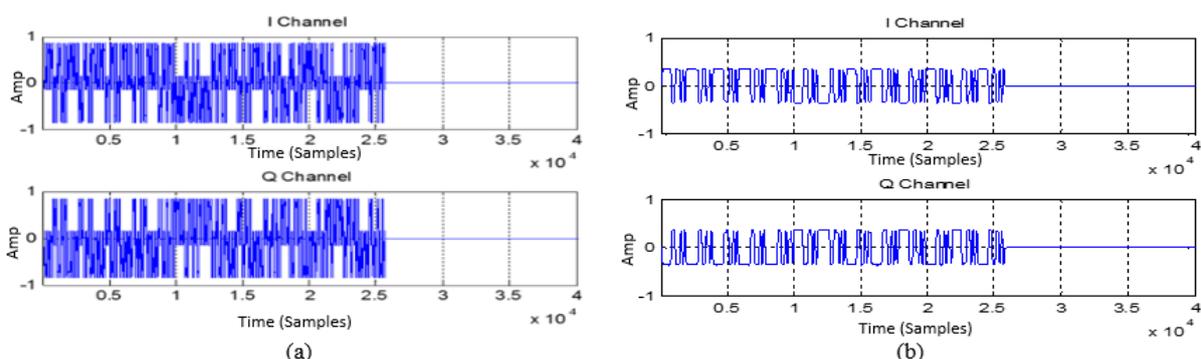


Fig. 3: Compound Barker code I/Q signals (a) before low-pass filter and (b) after low-pass filter

2.2 Matched Filter Design

The matched filter is a filter whose impulse response is a replica of the transmitted signal one. The matched filter can be designed using two methods: either the direct convolution method or the fast convolution method. The choice between both methods is done according to the length of the impulse response.

The direct convolution method means direct filtering that the time domain replica samples of the transmitted signal are used as coefficients in an FIR filter. Convolution in time domain is equivalent to multiplication in frequency domain and vice versa. When using long impulse responses (filter kernels), multiplication in frequency domain can be the more efficient of the two methods. This is done as follow:

$$FFT\{x(t) \otimes h(t)\} = X(f) \times H(f) \quad (1)$$

$$y = FFT^{-1}\{X(f) \times H(f)\} \quad (2)$$

The direct method requires N^2 real multiplications while the method of fast convolution, shown in Fig. 4, requires $12N \log_2 2N + 8N$ real multiplications. Consequently, the method of fast convolution offers an advantage of greater computational speed over the direct approach only if the number of values to be convolved is sufficiently large [2]. The simulation results for the matched filter output for both Barker 13 and compound Barker 169 are shown in Fig. 5.

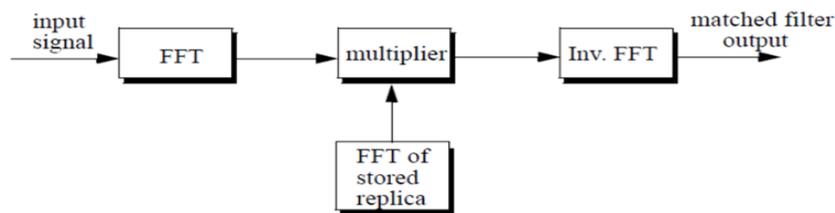


Fig. 4: Matched filter using fast convolution

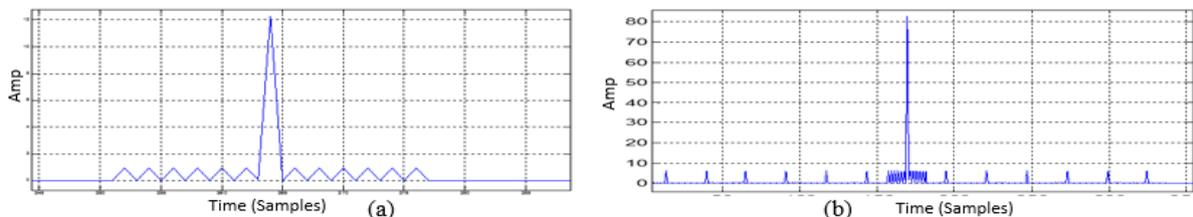


Fig. 5: Matched filter output for (a) Barker 13 and (b) compound Barker 169

2.3 Side-Lobe Suppression

The optimum filter is a filter whose impulse response is equal to the spectrum of the correlation function of the transmitted signal without time-side lobes divided by the spectrum of the correlation function of the transmitted signal [3]. The advantage of the optimum filter over other side-lobe reduction methods is that it does not reduce the amplitude of the time-side lobes; but it completely suppresses it. Another advantage over the mismatch filter is that it does not affect the amplitude of the input signal. Therefore, there is no bit growth in the word length at the filter output and no extra hardware resources are required after the optimum filter due to bit growth. The frequency response of the optimum filter for Barker 13 is given by for an N length sequence:

$$\omega(k) = \frac{\pi k}{N/2} 3 \quad (3)$$

$$H(k) = \frac{N}{N + 2 \sum_{i=1}^{(n-1)/2} \alpha(i) \cos 2i\omega(k)} \quad (4)$$

Both the frequency domain of the output of the matched and the frequency response for Barker 13 optimum filter are shown in Fig. 6. The result of the optimum filter is shown in Fig. 7. The optimum filter increases the main-to-side lobe ratio for Barker 13 from 22.3 dB to 369.1658 dB. In other words the side lobes are completely eliminated. For compound Barker, the optimum filter is given by:

$$H_1(k) = \frac{N}{N + 2 \sum_{i=1}^{(n-1)/2} \alpha(i) \cos(2i\omega(k))} \quad (5)$$

$$H_2(k) = \frac{N}{N + 2 \sum_{i=1}^{(n-1)/2} \alpha(i) \cos(26i\omega(k))} \quad (6)$$

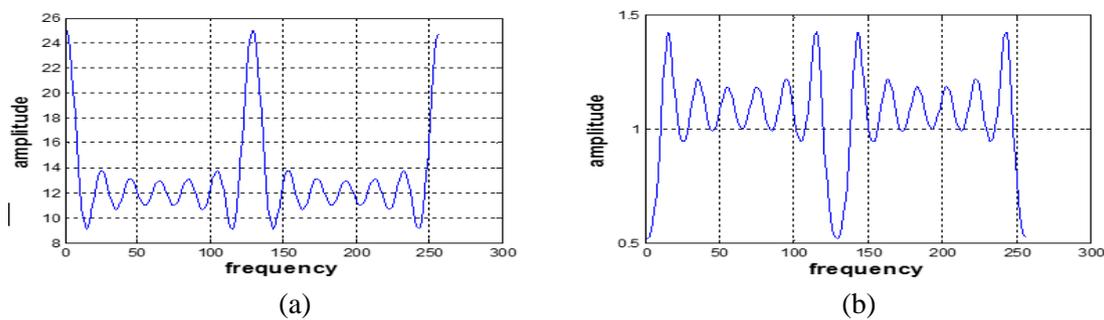


Fig. 6: (a) Frequency domain of Barker 13 correlation function and (b) Frequency response of Barker 13 optimum filter

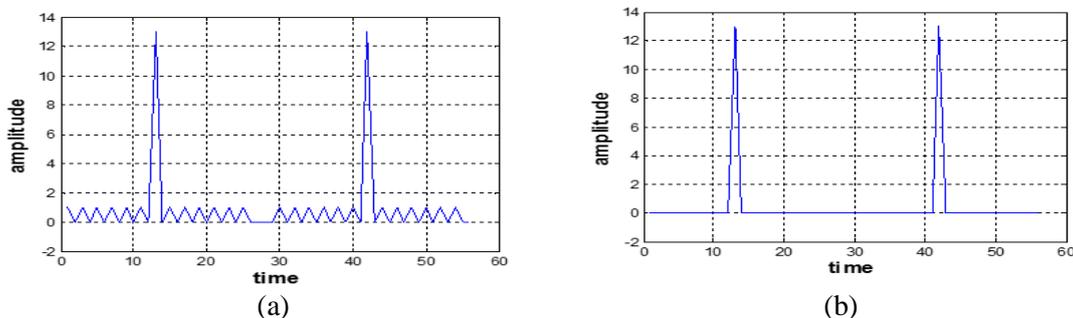


Fig. 7: Barker 13 correlation function (a) before optimum filter and (b) after optimum filter

Both the frequency domain of the output of the matched and the frequency response for compound Barker code optimum filter are shown in Fig. 8.

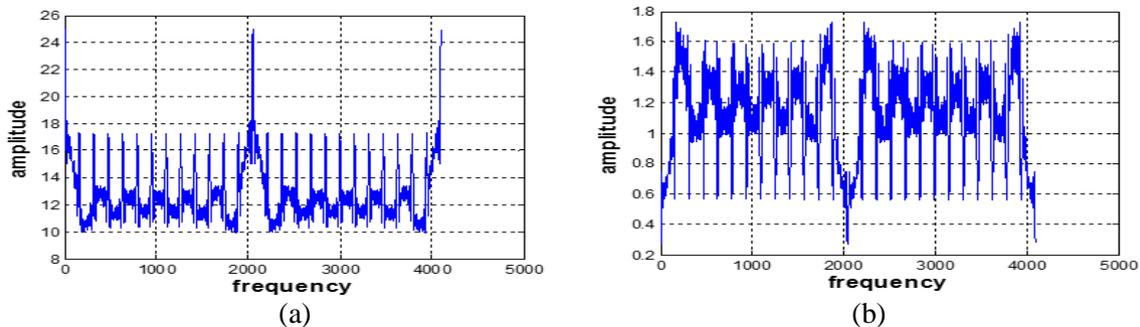


Fig. 8: (a) Frequency domain of compound Barker correlation function and (b) Frequency response of compound Barker optimum filter

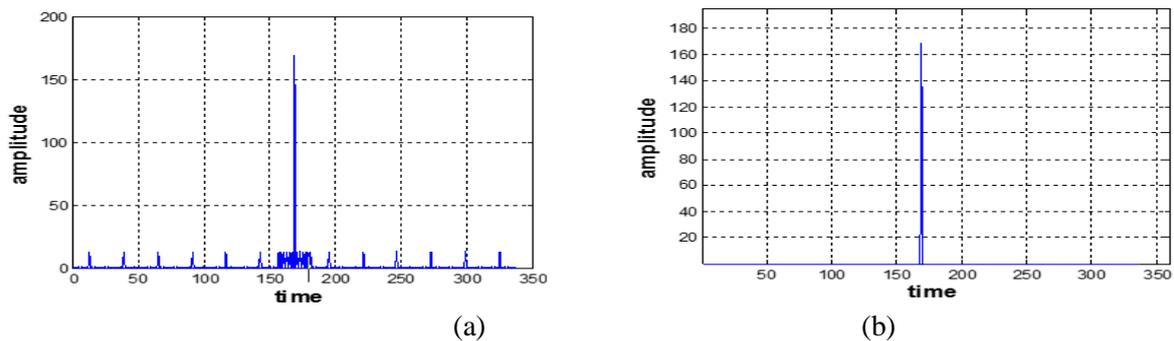


Fig. 9: Compound Barker correlation function (a) before optimum filter and (b) after optimum filter

2.4 Magnitude Calculation

The magnitude of both the I- and Q-signals must be calculated before sending the output of the MTI in the MTI channel or the output of the matched filter to the CFAR detection. The purpose of an MTI filter is to suppress target-like returns produced by clutter, and allow returns from moving targets to pass through with little or no degradation. The MTI used is a triple delay line canceller [4]. The magnitude calculation is done by summing the square of the I-signal and the square of the Q-signal, then calculating the square root of this sum.

During the design analysis, it was observed that the SNR appears to decrease when the square root is performed in the magnitude calculation. This decrease is logic since the amplitude of the signal before the CFAR detection is always greater than the amplitude of noise, and the ratio between two values increase when they are both squared. Fig. 9 shows the effect of calculating the square of the signal before applying it to CFAR detection when the SNR at the receiver input is 0 dB. The result shows clearly that the ratio between the amplitude of the echo signal and the amplitude of the noise is increased.

In order to study further the effect of calculating the square of the signal before CFAR detection, the Receiver Operation Characteristic (ROC) curve was derived at a probability of false alarm 10^{-6} for the following cases:

- Calculating the magnitude before CFAR detection.
- Calculating the square of the magnitude before CFAR detection.
- Calculating the 4th power of the magnitude before CFAR detection.

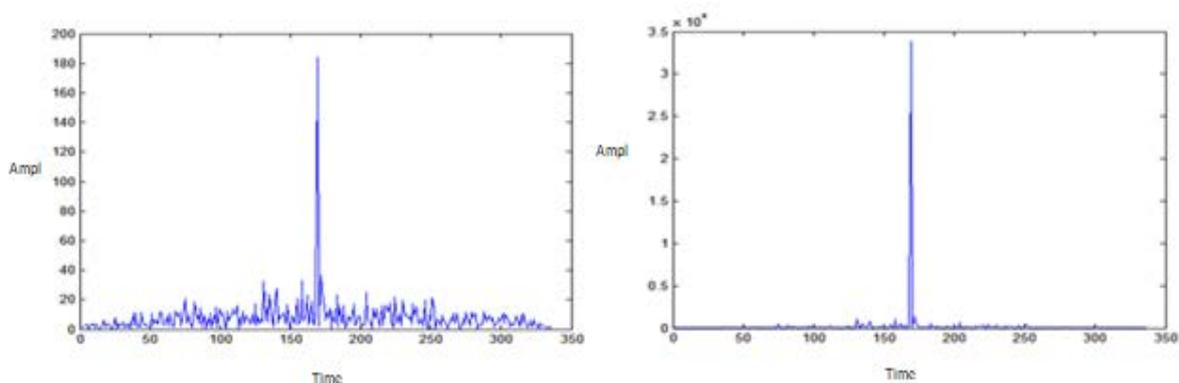


Fig. 10: Calculating the square of absolute signal: (a) Signal before square and (b) Signal after square signal

In Fig. 11, the ROC curve shows that taking the square of the magnitude improves the probability of detection; meanwhile, taking the 4th power of the magnitude gives nearly the same probability of detection obtained by using the square of the magnitude.

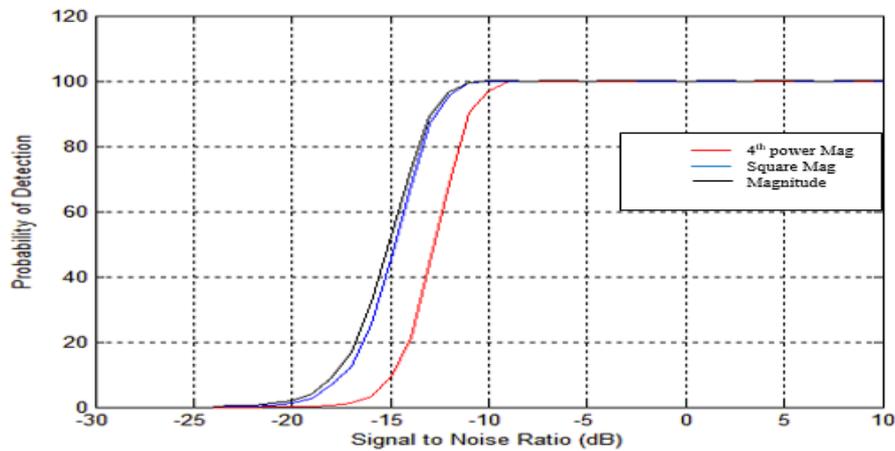


Fig. 11: ROC curve at the output of the CFAR

2.5 Constant False Alarm Rate (CFAR)

After the magnitude calculation for the output of both I and Q channels of the MTI, this magnitude value is fed to CFAR to take the decision. The implemented CFAR is the Greatest-of CRAF (GOCFAR) with 27 cells. The simplified block diagram of the implemented GOCFAR processor is shown in Fig. 12(a) [5]. Two guard cells around the test cell are introduced such that the summation of leading and lagging windows are not affected by the target returns in the test cell. The value of the required false alarm probability depends on the value of the multiplication factor (threshold factor), which is less than one. Fig.12(b) shows the threshold and decision of the CFAR for an echo signal of a moving target with $f_{Doppler} = 0.5 F_r$ and with noise level equal to 0 dB at the receiver input, where F_r is the pulse repetition frequency.

The used binary integrator in this design is an adaptive binary integrator based on CFAR output. The output of the CFAR is applied to a delay line chain. Each delay element possesses a delay time equal to the radar pulse repetition period (T_r) and the number of the required delay elements is equal to the CPI. The output of all delay lines are summed together in an adder tree. Then, the sum is compared with a threshold; if the sum exceeds this threshold, it is declared as a target. The decision taken when two targets are present at a noise level of 0dbm is shown in Fig. 13.

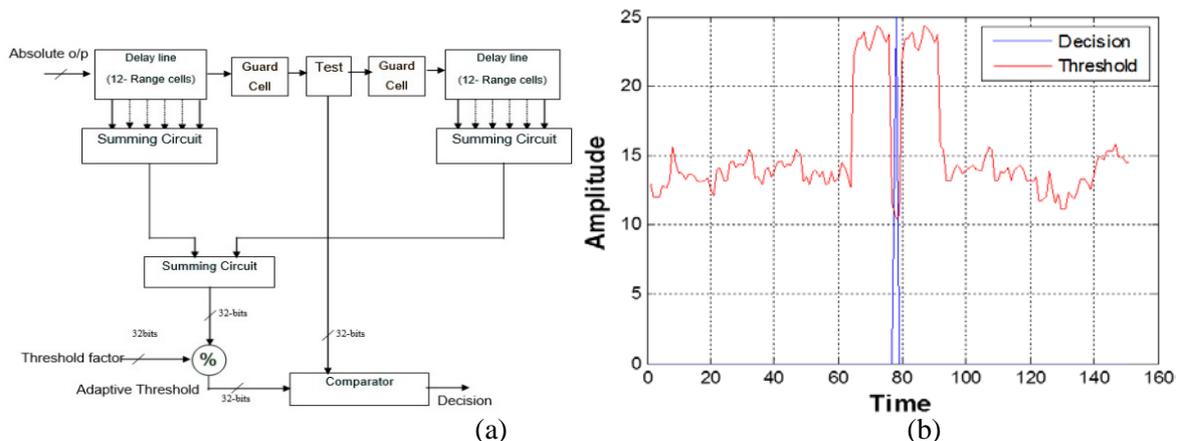


Fig. 12: Block diagram and the calculated threshold using a 27-cells GOCFAR

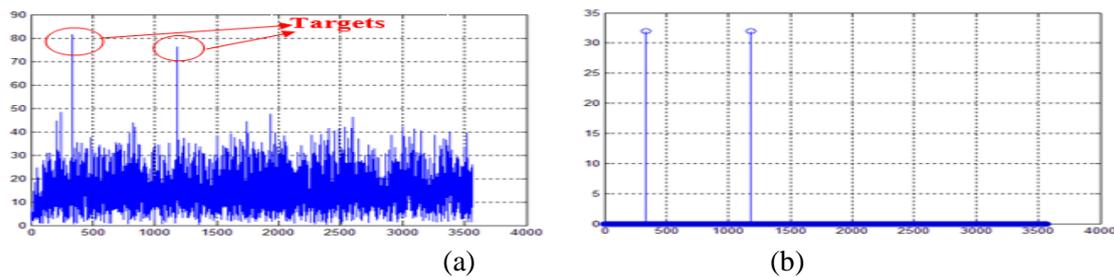


Fig. 13: (a) The output of the MTI shows 2 targets contaminated with noise and (b) The output of the binary integrator shows the detection of the two moving targets

3. Hardware Implementation

The hardware implementation is divided into two parts, the first part of the hardware design is generating the IF binary coded pulse, this pulse will be used for transmission and for testing with the ability of changing the phase of the generated analog signal to simulate a returning echo from moving target. The second part of the design is the radar signal processor which is capable of performing digital coherent demodulation, matched filtering, side-lobe cancellation, MTI filtering, CFAR detection and binary integration. The processor is designed for both Barker-13 modulated waveform and compound Barker-169. Arithmetic operations performed in the signal processor are done in single precision floating point data representation which is the same data representation used in the MATLAB design in order to have the hardware working with the same precision of the MATLAB simulations and to avoid quantization noise. The hardware used for implementing the design is the Xilinx ML605 DSP kit, this kit includes a development board with the Virtex-6 XC6VLX240T FPGA chip which has 241,152 logic cell equivalents, 768 DSP slices and about 216 Kbit RAM which makes it a suitable FPGA for implementing complex DSP algorithms, the FPGA board is equipped with an FMC daughter board that contains TI's ADS62P49/ADS4249 dual-channel 14-bit 250Mps ADC and TI's DAC3283 dual channel 16-bit 800Mps DAC on a daughter board.

3.1 Waveform Generator

The waveform generator is implemented in FPGA [6]. To generate any binary phase coded waveform for radar transmission in addition to generating an echo like signal. The echo testing signal is generated with a predefined amplitude, predefined Doppler shift amplitude and at a specific range cell. The waveform generator is also responsible for generating the radar sync pulse. The proposed digital circuit for generating the waveform generator consists of Direct Digital synthesizer, multiplexer, phase shifting circuit, comparator and binary counter Fig. 14(a) shows the block diagram of the complete system using the radar receiver and the waveform generator. The 16 bit output of the waveform generator is connected to the DAC3283 dual channel 16-bit 800Mps DAC available in the DSP150 data acquisition card. Fig.14(b) shows the results on oscilloscope for Nested Barker 169 waveform with a sub-pulse width of $0.6 \mu\text{s}$ and a total pulse-width of $104 \mu\text{s}$.

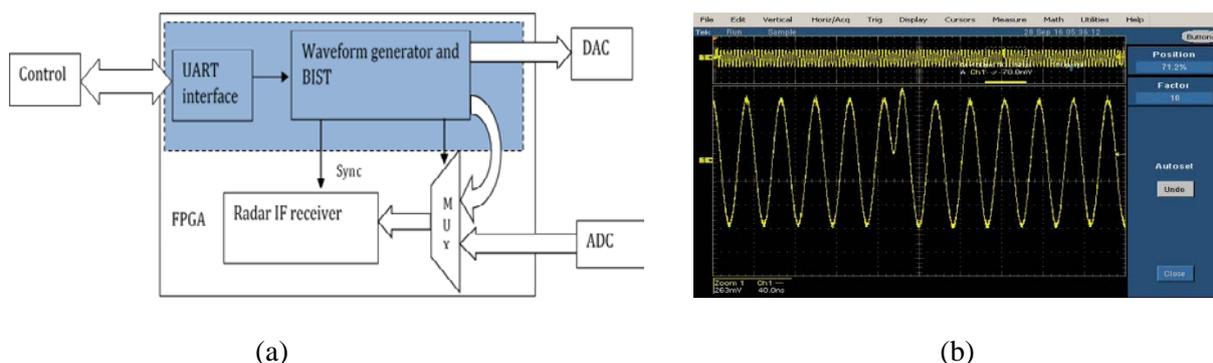


Fig. 14: (a) Waveform generator block diagram and (b) generated waveform on oscilloscope

3.2 Digital Coherent Demodulator

The input to the receiver is a 30MHz waveform sampled using the ADS62P49/ADS4249 dual-channel 14-bit ADC at a sampling frequency of 234.67MHz. A digital circuit is designed for demodulating the received signal based on the MATLAB calculations. The demodulator shown in Fig. 15(a) consists of a digital multiplier which multiplies the input signal with reference Sine for the in-phase channel and reference Cosine for the quadrature-phase channel, since the input word length is 14-bit, the output of the multiplier is 28-bit word length due to multiplication bit growth. The output of the multiplier is applied to a low pass filter for removing the high frequency components and leaving the envelope of the signal, the coefficients of the low pass filter are previously calculated using MATLAB, then the coefficient values are scaled to be used in the Filter IP core. The IP core input word length is 28-bit and the output word-length is 16-bit this is done using the truncation of least-significant bit option in the IP core as shown in Fig. 15(b).

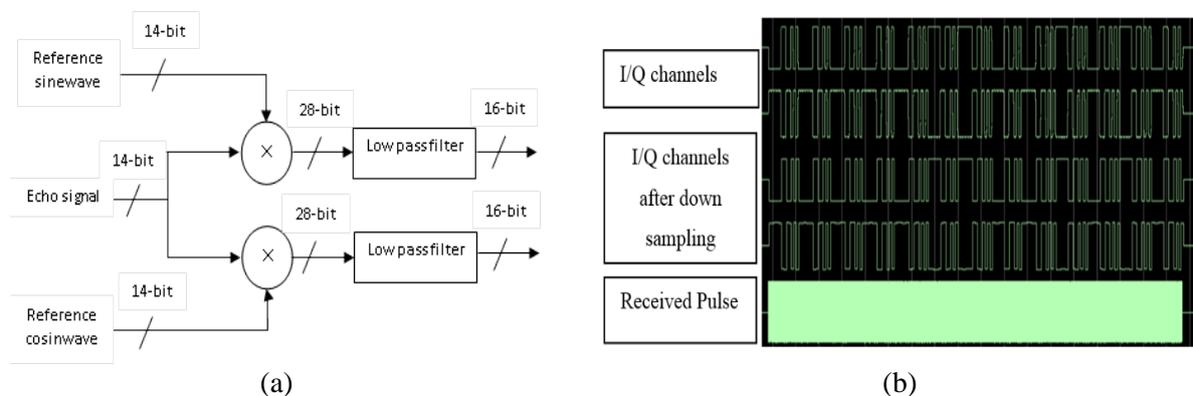


Fig. 15: (a) digital coherent-demodulator block diagram and (b) its simulation results

3.3 Matched Filter Implementation

The design is implemented for both Barker-13 and compound Barker-169. In case of Barker-13, shown in Fig. 16(a), the matched filter is simply an FIR filter with 13 coefficients representing the replica of the Barker code. This can be done in different ways including VHDL design, schematic design, and Xilinx logiccore IP FIR Compiler.

For compound Barker 169 shown in Fig. 16(b), since the filter coefficients are the replica of the 169 Nested Barker code, and its implementation in time domain requires about 17570 number of real multiplications, therefore it is not practical to implement it in time domain, and it is more convenient to be implemented using fast convolution (frequency domain filtering). The FFT of the replica for Baker 169 padded to the T_r length is calculated using MATLAB; this is multiplied by the FFT of the optimum filter coefficients. The resulting real and imaginary values are stored in a ROM whose address is a counter that counts from zero to T_r and it resets every start of T_r , the input data from the coherent demodulator is applied to FFT, then the output of the FFT is multiplied with the output of the ROM, the product is then send to IFFT unit to convert it back to time domain. Single precision floating point is used during the Matched filtering and optimum filtering processing to allow for high accuracy in calculations since no fixed point approximations takes place during FFT, IFFT calculations and coefficients multiplication.

3.4 MTI and Magnitude Calculator

The MTI, shown in Fig. 17(a) is designed using 2 block RAMs whose depth is equal to the number of range cells in each repetition period, the RAMs write enable is always set such that it stores the incoming data at each rising edge clock and output the old data of the previous period at the same edge, a counter is used to supply the RAMs with address, this counter is reset with the Radar sync pulse, the input of the first RAM represents the data off the third period and its output represents the

data of the second period, the output of the second RAM represents the data of the first period. The output of the first RAM is multiplied by negative 2 and added to the input of the first RAM and the output of the second RAM. The calculation the magnitude is done without using square root in order to increase the SNR before CFAR processing. Both I and Q outputs of the MTI filters are applied to multipliers which works as a squaring circuit to calculate I^2 and Q^2 both values are then send to an adder, as shown in Fig. 17(b).

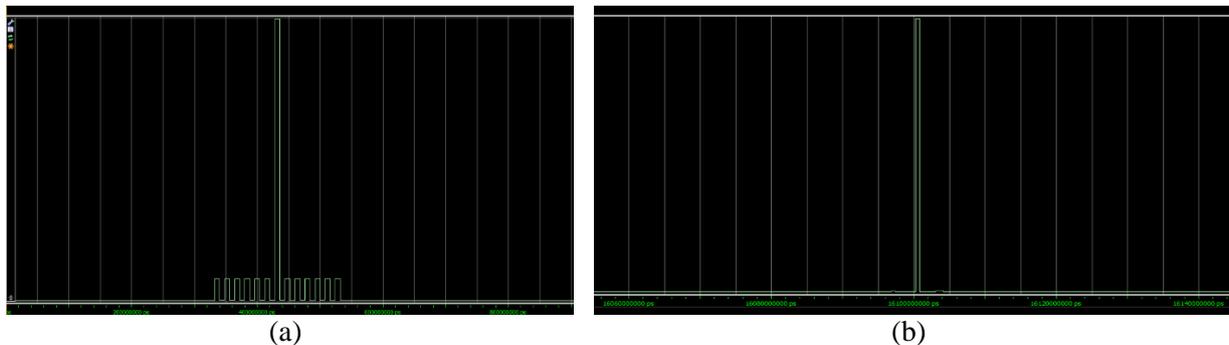


Fig. 16: (a) Barker-13 simulation results and (b) Nested Barker 169 with side-lobe canceller simulation results

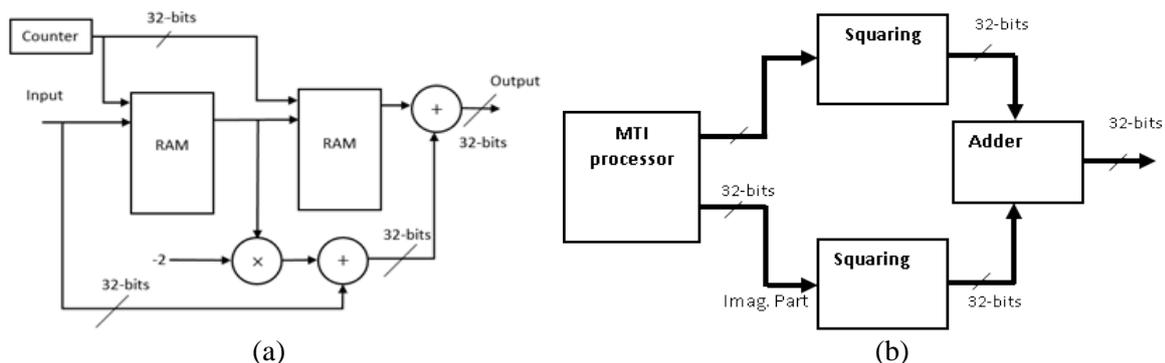


Fig. 17: (a) MTI and (b) magnitude calculator block diagrams

3.5 CFAR

The CFAR is implemented in single precession floating point, the addition is performed using recursive approach proposed in [7]. Fig. 18 shows the CFAR simulation results.

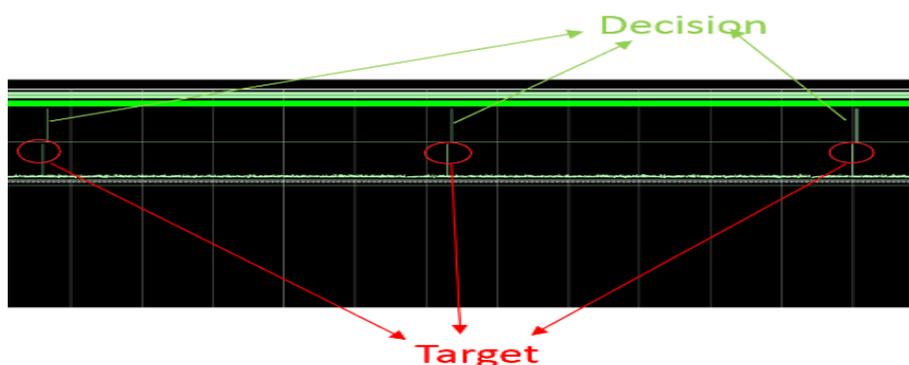


Fig. 18: CFAR simulation results

3.6 Total Device Utilization

The final design consists of the waveform generator, BIST unit and the signal processor, the design is implemented for both Barker 13 and Nested Barker 169 on the Xilinx Verix6 XC6VLX240T. Table 1 shows the device utilization for both Barker 13 and Nested Barker 169. The Nested Barker 169 processor consumes nearly twice the resources of the Barker 13 processor.

Table 1: The device utilization for both Barker 13 and Nested Barker 169

Required	Available	Barker 13	Nested Barker 169
Slice Registers	301,440	22,382(7%)	50,592(16%)
Slice LUTs	150,720	15,693(10%)	40,919(27%)
RAMB36E1/FIFO36E1s	416	26(6%)	53(12%)
RAMB18E1/FIFO18E1s	832	25(3%)	206(24%)
DSP48E1s	768	172(22%)	256(33%)

4. Conclusion

In the present paper, complete hardware design and FPGA implementation of an advanced pulse compression radar processor with waveform generator and built-in self-test have been proposed. Two approaches have been used in implementing the proposed hardware, the first approach is based on Barker 13 radar pulse, while the second one uses compound Barker 169. The optimum filter used for side-lobe suppression gives a great advantage in the practical hardware implementation of a processor for compound Barker 169. Calculation of the square of the magnitude before CFAR detection can increase the probability of detection of the CFAR.

5. References

- [1] Merrill I. Skolnik, Introduction to radar systems (3rd ed.), McGraw-Hill, (2001).
- [2] E. C. Ifeachor, Digital Signal Processing, A Practical Approach, Prentice Hall, 2002.
- [3] B. A. S. A. Abd El-Rahman H. Elbardawiny, C. Fathy M. Ahmed and D. Mamdouh Hassan, "A Novel Sidelobe Cancellation Method for Binary Barker Code Pulse Compression," 17th International Conference on AEROSPACE SCIENCES & AVIATION TECHNOLOGY, April 11 - 13 2017..
- [4] B. R. Mahafza, Matlab Simulations for Radar Systems Design, CHAPMAN & HALL/CRC, 2004.
- [5] Assem A. Moustafa, Fathy M. Ahmed, K. H. Moustafa, Halwagy y., "A New CFAR Processor Based on Guard Cells Information", IEEE International Radar Conference, Atlanta, USA, pp. 133-37, May, 2012.
- [6] A. M. A. Razik and F. M. Ahmed, "FPGA Based Waveform Generator and BIST unit for Binary Phase Coded Pulse Compression Radar."
- [7] F. M. A. Kader, Ali M. Abou Zeid, and Ali M. Abou Zeid, "Analysis and design of a modified GO-CFAR processor – Hardware implementation using PLDs," in Proceeding of the 2nd ICEENG Conference, M.T.C, Cairo, Egypt 1999