Shared memory based real-time implementation of UWB-OFDM SAR imaging system

MD ANOWAR HOSSAIN*, IBRAHIM ELSHAFIEY, MAJEED A. S. ALKANHAL
Department of Electrical Engineering, King Saud University, Riyadh, Kingdom of Saudi Arabia

A novel technique for FPGA implementation of potentially jamming-resistant and high-resolution synthetic aperture radar (SAR) system using UWB-OFDM architecture has been presented. The greater computation time and larger memory requirement is still the main difficulties in processing huge SAR raw data in real-time UWB SAR imaging. Significant part of the computation time is related to compression of raw data in range and cross-range domain which in turns heavily depends on computing FFT, IFFT and complex multiplication as part of SAR imaging. A shared memory based FPGA model is developed using Xilinx block-set which is able to perform range and azimuth compression with high accuracy. The design procedures are straightforward and can be applied to any practical SAR system for real-time imaging. The model is designed as hardware co-simulation using Xilinx system generator and implemented on Xilinx Virtex-6 ML605 FPGA.

(Received May 19, 2014; accepted July 10, 2014)

Keywords: Field Programmable Gate Araay (FPGA), Ultra-wideband (UWB), Orthogonal frequency division multiplexing (OFDM), Synthetic Aperture Radar (SAR)

1. Introduction

Synthetic aperture radar (SAR) is used to achieve high-resolution images of a large target area. SAR transmits signals at spaced intervals called pulse repetition intervals (PRI). The reflectionsat each PRI arestored and processed togetherto reconstruct a radar image[1]. In general, high-resolution SAR image is obtained using Ultra-wideband (UWB) waveforms as radar transmitted pulse. UWB pulses (500-MHz bandwidth and above) can enhance the range resolution considerably. UWB technology has dual advantages: good penetration capability and high-resolution target detection for radar applications [2].

Orthogonal frequency division multiplexing (OFDM) is a method of digital modulation shows a great potential to be used in forming radar waveforms. An OFDM signal is comprised of several orthogonal sub-carriers, which are passed over a single transmission path simultaneously. Each sub-carrier contains a small portion of the entire signal bandwidth [3]. Technology advances for increasing the sampling speed capabilities, allows accurate generation of UWB-OFDM waveforms. This results in a diverse signal that is capable of high resolution imaging. Although OFDM has been studied and implemented in the digital communication field, it has not been widely considered by the radar community rather than few efforts [4, 5].

Greater computation time of SAR imaging algorithms poses a limitation for real-time SAR imaging application. Advances in the operating speed of the field programmable gate arrays (FPGA) allow signal processing applications to be solved in commercially available hardware. Because of their highly parallel nature, a 10 to 20 times increase in SAR processing density is expected using modern FPGAs as compared to current multi-node RISC processors, such as the PowerPC. Historically, FPGAs were much harder to program than a processor. Recently, system generator for DSP from Xilinx® has made FPGA technology accessible to DSP engineers. Xilinx System Generator pioneered the idea of compiling an FPGA program from a high-level Simulink model based on Xilinx block-sets [6].

2. UWB OFDM signal generation

UWB-OFDM waveform can be generated by randomly populating the digital frequency domain vector as:

$$\Psi_{\omega} = \left[\Pi_{ns} \ \Pi_0 \ \Pi_{ps} \right] \tag{1}$$

where, Π_{ps} and Π_{ns} represent the positive and negative sub-carriers respectively whereas Π_0 represents the baseband DC value. IFFT is then applied to Ψ_{ω} to obtain the discrete time-domain OFDM signal as:

$$\Psi_{tx}(t) = F^{-1}\left[\Psi_{\omega} \right] \tag{2}$$

As an example, UWB-OFDM waveform can be formed using the following parameters: number of OFDM sub-carriers = 256, sampling time, Δt_s= 1ns results in baseband bandwidth, $B_0 = 1/2\Delta t_s$ = 500 MHz, dividing by a factor of two to satisfy Nyquist criterion. When Π_{ps} is populated with all 0's and modulation scheme is chosen as BPSK, UWB-OFDM waveform turn into a short spike as shown in Fig. 1 which can provide the best possible resolution in range domain and exact location of the target in radar applications.

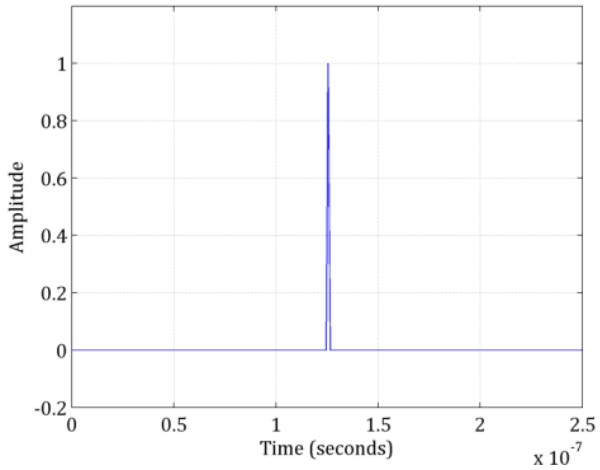

Fig. 1. UWB-OFDM signal using all sub-carriers.

However, the constant pulse shape causes the SAR system to suffer in jamming scenarios. We can use this signal for the SAR system in friendly environment to obtain high resolution images. When Π_{ps} is randomly populated and modulation scheme is chosen as BPSK, it is observed that the waveform is noise-like as shown in Fig. 2 and provides a unique signal at each PRI which is ideal for the SAR system in jamming scenarios i.e. hostile environment. UWB-OFDM SAR can be conformed to both friendly and hostile environments by just changing the transmitted signal.

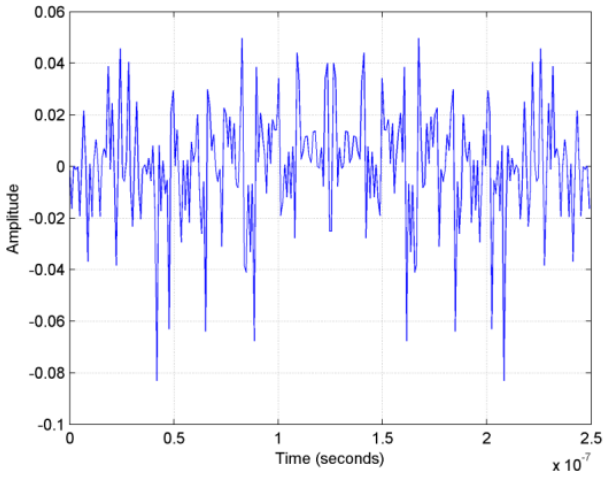

Fig. 2. UWB-OFDM signal using random sub-carriers.

3. Raw data generation

The raw SAR data is generated based on UWB-OFDM SAR simulation parameter shown in Table 1 using a gray-scale image of an 'aircraft' as a target profile. The input image is converted to a matrix. The positions of each element are assumed to be the range and the cross-range (azimuth) of the point target respectively while the normalized value of each element of the matrix is used as reflectivity of the target.

At each synthetic aperture position, a UWB-OFDM waveform is transmitted and the time-delayed signals reflected from the target are stored to form the SAR raw image space as shown in Fig. 3 containing the reflected signals for each synthetic aperture positions. The received radar signal is given as

$$\Psi_{rx}(t,u) = \sum_{n=1}^{N} \sigma_n \Psi(t - t_{dn}) + \eta(t) \tag{3}$$

where, N, σ_n and t_{dn} are the number of targets, reflectivity and round-trip delay associated with n^{th} target respectively. The term $\eta(t)$ denotes the AWGN. The round-trip delay t_{dn} is given by

$$t_{dn} = \frac{2}{c}\sqrt{(X_c + x_n)^2 + (y_n - u)^2} \tag{4}$$

where, X_c is the range distance to the center of the target area (swath) and (x_n, y_n) represents the location of the n^{th} target. The term u and c represents the synthetic aperture positions in azimuth direction and speed of light respectively.

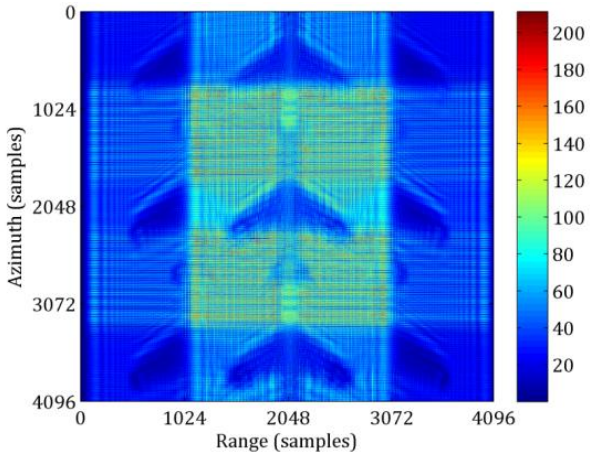

Fig. 3. Raw SAR image (Aircraft).

Table 1.UWB-OFDM SAR simulation parameters.

Parameter	Symbol	Value
Pulse Repetition Frequency	PRF	1024 Hz
Duration of flight	Dur	4 seconds
Velocity of the platform	V_P	200 m/s
Baseband bandwidth	B_0	500 MHz
Carrier frequency	f_c	7.5 GHz
Number of OFDM sub-carrier	N_{sub}	256
Range distance to center of swath	X_c	20000 m
Half target area width	X_0	200 m

4. Range and azimuth compression

Different imaging algorithms such as Range-Doppler algorithm and Omega-k algorithm is used to process raw SAR data to reconstruct the final SAR image [7]. Almost all imaging algorithm performs matched filtering separately in range and cross-range domains. Range compression, is performed by match filtering between each row of raw data and range reference signal as

$$\Psi_{Mx}(t,u) = F^{-1}\left[\Psi_{rx}(\omega,u)\Psi^{*}_{ref}(\omega,u)\right] \qquad (5)$$

The range reference signal $\Psi_{ref}(\omega,u)$ is an ideal echo signal from a unit reflector at the center of the current range swath and is given by

$$\Psi_{ref}(t,u) = \Psi(t - t_{d0}) \text{ where, } t_{d0} = \frac{2X_c}{c} \qquad (6)$$

Azimuth compression is obtained by match filtering in frequency domain between each column of the range compressed data and azimuth reference signal. IFFT is then applied to reconstruct the final SAR image. Azimuth reference signal is an ideal return from a unit reflector located at the center of radar-scanned area-i.e. $(x_n, y_n) = (X_c, 0)$ and is given as

$$\Psi_{azref}(\omega,u) = e^{-j\frac{2\omega}{c}\sqrt{X_c^2+u^2}} \qquad (7)$$

5. FPGA implementation

A 'shared memory' based FPGA model is developed to reduce the computation time for range/azimuth compression. The prominent blocks used in the developed FPGA model from Xilinx block-set are introduced in the following subsections.

5.1 System generator

System generator generates an FPGA bit-stream that is ready to run in FPGA hardware platform. It also creates a hardware co-simulation block to which the bit-stream is associated. In a simulation, the block provides the same results as co-simulated portion, but the results are computed in actual FPGA hardware which speed up the computation time dramatically.

5.2 Shared memory

The Xilinx shared memory block implements a random access memory (RAM) that can be shared among multiple designs or sections of a design. A shared memory block is uniquely identified by its name. Instances of shared memories of same name within the same model automatically share the same memory space. These interfaces make it possible for hardware-based shared memory resources to map transparently to common address spaces on a host PC.System generator's hardware co-simulation interfaces allow shared memory to be compiled and co-simulated in FPGA hardware. Shared memories facilitate high-speed data transfers between the host computer and FPGA, and bolster the tool's real-time hardware co-simulation capabilities. The access to the shared memory can either be Lockable or Unprotected.An unprotected memory has no restrictions concerning when a read or write can occur. In a locked shared memory, the block can only be written to when granted access to the memory. When the grant port outputs a 1, access is granted to the memory and the write or read request can proceed. The depth specifies the number of words in the memory. The word size is inferred from the bit width of the data input port.

5.3 Shared memory write / shared memory read

The Xilinx shared memory write block facilitates a high-speed interface for writing data into a Xilinx shared memory object. The shared memory write block is driven by the vector or matrix signal containing the data to be written into the shared memory object.

6. FPGA model

The schematic block diagram of the proposed FPGA model shown is Fig. 4 includes: Input buffer subsystem, FFT and IFFT block, complex multiplier and output buffer subsystem. The developed FPGA model based on Xilinx block-set and system generator is also presented in Fig. 5. The design is comprised of two subsystems that implements input and output buffer storage, named input buffer subsystem shown in Fig. 6 and output buffer subsystem in Fig. 7.

Each buffering subsystem consists of two shared memory blocks that provides the buffer storage for real and imaginary part of the raw data and output data respectively. Each shared memory is wrapped by logic circuitry (Counter, SR Flip-flops, delay and relational blocks) that controls the flow of data from the host computer, through the FPGA interface, and back to the host computer. This circuitry includes logic to enable high-speed transfers of the memory data when the FPGA acquires or releases lock of the shared memory. It takes advantage of the lockable shared memory mutual exclusion to implement a high speed I/O buffering interface for hardware co-simulation.

The Fast Fourier Transform block computes the FFT of the SAR raw data and the FFT of reference data is computed in MATLAB® for resource optimization as it is a row (column) matrix with fewer samples. The output of FFT block and the output of the shared memories of reference data are fed to the complex multiplier. Inverse Fast Fourier Transform block is then computes IFFT of the data found after complex multiplication.

Finally, the real and imaginary part of the output data is separately stored in two shared memories named shared memory (Real_out) and shared memory (Imag_out) inside the output buffer subsystem.

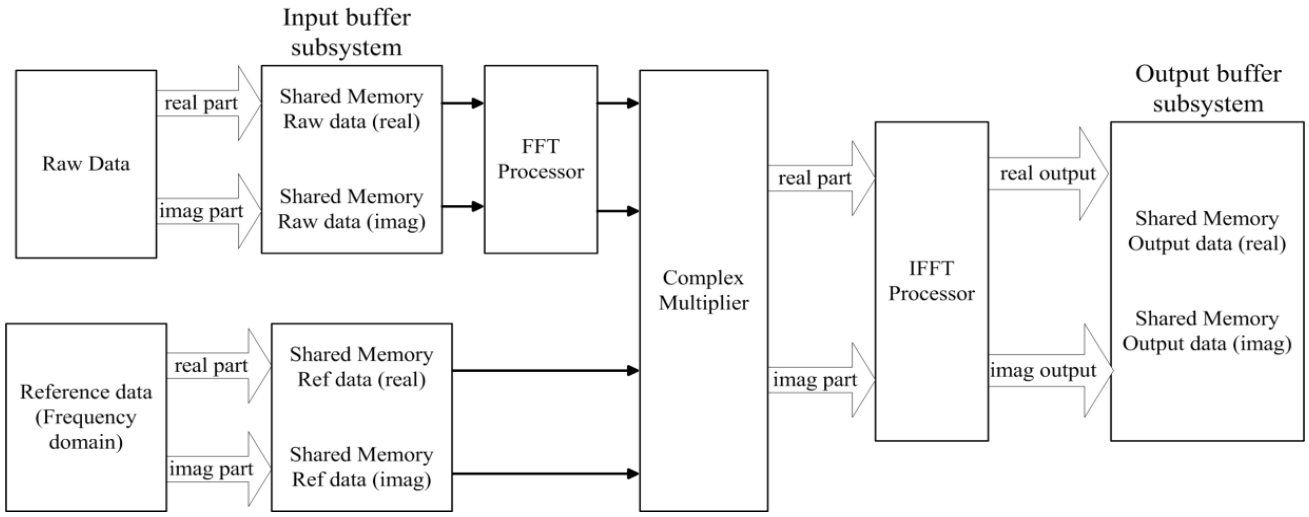

Fig. 4. Schematic block diagram of FPGA model.

7. Testbench model

The FPGA model shown in Fig. 5 is then compiled using system generator to generate hardware co-simulation block. A testbench model shown in Fig. 8 is also designed which includes the hardware co-simulation block. The test-bench model includes a 'From Workspace' block from Simulink to read the SAR raw data saved in a *.MAT file. The real and imaginary part of the data is separated and written to two lockable shared memories named shared memory write (Real_data) and shared memory write (Imag_data).Similar name is used for the shared memories inside the input buffer subsystem so that the data are shared to the FPGA hardware.

Finally, the real and imaginary part of the FPGA processed output data is collected by using two shared memory write blocks named shared memory write (Real_out) and shared memory write (Imag_out). To control the data processing flow shown in Fig. 9, the blocks provided in the testbench model is pre-configured with appropriate priorities for proper wake-up sequence such that: shared memory write (1), hardware co-simulation block (2) and shared memory read (3). The operation of the model is described as follows: (a) Shared memory write blocks wake up and request a lock of the input buffer lockable shared memories. Once lock is granted, the blocks write the data into lockable shared memories and release the lock. (b) Hardware co-simulation block then wakes up and the host computer shared memory data are transferred to the FPGA. The FPGA processes the input buffer data and writes the output into the output buffer shared memories. Finally, the FPGA releases lock, causing the FPGA shared memory data to be transferred back to the host computer.(c) The shared memory read blocks wake up and request a lock of the output buffer lockable shared memories. The blocks read data from output buffer and drive its output port with the processed output data.

The experimental test is then performed for the raw data generated by using an 'aircraft' as a target profile. Range compression is performed by feeding raw data to testbench model and running the model on Xilinx Virtex-6 ML605 FPGA board [8]. Fig. 10 shows the SAR image after range compression. Finally, azimuth compression is accomplished by using the same model to reconstruct the final SAR image shown in Fig. 11.

8. Results and discussions

The error level of FPGA output shown in Fig. 12in comparison with MATLAB® is computed by subtracting the FPGA output from MATLAB® output. We observe that very low error fluctuated between 0 and .0001 i.e. at 4^{th} decimal place which is negligible. Significant parts of the errors come from the truncation at the complex multiplier output. The reason behind the FPGA output with low error is that the raw data is converted into integer format before feeding to FPGA.

Table 3 summarizes the resource estimation; we observe that the lockable shared memory requires slightly more resources than unprotected shared memory because it requires some logic circuitry to handle mutual exclusions. But the lockable shared memory requires less computation time as tabulated in Table 2. So, we can summarize that lockable shared memory is the best choice for the proposed FPGA model.

Table 2.Computation time and frequency.

Shared Memory	Clock Period	Maximum Frequency
Lockable	2.979 ns	335.683 MHz
Unprotected	3.993 ns	250.438 MHz

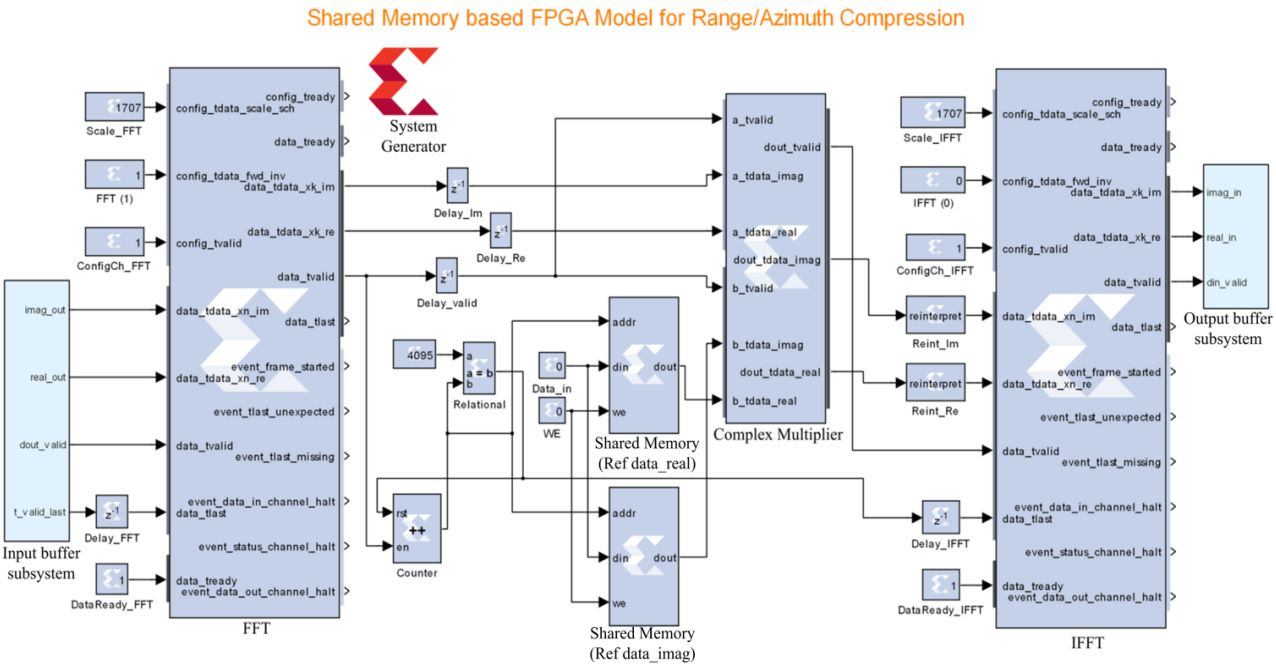

Fig. 5. FPGA model for range/azimuth compression.

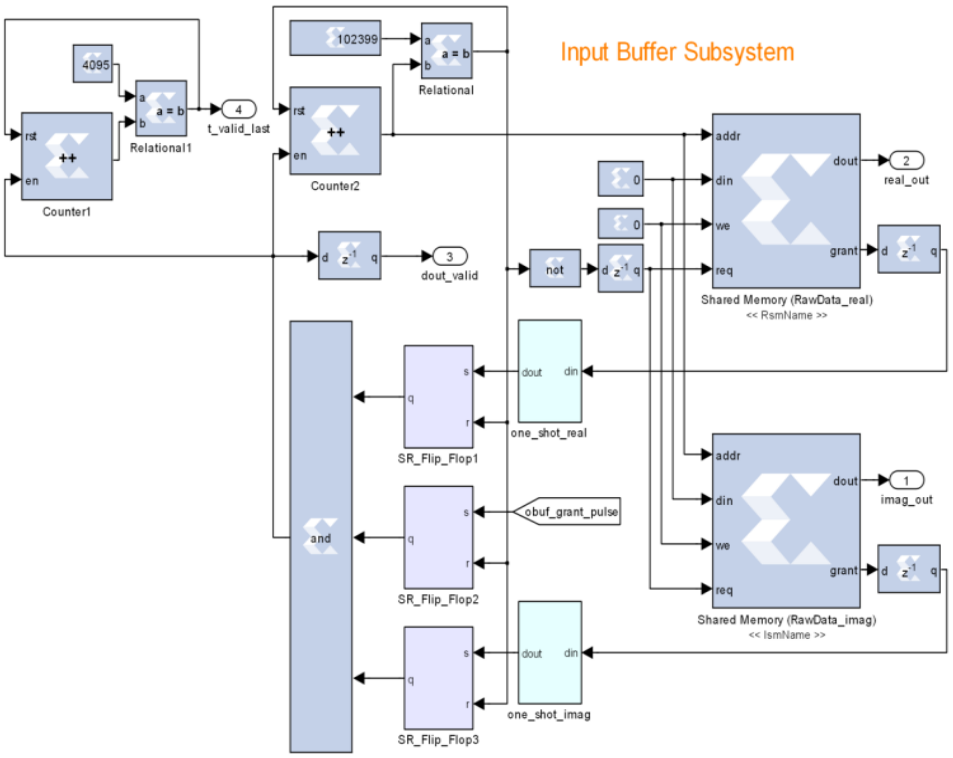

Fig. 6. Input buffer subsystem.

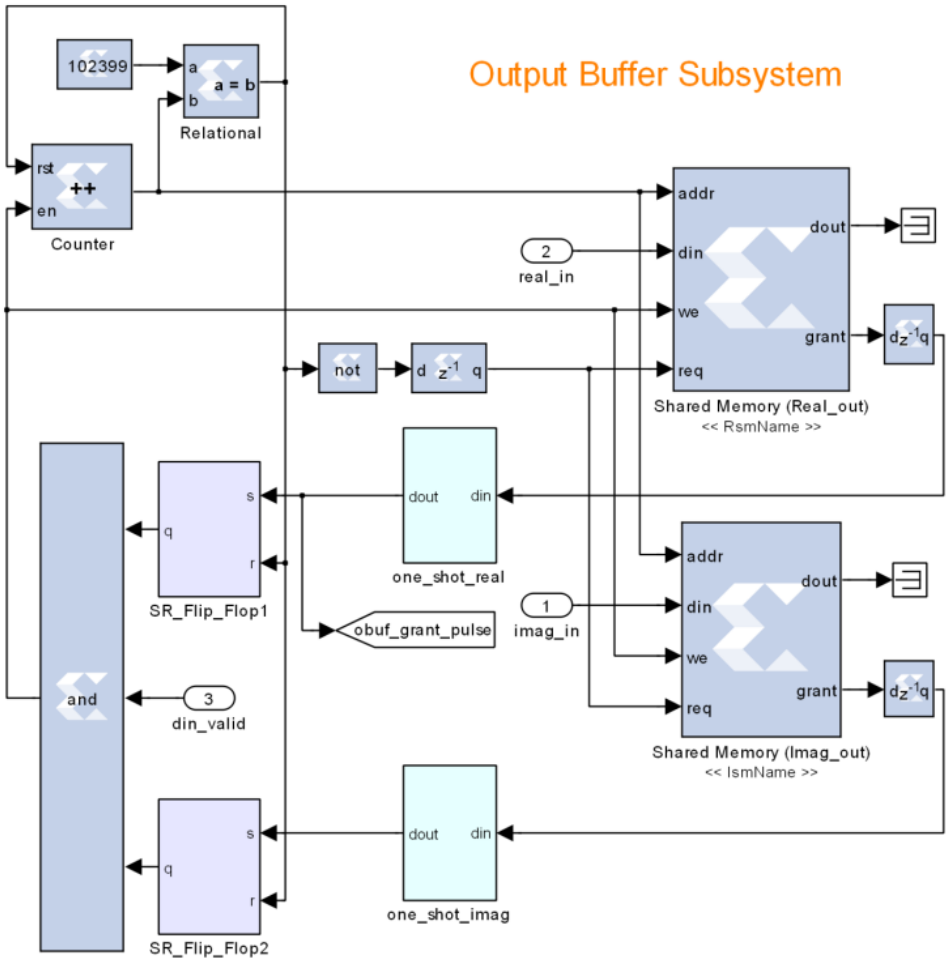

Fig. 7. Output buffer subsystem.

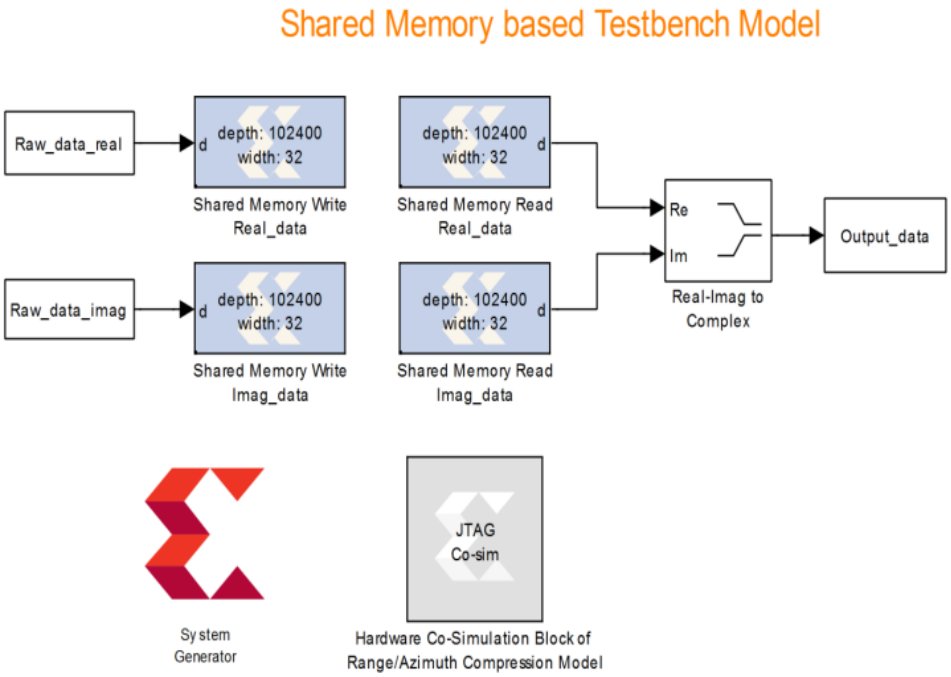

Fig. 8. Testbench model.

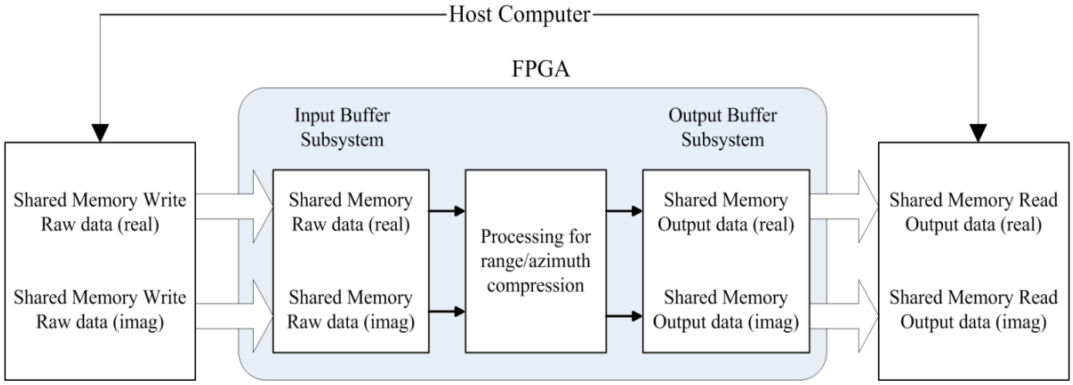

Fig. 9. Data processing flow.

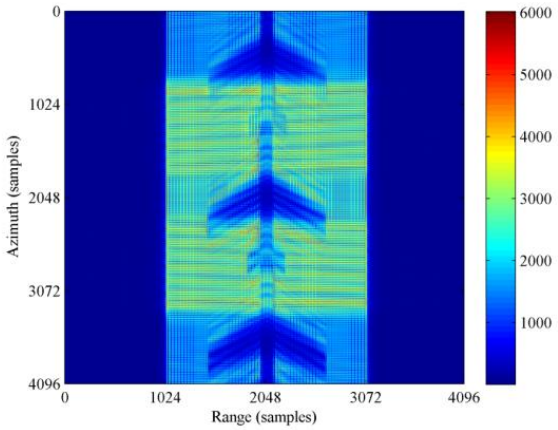

Fig. 10. SAR image after range compression.

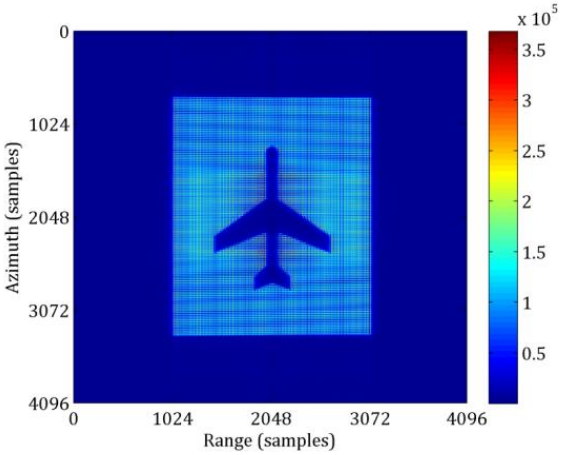

Fig. 11. Final SAR image after azimuth compression.

Table 3. Resource estimation.

Component	Shared Memory Type	
	Lockable	Unprotected
Flipflop	6%	5%
LUTs	8.5%	7%
BRAM	3.6%	3.6%
IOB	73%	72.5%
DSP48s	12.5%	10%

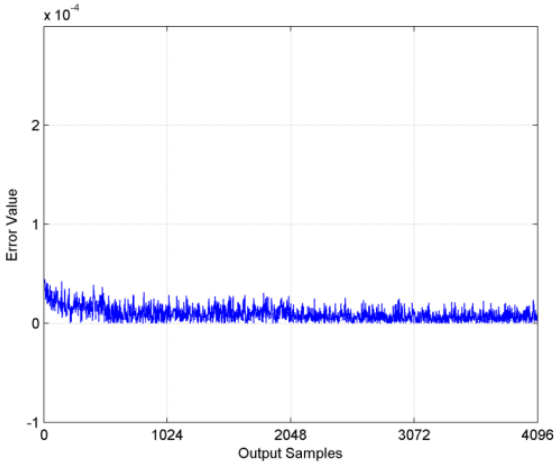

Fig. 12. Error level of FPGA output in comparison with MATLAB® simulated results.

9. Conclusion

Real-time implementation of UWB-OFDM SAR imaging system is investigated using high speed ML605 system implementing Xilinx Virtex-6 FPGA. FPGA based hardware output of the FFT/IFFT computations and complex multiplications as part of match filtering are compared with simulated results obtained under MATLAB® environment. Experimental results reveal that FPGA implementation based on the shared memory approach is effective in implementing SAR imaging algorithm and providing a real-time tool for SAR imaging. The prospects of relatively inexpensive implementation of UWB-OFDM based SAR system heavily based on digital components will become a subject of subsequent study.

Acknowledgement

This work is funded by National Plan for Sciences and Technology (NPST), Saudi Arabia, under project number: 08-ELE262-2.

References

[1] M. Soumekh, Synthetic Aperture Radar Signal Processing with MATLAB Algorithms. Wiley, 1999.
[2] J. D. Taylor. Ultra-wideband Radar Technology. CRC Press, 2001.
[3] L. Hanzo, T. Keller, OFDM and MC-CDMA. Wiley, 2006.
[4] A. Hossain, I. Elshafiey, M. Alkanhal, "High resolution UWB SAR based on OFDM Architecture", Asia-Pacific Int. Conf. on Synthetic Aperture Radar (APSAR). Seoul, South Korea, 2011, P.1-4.
[5] A. Hossain, I. Elshafiey, M. Alkanhal, A.Mabrouk, European Radar Conference (EuRad), Manchester, United Kingdom, P.1-6, 2011.
[6] Ian G. Cumming, Frank H. Wong, Digital Processing of Synthetic Aperture Radar Data, 2nd ed. London, United Kingdom: Artech House, 2004.
[7] Xilinx Inc.(January, 2012), Xilinx Website. [Online], Xilinx ISE Design Suite 13.4, http://www.xilinx.com/
[8] Xilinx Inc. (September, 2011), Xilinx Website. [Online], http://www.xilinx.com/products/devkits/EK-V6-ML605-G.htm

*Corresponding author: ahossain@ksu.edu.sa