Fast Realization of 3-D Space-Time Correlation Sea Clutter of Large-Scale Sea Scene Based on FPGA: From EM Model to Statistical Model

Shuhao Zhang, Jinxing Li, Yachao Li, Pengbo Wei, and Min Zhang, *Member, IEEE*

***Abstract*—Memoryless nonlinear transform (MNLT) method was widely used in the statistical model for sea clutter simulations. When the radar scattering data sets were obtained, we can simulate large scene and long-time varying 3-D sea surface scattering using expended power spectral quickly and accurately. Compared with the personal computer platform, field-programmable gate array (FPGA) has the unique merit of energy efficiency, high performance and adaptability. In this article, we proposed a novel architecture for implementing the 3-D MNLT algorithm on FPGA using high-level synthesis. As the simulation size increases, the demand for storage resources will also increase rapidly, and the on-chip memory resource will be limited on FPGA. Aiming at these problems, we divided the 3-D space-time simulation into a 2-D spatial simulation and a 3-D temporal simulation, so that we can make full use of the off-chip memory. Our design employs multiple on-chip buffer structures to decrease the transfer time of internal and external data on the FPGA. We also design a dataflow inverse fast Fourier transform processing engine (PE). The dataflow implementation overlapped butterfly operation, increasing concurrency and the overall throughput of this PE. Experimental results show that we can obtain the same accuracy and higher efficiency on a Xilinx Zynq XC7Z100 SoC platform.**

***Index Terms*—Inverse fast Fourier transform (IFFT), memoryless nonlinear transform(MNLT) method, sea clutter, space-time correlation, zynq.**

I. Introduction

DUE to the external factors such as sea surface wind speeds, changes of the atmospheric pressure and changes of the gravity, the sea surface will exhibit random height fluctuations. Radar signals scattered by the time-varying sea surface is named sea clutter. Sea clutter simulation is an integral part of the commissioning and testing radar performance [1]. With the developing techniques of high-resolution imaging and target detection, the traditional simulations appear to be unsatisfactory. Generally, the acquisition of radar clutter is expensive, and the various ranges of testing environments are limited, such as radar parameters, viewing geometry, and weather conditions. The statistical model of the dynamic sea clutter is the basic model of the returns from radar, which is an important tool in the understanding of maritime radars. Recently, the temporal variations of sea clutter increasingly play a significant role in dynamic sea surface electromagnetic (EM) scattering and target detection [2], [3]. There are two categories of sea clutter models: the amplitude statistical approach-based models and models based on the EM [4]. EM scattering models, such as Kirchhoff approximation [5], two-scale model [6], small slope approximation [7], [8], capillary wave modified facet scattering model (CWMFSM) [9], etc. The EM scattering simulations using models abovementioned are difficult work when dealing with the electrically large and long time case. Statistical models based on the statistical characteristics of available sample data provide the preferred approaches for the abovementioned issue. At present, there are three most widely used statistical models: memoryless nonlinear transform (MNLT) [2], [10]–[12], spherically invariant random processes [13] and auto-regressive model [14]. These kinds of models are based on the probability density function (PDF), and spectral density function to generator correlated non-Gaussian random properties. Concerning the MNLT method; however, it is based on the spectral representation techniques and executed through inverse fast Fourier transform (IFFT) algorithm, hence achieving a huge advantage in three-dimensional (3-D) large maritime scene scattering simulation [12].

The 3-D sea clutter simulation with the statistical model had demonstrated success on the PC platform, but the field-programmable gate array (FPGA) implementation only stay at simulation 1-D correlated sequence [15]–[17] and 2-D temporal and spatial correlated clutter sequence [18]. Clutter simulation using the statistical model based on GPU is still at 1-D sequence simulation [19]. The focus of implementation of the 1-D simulation is to change the probability density distribution of the random sequence, and a Gaussian power spectrum is used to change the autocorrelation of the sequence simply. We pay more attention to changing the power spectrum of the 3-D sea clutter based on the actual situation fleetly. Among the other platforms such as CPU and GPU, FPGAs have emerged as a promising solution due to its high performance, energy efficiency, and reprogram ability [20]. More critical, FPGA design automation tools developed rapidly over the past decade.

Manuscript received June 4, 2020; revised August 24, 2020 and November 2, 2020; accepted November 26, 2020. Date of publication December 8, 2020; date of current version January 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61771355 and Grant 41701386, and in part by the National Key Laboratory Foundation under Grant 6142A01180206 and Grant 621902Y010202. *(Corresponding author: Min Zhang.)*

Shuhao Zhang, Jinxing Li, Pengbo Wei, and Min Zhang are with the School of Physics and Optoelectronic Engineering, Xidian University, Xi'an 710071, China (e-mail: mzhang@mail.xidian.edu.cn; gstarljx@163.com; pbwei@xidian.edu.cn; mzhang@mail.xidian.edu.cn).

Yachao Li is with the School of Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: ycli@mail.xidian.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3043272

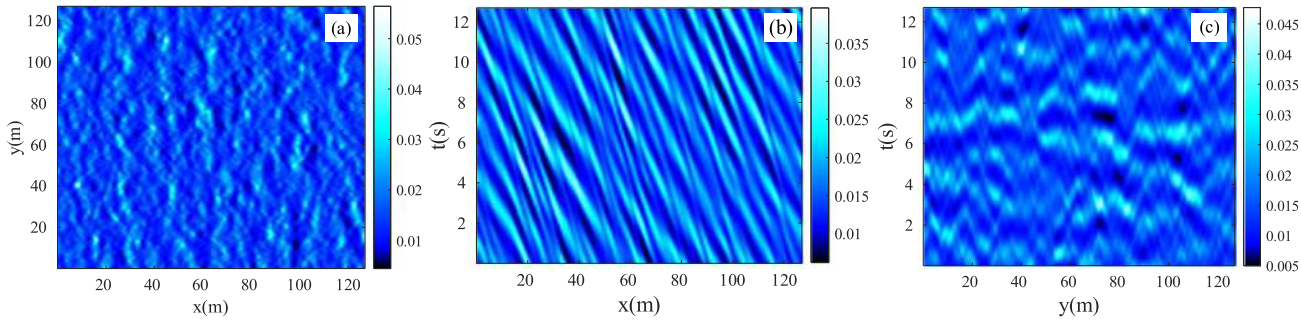


Fig. 1. Amplitude of 3-D scattering from the sea surface generated by the EM model. (a) x*O*y plane. (b) x*O*t plane. (c) y*O*t plane.

TABLE I
EM MODEL SIMULATION PARAMETERS

Area(m^2)	128×128	Frequency(GHz)	5
Grid size(m^2)	1×1	θi(deg)	40
simulation time(s)	12.8	φi(deg)	0
Time interval(s)	0.1	V_{10m}(ms^{-1})	5
Wind direction	x-axial	Polarization	HH

High-level synthesis (HLS) tools using C/C++ or OpenCL have significantly increased programmer productivity when designing at FPGA platforms [21], [22]. A more recent software-defined tool like Xilinx SDSoC provides a comprehensive design environment for heterogeneous Zynq system on chip (SoC) and deployment. We can use SDSoC to realize algorithm porting as well as the software and hardware design rapidly [23], [24].

The difficulty of 3-D MNLT simulation was the need for a large amount of intermediate data cache. We designed a separable 3-D computing structure to solve this problem. This design took advantage of the on-chip memory and the off-chip memory. It also gave our system the flexibility to simulate different sizes and duration. In this article, the CWMFSM was adopted as the EM scattering model to generate the sea clutter and then the statistic and correlation properties were acquired for the establishment of a statistical model in the subsequence.

The rest of this article was organized as follows: Section II briefly presented the generation of simulated data of dynamic sea surface EM backscattering. Section III contained the MNLT method to the 3-D case and the fast Fourier transform (FFT) principle. Section IV devoted to describe the design detail of our system and optimization method. Section V analyzed the experimental results and displayed the comparisons with software results on the PC. Then, the consumed time between the two platforms was shown. Conclusions were presented in Section VI.

II. GENERATION OF 3-D SEA CLUTTER RESULT

A. EM Scattering Model

Statistical methods of simulation need sample data. In the absence of measured data, we use the EM scattering model CWMFSM to calculate small-size sea clutter as sample data of statistical simulation. The CWMFSM is a representative two-scale model. It assumes that the main contribution to the radar echo is the part where the Bragg capillary wave propagates away from or near the radar sight and calculates the total field based on the two-scale small surface element field [25].

In this part, the 3-D space-time correlated sea scattering result is simulated based on the facet-based sea surface scattering model. Simulation parameters are summarized in Table I. Moreover, the relative dielectric constant of the seawater is calculated by the Klein dielectric constant model [26] at 20°C and 35‰ salinity. The *xOy* section view at 0.64s is displayed in Fig. 1.

B. Extraction of the Extended Power Spectrum

As the sample of sea surface backscattering has been obtained, we should first calculate its power spectrum. After that, we use small-scale power spectral to get expanded power spectral by interpolation operation.

For the 3-D random signal, the power spectral density (PSD) function $S(k_x, k_y, \omega)$and autocorrelation function $R(x, y, \tau)$ of the stochastic field consists of a pair of Wiener–Khinchin relationships. For the discrete-time case, the transform is shown as follows:

$$R(x,y,\tau) = \sum_{k_x=-\infty}^{\infty}\sum_{k_y=-\infty}^{\infty}\sum_{\omega=-\infty}^{\infty} S(k_x,k_y,\omega) \cdot e^{j2\pi(k_x x+k_y y+\omega\tau)} \tag{1}$$

$$S(k_x,k_y,\omega) = \frac{1}{(2\pi)^3}\sum_{k_x=-\infty}^{\infty}\sum_{k_y=-\infty}^{\infty}\sum_{\omega=-\infty}^{\infty} R(x,y,t) \cdot e^{-i(k_x x+k_y y+\omega t)} \tag{2}$$

where, x and y are separation distances in two directions; k_x and k_y are wave numbers in x and y directions, respectively, ω is the angular frequency and τ represents the time interval.

We assume $F_{\mathrm{EM}}(x, y, t)$ representative an EM simulated random process and $f_{\mathrm{EM}}(x, y, t)$ is one of its samples. The following formulas can express the PSD:

$$S(k_x,k_y,\omega) = \lim_{L_x,L_y,T\to\infty} E\left[\frac{1}{\sigma_f^2 L_x L_y T}\left|\mathcal{F}\left\{F_{\mathrm{EM}}(x,y,t) - E\left[f_{\mathrm{EM}}(x,y,t)\right]\right\}\right|^2\right] \tag{3}$$

In the abovementioned equation, $E(\bullet)$ and σ_f^2 respectively represent mean value and variance of the specific

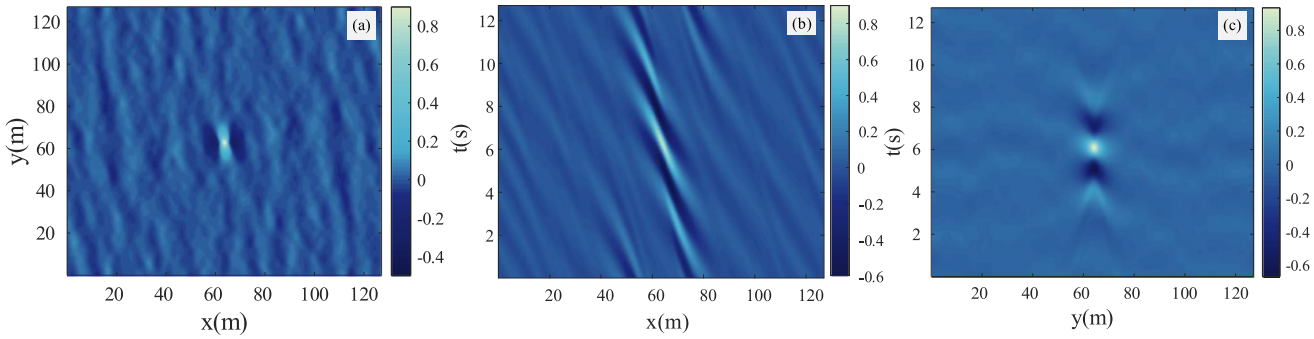


Fig. 2. Section views of the 3-D correlation function of the amplitude of the sea surface generated by the EM model. (a) x*O*y plane. (b) x*O*t plane. (c) y*O*t plane.

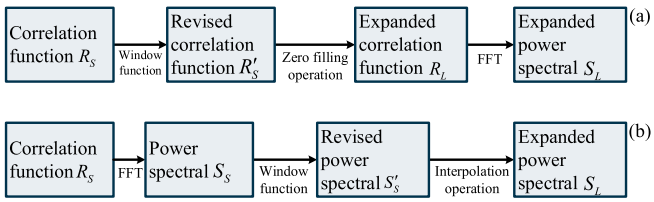


Fig. 3. Acquisition procedure of the expanded power spectral. (a) Correlation function zero filling approach. (b) Power spectral interpolation approach.

sample $f_{EM}(x, y, t)$; $\mathcal{F}(\bullet)$ represents fast Fourier transform; L_x, L_y, Tspans in *X, Y, T* dimensions, respectively.

To give the comparison with the MNLT method, the amplitude of 3-D scattering from the sea surface generated by the EM model in *xOy* plane, *xOt* plane, and *yOt* plane are shown in Fig. 1. Section views of 3-D correlation function derived from the CWMFSM simulation results in *xOy* plane, *xOt* plane, and *yOt* plane are shown in Fig. 2.

On the strength of expanded power spectral density, two main methods are shown below. The first approach, in the time dimension, is based on the fact that correlation function values of a stochastic process tend to be zero as the absolute value of the variables increase. We can use zero filling operation to expand correlation function and make the transition from expanded correlation function to expanded power spectral. The second approach, we can use the interpolation operation to generate expanded power spectral. Not only did the spectrum leakage distort the spectrum estimation, but also caused blurring and distortion in the spectrum estimation. To improve this problem, we used a window function to eliminate the sudden change between the original and expanded domain. When the small correlation function values were available, the window function was added before the zero filling operation to eliminate the sudden change between the original R_S and expanded result R_L. Then, we used FFT operation to get expanded power spectral S_L, as shown in Fig. 3(a). The acquisition procedure of the second approach was shown in Fig. 3(b). This approach performed the window function and interpolation operation on the original power spectral itself. We used two approaches to generate the expanded power spectral. The spatial resolution of the spectral is 256 $\times$ 256 with the time resolution of 256 frames. We compared the effect of two approaches in R(x,0,0), R(y,0,0), R(t,0,0) as shown in Fig. 4. For the comparison purpose, the same Kaiser window is added in two approaches [27]. Although using window function to reduce the sudden change between the original and expanded domain, the truncation error caused by zero-filling operation cannot be eliminated, manifested by the correlation function of MNLT simulation oscillating near zero. Through comparison, we chose the power spectral interpolation approach to generate expanded power spectral.

When we obtained the new expanded power spectral, we can employ the MNLT method to generate the large scene and long-time sea-surface scattering results.

III. 3-D MNLT Method for Sea Clutter

The MNLT method was extended to the 3-D case for establishing a statistical model, which contained two major operational procedures, those were, the simulation of the correlated Gaussian process of zero mean and unit variance, and the transformation from the Gaussian process into the desired non-Gaussian process.

The 3-D correlated Gaussian process $f_{\text{states}}(x, y, t)$ can be expressed by the following series as $L_x, L_y, T \to \infty$ [28], namely,

$$\begin{aligned} f_{\text{stats}}(x, y, t) &= \sum_{m=-L_x}^{L_x} \sum_{n=-L_y}^{L_y} \sum_{l=-T}^{T} A_{mnl} \\ &\quad \times \cos\left(k_m^x + k_n^y + \omega_t t + \phi_{mnl}\right) \\ k_m^x &= m \cdot \Delta k_x, m = -L_x, \ldots, L_x \\ k_n^y &= n \cdot \Delta k_y, n = -L_y, .., L_y \\ \omega_l &= l \cdot \Delta\omega, l = -T, \ldots, T \end{aligned} \tag{4}$$

where ϕ_{mnl} is the independent random phase angle uniformly distributed in range $[0, 2\pi]$; k_{ux}, k_{uy}, and ω_u represent the upper cutoff wave numbers and frequency in corresponding dimensions; $\Delta k_x = k_{ux}/L_x$, $\Delta k_y = k_{uy}/L_y$ and $\Delta\omega = \omega_u/T$ represent discrete interval. At last, A_{mnl} is defined in (5) and $S(k_x, k_y, \omega)$ is the expended power spectral in Section II.

$$A_{mnl} = \sqrt{S\left(k_x, k_y, \omega\right) \Delta k_x \Delta k_y \Delta\omega} \tag{5}$$

Additionally, the multiply-accumulate calculation in (4) can be accelerated by using the IFFT technique. When the Gaussian process was completed, the transformation from the Gaussian

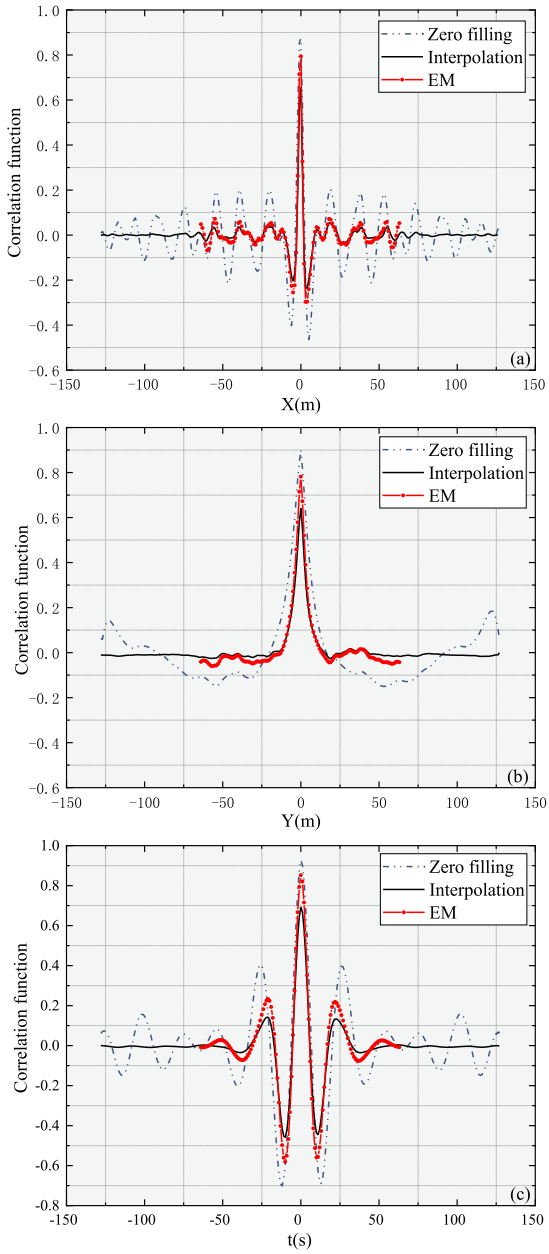


Fig. 4. Comparison of two approaches to expand power spectral. (a) R(x,0,0). (b) R(0,y,0). (c) R(0,0,t).

process into the desired non-Gaussian process was implemented as follows:

$$g(x, y, t) = F_{\text{non}-G}^{-1}[F_G(f_{\text{states}})] \tag{6}$$

where F_G represents the cumulative distribution function (CDF) of the Gaussian process f_{stats} and $g(x, y, t)$ is the MNLT result. For the non-Gaussian process, we use log-normal distribution to fit the PDF of EM sea-surface scattering result. The CDF with the shape parameter σ and scale parameter μ is given by

$$F_G(x) = \frac{1}{2} + \frac{1}{2} erf\left[\frac{\ln x - \mu}{\sqrt{2}\sigma}\right] \tag{7}$$

TABLE II
COMPARISON OF DIFFERENT ON-CHIP AND OFF-CHIP MEMORY STORAGE OPTIONS

	SD card	DDR3	BRAM	FFs
Count	1	1-4	thousands	millions
Total size	GBytes	GBytes	MBytes	100s of KBytes
Total bandwidth	MBytes/sec	GBytes/sec	TBytes/sec	100s of TBytes/sec

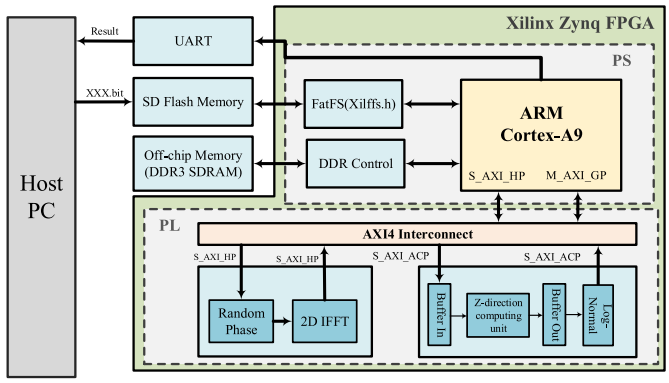


Fig. 5. Architectural diagrams of our MNLT system.

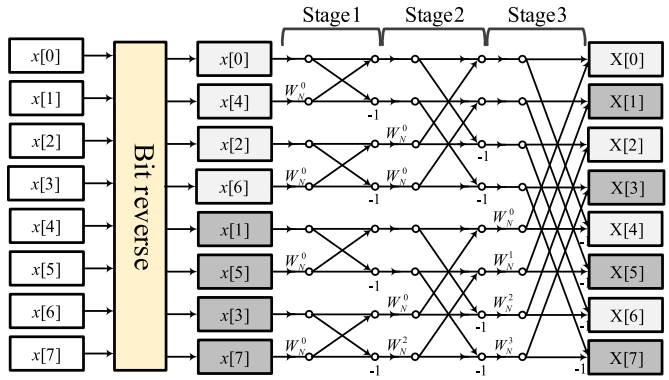


Fig. 6. Radix-2 FFT algorithm with $N = 8$.

IV. FPGA ARCHITECTURE DESIGN

In this section, we outline the optimizations and techniques we used, the general system architecture, hardware and software functional partitioning, then describe the specific compute units and optimization scheme.

A. System Architectural diagrams Overview

Large scale and long-time simulations involved a large amount of data. For them, on-chip memory was insufficient to store all the data due to the actual requiring of gigabytes external memory. Therefore, we set up four levels of storage hierarchy for our system:

1) external nonvolatile memory;
2) external volatile memory;
3) on-chip buffers; and

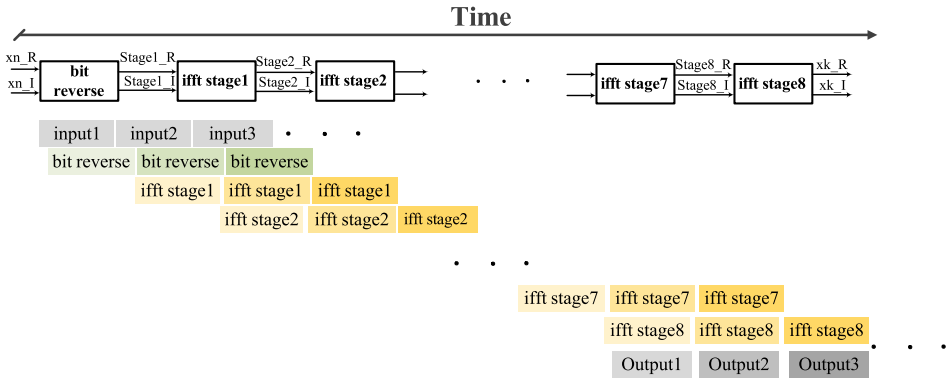


Fig. 7. Dataflow for IFFT with size = 256.

4) registers associated with the processing engines (PEs) [29].

As shown in Table II, with the transmission speed increasing, the available storage space decreased rapid-ly [32].

The basic flow of our system was to fetch data from external nonvolatile memory to external volatile memory and then load data from external volatile memory to on-chip buffers, at last feed them into registers and PEs. We used the AXI4 bus communicate between processing system (PS) and programmable logic (PL). For the system, we used the float data type. Our system architecture, shown in Fig. 5, was mainly composed of three parts: peripheral part, software part, and composed hardware part. Peripherals included a secure digital (SD) card to storage power spectral data and simulation result, a universal asynchronous receiver-transmitter to output the state of the calculation process, and the off-chip memory to cache large amount of intermediate data. Software running in the PS side of Zynq, its tasks included controlling Zynq to communicate with the host PC, carring out reading and writing for SD card and double data rate (DDR) controlling. On the PL part, we established multiple processing modules following every calculation step. In the following, we mainly introduced the parts that make up the correlated Gaussian random processes.

In our system, the PS side was used to control the transmission and storage of data, and the PL side was used to complete the calculation. The 3-D MNLT method can use the 3-D IFFT to speed up the calculation of coherent Gaussian process, that was, the data in the three dimensions of *X, Y,* and *T* need to be buffered for calculation. This required the use of a larger capacity DDR3 memory, but the data transmission would consume much time. Our design goal was reducing the number of off-chip memory accesses, which required full use of on-chip memory. According to the feature that the 3-D IFFT calculations were independent of each other, this article used a paged cache method to sequentially read a 2-D data matrix of a size to the on-chip cache and calculate it. After calculating all the 2-D data in the *X–Y* dimension, the *T*-dimension data was calculated in order. After generating the Gaussian process, the transformation from the Gaussian process into the ono-Gaussian process was implemented by a nonlinear transformation simply.

B. Dataflow IFFT PE Design

FFT was proposed by Cooley–Turkey to accelerate the machine's computing speed, which can reduce constant complex multiplications from $O(n^2)$ to $O(n\log(n))$ with no accuracy loss. Cooley–Turkey 1D-FFT was computed as follows:

$$\begin{cases} X(k) = X_1(k) + W_N^k X_2(k) \\ X(k+\frac{N}{2}) = X_1(k) - W_N^k X_2(k) \end{cases} k = 0, 1, 2, \ldots, N/2-1 \tag{8}$$

where $W_N^k = e^{-2i\pi k/N}$ and $X_1(k), X_2(k)$ are defined as follows:

$$\begin{cases} X_1(k) = \sum_{r=0}^{N/2-1} x(2r) W_N^{2rk} \\ X_2(k) = \sum_{r=0}^{N/2-1} x(2r+1) W_N^{2rk} \end{cases} \tag{9}$$

In (8) and (9), FFT was inducted from small-scale discrete fourier transform (DFT) divided according to the parity, which was also named butterfly computation. It should be noted that the Cooley–Turkey algorithm was only valid when FFT size was a power of 2. Fig. 6 showed a brief example of the FFT algorithm with FFT size = 8. In this example, it took three stages to complete FFT. In these stages, several butterfly computations were performed with two points (Radix-2 FFT). IFFT was similar to FFT. The rotation factor of IFFT defined as $W_N^k = e^{2i\pi k/N}$. Traditional IFFT implementations focus on data with a large size, which may contain hundreds of signals. Due to sequential calculation, it took several steps to complete IFFT. For large scale 3-D case, large amounts of IFFT calculation would cost unacceptable computing latency. We can see in Fig. 6 that the input of each step was the output of the previous stage. Based on this feature, we leveraged dataflow to operat IFFT. We partitioned bit reverse and every stage to function modules, streaming data between them through explicit interfaces according to the dataflow between them [31], [32]. In contrast to conventional pipelining, functions arranged in a streaming dataflow architecture were scheduled separately, as shown in Fig. 7. The dataflow allowed stages to overlap in their operation, increasing concurrency and the overall throughput of this unit.

Algorithm 1. Two-dimensional computing unit

Input: *N*×*N row-col Array.*

Output: *N*×*N row-col real data array; N*×*N row-col image data array.*

```
1: Function IFFT_2D(xn, xk_real, xk_imag)
2:    buf_2D_R[SIZE][SIZE], buf_2D_I[SIZE][SIZE];
3:    tran_R[SIZE][SIZE], tran_I[SIZE][SIZE];
4:    read_data(xn, buf_2D_R, buf_2D_I);
5:    random_phase(buf_2D_R, buf_2D_I);
6:  for row ← 0 to N-1 do
7:        ifft(buf_2D_R[row][], buf_2D_I[row][], transpose_R[][row], transpose_I[][row]);
8:  end for
9:  for col ← 0 to N-1 do
10:       ifft(transpose_R[][col], transpose_I[][col]);
11: end for
12:   write_data_out(transpose_R, transpose_I, xk_real, xk_imag);
13: End Function
```

Fig. 8. HLS pseudocode for 2-D computing unit.

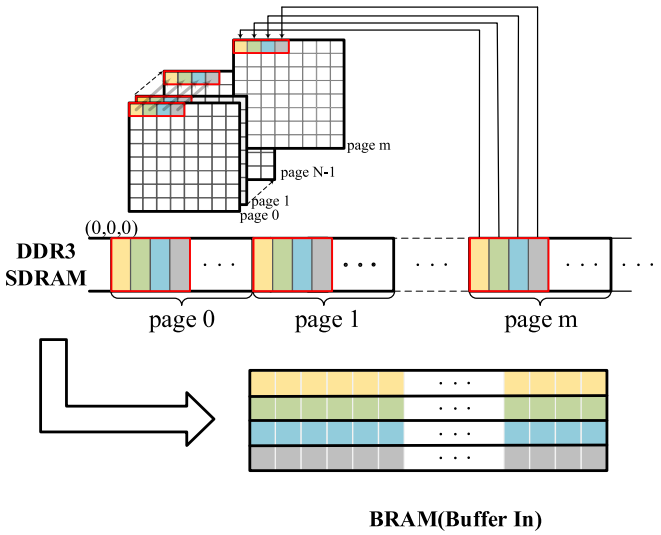


Fig. 9. Schematic diagram of small-size data sequential read/write.

Algorithm 2. Z-direction computing unit parallel optimization.

Input: *BLOCK_N*×*N Z-Dimension Array real data; Z-Dimension Array image data.*

Output: *BLOCK_N*×*N Z-Dimension Array real data; Z-Dimension Array image data.*

```
1.  Function Z_direction_unit(pagePtr_R , pagePtr_I)
2.    blockbuf_R[BLOCK_N][SZIE];
3.    blockbuf_I [BLOCK_N][SZIE];
4.    #pragma HLS ARRAY_PARTITION variable= blockbuf_R, blockbuf_I
5.    read_data(pagePtr_R, pagePtr_I, blockbuf_R, blockbuf_I)
6.    …
7.    for block ← 0 to BLOCK_N-1 do
8.      #pragma HLS UNROLL factor= BLOCK_N
9.      ifft(blockbuf_R[], blockbuf_I);
10.   end for
10.   …
11.   write_data(blockbuf_R, blockbuf_I, pagePtr_R, pagePtr_I)
12. End Function
```

Fig. 10. HLS pseudocode for *Z*-direction computing unit with unroll optimization.

C. 2-D Computing Unit

In our analysis, a simulation with a spatial resolution of 256 × 256 samples and a time resolution of 256 frames will expend 128 MB storage space. However, there are only 26.5 Mb available block RAM (BRAM) in our FPGA. Therefore, the power spectral data and intermediate data must be stored in off-chip memory, namely DRAM, when mapping large scale 3-D MNLT on FPGA. The DRAM accesses are of higher latency and energy than on-chip BRAM memory accesses.

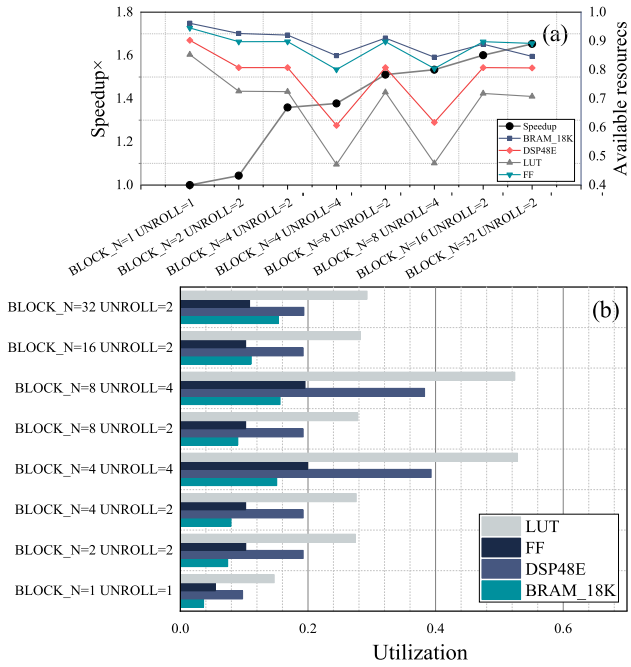


Fig. 11. Resource utilization and performance results for different *BLOCK_N* and UNROLL factors. (a) Remaining resource proportion and speedup compare. (b) Resource utilization of the *Z*-dimension unit using different *BLOCK_N* value and UNROLL factors.

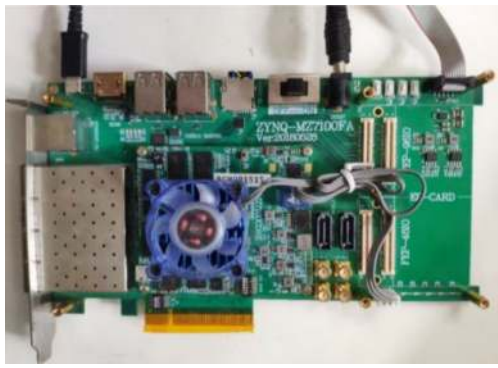


Fig. 12. Appearance of the FPGA platform.

Therefore, reducing the number of external memory accesses is the key approach to improve the overall performance and energy efficiency.

N-dimensional IFFT is the composition of the *N* dimension 1-D IFFTs. Based on this characteristic, we can perform along 1-D at a time in any order. The minimum number of DRAM accesses is achieved by allowing sufficiently large on-chip buffers and proper loop computing orders, such that every power spectral data needs to be transferred from DRAM only once. Otherwise, the same data has to be read multiple times from DRAM to be consumed for multiple tiles.

Fig. 8 showed the HLS pseudocode of the 2-D computing unit and demonstrated the calculating process. We used function names to replace the actual realization process. The data type of input data was the real number, and after 2-D IFFT, we got complex data type results. To reduce the number of data transfers, we cached the 3-D power spectrum of each

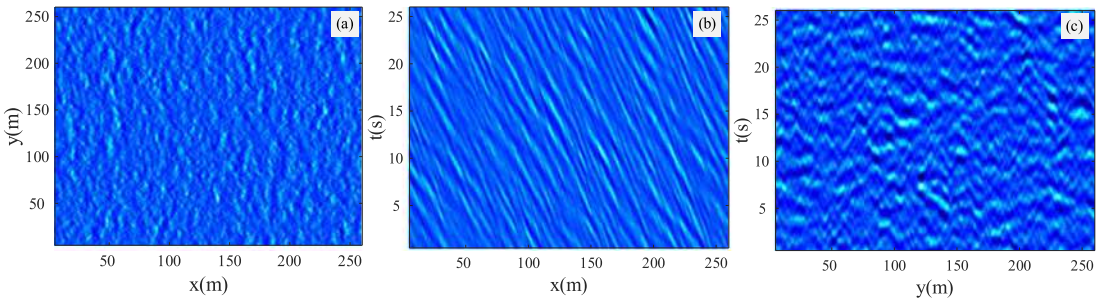


Fig. 13. Amplitude of 3-D scattering from the sea surface generated by the MNLT model on the Zynq platform. (a) *xOy* plane. (b) *xOt* plane. (c) *yOt* plane.

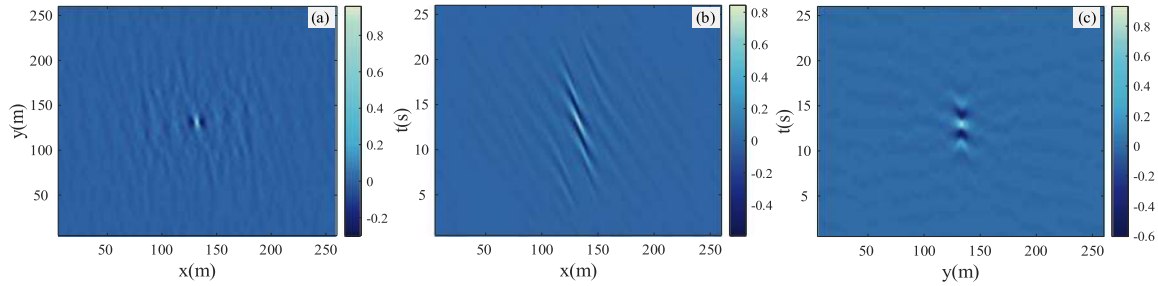


Fig. 14. Section views of 3-D correlation function of amplitude of the sea surface generated by the MNLT model on the Zynq platform. (a) *xOy* plane.

row-col page through on-chip memory and then calculated the random phase and 2-D IFFT in turn. In the calculation process, to improve the throughput for the design, we used transpose buffers for caching calculation results from the row IFFT. The 2-D buffers were implemented with core RAM_T2P_BRAM to implement variable var in the RTL. RAM_T2P_BRAM is a core, described as a true dual-port RAM with support for both read and write on both ports implemented with a block RAM in the technology library [30]. Coordinate with dataflow IFFT PE can implement data reading, IFFT calculation and caching resulted in an efficient pipelined loop.

We used row-major order for storing multidimensional arrays in off-chip memory. This way, every page data read and write were performed on the consecutive addresses, and we can get the best performance when processing data.

D. Z-direction 1-D Computing Unit

CPUs process sequential data more efficiently than nonsequential data. Since the 3-D matrix was stored in row-major order, each *Z*-direction sequence with a length of *N* can only be read out in a cell. Random accessing in linear memory needed to addressing each time when a data was read. The sparse data locality led to severe access latency. To reduce the influence of this problem, as shown in Fig. 9, we leveraged the technology of cache prefetching to the design of small-size data sequential read/write. Continuous data were fetched from the DRAMs and stored by BRAMs, and each row of on-chip buffer stored one line of *Z*-direction sequence. This working model still extracted the data sequence from the Z-dimension in the row order from (0, 0) to (N-1, N-1), but the minimum element of data read became *BLOCK_N* each time.

To reduce judgment, *BLOCK_N* was required to meet the exact division of *N*. In this way, the read/write operating times to traverse all the data became $(N/BLOCK\_N) \times N \times N$ and the off-chip memory can work at burst access mode. It seemed that the larger *BLOCK_N* would be more efficacious. Nevertheless, the sizeable *BLOCK_N* value increased the consumption of resource. Accordingly, we had to balance resource consumption and performance.

The optimization process of our proposed acceleration scheme was presented in Fig. 10, which included an appropriate selection of the on-chip memory partitioning optimization and loop unroll optimization. The physical implementation of memories had only a limited number of read ports and write ports, which can limit the throughput of a load/store intensive algorithm. The memory bandwidth can be improved by splitting up the original array into multiple smaller arrays. We split the [BLOCK_N][N] array into *BLOCK_N* arrays of dimension [1][N] by array_partition, effectively increasing the number of load/store ports. Multiple buffers operation facilitated subsequent calculation implementation. In the case of none of the statements in the loop depend upon any of the data generated in the previous iterations, we can unroll the loop to parallel several IFFT PEs to reduce the runtime. The parallelism value is not as bigger as better. Further comparisons were shown in Fig. 11. Fig. 11(b) showed the resource utilization of the *Z*-dimension unit using different values IFFT PEs and different size buffers. We considered that the acceleration ratio was 1 in the case of $BLOCK\_N = 1$ and $UNROLL\ factor = 1$, and gave the acceleration proportion of different operation levels

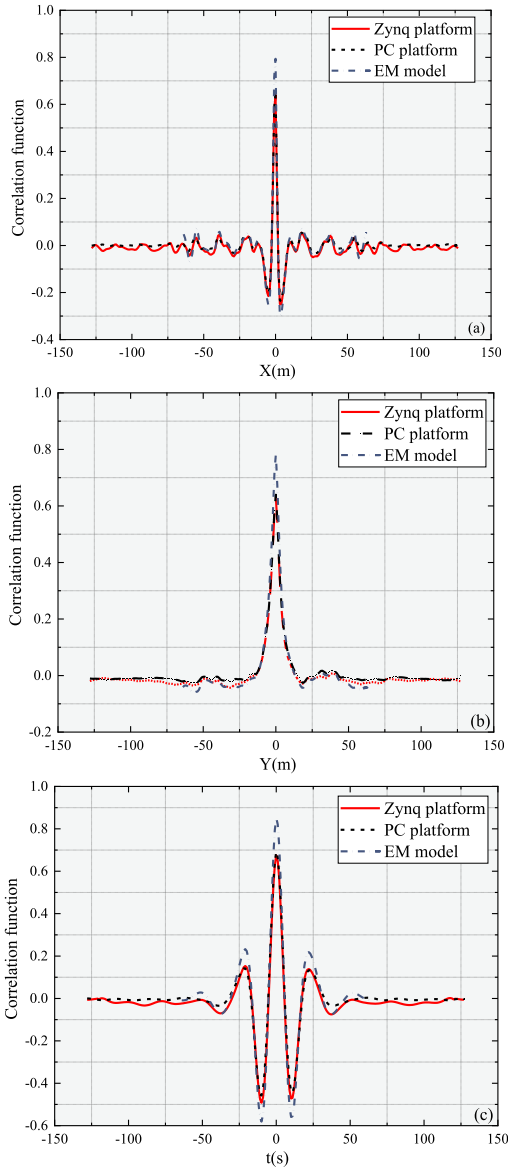


Fig. 15. Comparative 1-D plots of the correlation functions from the EM model and MNLT models generated on the two platforms spectral. (a) R(x,0,0). (b) R(0,y,0). (c) R(0,0,t).

compared with level 1 in Fig. 11(a). Besides, the remaining proportion of resources was given in Fig. 11(a) to prove that the length of the data read at one time is the most significant limitation of the module efficiency. Excessive consumption of resources will cause path delays during routing, which does not contribute much to efficiency improvement. In summary, we chose *BLOCK_N* = 32 and *UNROLL factor* = *2* as the optimal length for continuous reading and writing of small data.

V. Experimental Results and Analysis

A Xilinx Zynq-7100 system on chip (SoC) board was chosen as our development device, which was equipped with a dual-core ARM Cortex-A9 processors integrated with Kintex-7 based programmable logic. Besides, the board was equipped with two

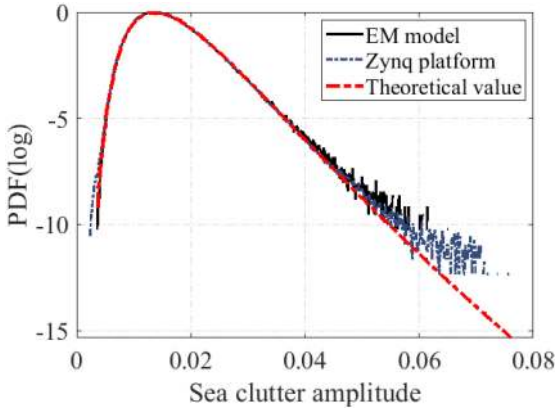


Fig. 16. Comparison of PDF of 3-D sea clutter between EM models and Zynq platform simulation results.

banks of 512MB DDR3 DRAMs. The FPGA platform was shown in Fig. 12. We made use of Xilinx SDSoC 2017.4 as a primary design tool, which provided an embedded C/C++ application programming, including an eclipse integrated design environment and a comprehensive development platform. We compared our design against computing platform: an Intel i5-7400 multicore processor and 16GB main memory. The serial C++ program was executed on the Windows 7 operating system. The rest of this section is organized as follows. First gives a comparison of the simulation results on the two platforms. Then gives a comparison of the processing time of the MNLT algorithm in C++ and Zynq, we analyzed the advantages of the Zynq implementation of MNLT in terms of logic resources utilization and data processing speed.

A. Amplitude Generation of the 3-D Sea Scattering Results on the Zynq Platform

When comparing the FPGA simulation result in Fig. 13 with the original simulation result in Fig. 1, we can see that the section views in different planes all take on the quite similar textures, which suggested the MNLT method effectiveness for the 3-D result simulation running on the Zynq platform.

Additionally, Fig. 14 provided the 3-D correlation function in the three-section views: *xOy* plane, *xOt* plane, and *yOt* plane. Through the comparison of the figures in Fig. 2 and Fig. 14, it was seen that these correlation functions had almost the same appearance. As the scale in the time dimension and spatial dimension increasing, points exceeding the correlation length appear not to correlate. In the section views of the correlation function, we can see that except the middle area manifesting as strong correlation, most areas tend to be zero.

Moreover, to give a more explicit comparison, the comparative 1-D plots of the correlation functions in three dimensions, namely R(x,0,0), R(0,y,0), and R(0,0,t), were reported in Fig. 15, respectively. We compared the correlation function of the original EM result with the correlation function of simulation results from both the PC platform and the Zynq platform. Through the corresponding correlation function comparisons, we can easily

TABLE III
RESOURCE UTILIZATION FOR THE ZYNQ IMPLEMENTATION OF 3-D MNLT

Module	BRAM	DSP	FF	LUT
1-D IFFT	50	192	25939	34602
Log-normal	1	11	2740	3884
Random phase	0	60	10267	9933
Z-direction unit	232	391	60157	81191
2-D computing unit	574	263	39020	49128
Total	844	794	170623	191961
Utilization	55.9%	39.31%	30.75 %	69.2%

TABLE IV
COMPARISON OF THE PERFORMANCE FOR 3-D MNLT RUNNING ON THE TWO PLATFORMS

Data type	Time(ms)	Fmax(MHz)	Speedup
Software	49905	3000	1
Hardware	7112	166.7	7.017x

observe the simulation performances and accuracy of the Zynq platform.

Besides, the simulated 3-D space-time sea surface scattering by the EM model is of log-normal distribution and the maximum likelihood estimate results of the shape and scale parameters are -4.196925 and 0.30979, respectively. Besides, the comparison of PDF between EM models and Zynq platform simulation results were illustrated in Fig. 16.

They can fit well with each other together with theoretical result.

B. Cross-Platform Performance Comparison

From the previous section, we can see that the 3-D MNLT running on the Zynq platform was very close to the PC platform result. The PL side of the XC7Z100-2FFG900I chip used in this article contains 444k logic resources (Logic Cells), 277400 look up tables (LUTs), BRAM with a total capacity of 26.5Mb, 2020 DSP processing units (DSP slices) and 554800 Flip-flops (FFs). Table III illustrated the utilization of the different modules implemented in the PL of a 3-D MNLT system on FPGA of the simulation with a spatial resolution of 256×256 samples and a time resolution of 256 frames.

Table IV showed the implementation performance of the two platforms. The performance of the 3-D MNLT had been greatly improved compared with the PC platform, because the FPGA platform had a good acceleration effect for the computation-intensive model.

VI. CONCLUSION

In this article, a dynamic sea surface EM scattering simulation was implemented on FPGA, which can simulate large-scale and long-term 3-D space-time related sea surface scattering and achieve excellent results. The MNLT statistical algorithm can use the IFFT to speed up the calculations, and this calculation-intensive algorithm was inherently suitable for FPGA calculations. The experimental results showed that SDSoC could be used to write code efficiently to achieve our design. The simulation results obtained by FPGA calculations showed high accuracy and the calculation speed was significantly improved compared to the PC platform.

ACKNOWLEDGMENT

The authors would like to thank the Editor who handled their article and the three anonymous reviewers for providing truly outstanding comments and suggestions that significantly helped us improve the technical quality and presentation of our article.

REFERENCES

[1] S. Watts, "Modeling and simulation of coherent sea clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 3303–3317, Oct. 2012.
[2] J. Wang and X. Xu, "Simulation of correlated low-grazing-angle sea clutter based on phase retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3917–3930, Jul. 2015.
[3] S. Shi, X. Liang, P. Shui, J. Zhang, and S. Zhang, "Low-Velocity small target detection with doppler-guided retrospective filter in high-resolution radar at fast scan mode," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8937–8953, Nov. 2019.
[4] K. D. Ward, S. Watts, and R. J. A. Tough, *Sea Clutter: Scattering, the K Distribution and Radar Performance*, vol. 20, Stevenage, Herts., U.K.: IET, 2006.
[5] T. M. Elfouhaily and C. A. Guérin, "A critical survey of approximate scattering wave theories from random rough surfaces," *Waves Random Media*, vol. 14, no. 4, pp. 37–41, 2004.
[6] D. Li, Z. Zhao, C. Qi, Y. Huang, Y. Zhao, and Z. Nie, "An improved two-scale model for electromagnetic backscattering from sea surface," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 953–957, Jun. 2020.
[7] A. Awada, M. Y. Ayari, A. Khenchaf, and A. Coatanhay, "Bistatic scattering from an anisotropic sea surface: Numerical comparison between the first-order SSA and the TSM models," *Waves Random Complex Media*, vol. 16, no. 3, pp. 383–394, 2006.
[8] H. Li and J. T. Johnson, "On the amplitude distributions of bistatic scattered fields from rough surfaces," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 6883–6892, Dec. 2017.
[9] M. Zhang, H. Chen, and H. Yin, "Facet-Based investigation on EM scattering from electrically large sea surface with two-scale profiles: Theoretical model," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 1967–1975, Jun. 2011.
[10] B. C. Armstrong and H. D. Griffiths, "Modelling spatially correlated K-distributed clutter," *Electron. Lett.*, vol. 27, no. 15, pp. 1355–1356, Jul. 1991.
[11] L. Rosenberg, S. Watts, and M. S. Greco, "Modeling the statistics of microwave radar sea clutter," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 10, pp. 44–75, Oct. 2019.
[12] P. B. Wei, M. Zhang, D. Nie, and Y. C. Jiao, "Statistical realisation of CWMFSM for scattering simulation of space-time varying sea surface," *Int. J. Remote Sens.*, vol. 40, no. 1, pp. 332–345, 2019.
[13] L. Dejean, D. Pastor, J. M. Quellec, M. Chabah, and N. Bon, "The clutter SIRP and Gaussian models: A brief overview and a comparison," in *Proc. IET Seminar Radar Clutter Model.*, 2008, pp. 41–47.
[14] C. Bouvier, L. Martinet, G. Favier, and M. Artaud, "Simulation of radar sea clutter using autoregressive modelling and K-distribution," in *Proc. Int. Radar Conf.*, 1995, pp. 425–430.
[15] D. Huang, D. Z. Zeng, T. Long, and J. Y. Yu, "Design of a correlated log-normal distributed sequence generator based on Virtex-IV series FPGA," in *Proc. Int. Conf. Comput. Appl. Syst. Model.*, 2010, pp. V2–340–V2–343.
[16] T. Pardhu and N. A. Reddy, "FPGA design of clutter generator for radar testing," *Int. J.*, vol. 2, no. 1, pp. 13–24, 2015.
[17] J. R. van der Merwe, J. J. Strydom, and J. E. Cilliers, "Realtime generation of K-Distributed sea clutter for hardware in the loop radar evaluation," in *Proc. Eur. Radar Conf.*, 2016, pp. 17–20.
[18] D. Huang, J. Yu, and G. Li, "Two-dimensional temporal and spatial correlated lognormal clutter sequence generated based on FPGA," in *Proc. 2nd IEEE Int. Conf. Comput. Commun.*, 2016, pp. 2835–2839.
[19] G. Xu, F. Hao, J. Chen, and C. Xiu, "Optimization of clutter simulation based on GPU," *IEEE Access*, vol. 8, pp. 29501–29507, 2020.

[20] J. Cong, Z. Fang, M. Lo, H. Wang, J. Xu, and S. Zhang, "Understanding performance differences of FPGAs and GPUs," in *Proc. IEEE 26th Annu. Int. Symp. Field-Programmable Custom Comput. Mach.*, 2018, pp. 93–96.
[21] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, "High-level synthesis for FPGAs: From prototyping to deployment," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 4, pp. 473–491, Apr. 2011.
[22] G. Martin and G. Smith, "High-level synthesis: Past, present, and future," *IEEE Des. Test Comput.*, vol. 26, no. 4, pp. 18–25, Jul./Aug. 2009.
[23] SDSoC Environment User Guide (UG1027). Accessed: Apr. 2020. [Online] Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_4/ug1027-sdsoc-user-guide.pdf
[24] C. Sekar and Hemasunder, "Tutorial T7: Designing with xilinx SDSoC," in *Proc. 30th Int. Conf. VLSI Des. 16th Int. Conf. Embedded Syst.*, 2017, pp. 11–12.
[25] P. B. Wei, M. Zhang, D. Nie, and Y. C. Jiao, "Statistical realisation of CWMFSM for scattering simulation of space-time varying sea surface," *Int. J. Remote Sens.*, vol. 40, no. 1, pp. 332–345, 2019.
[26] L. Klein and C. Swift, "An improved model for the dielectric constant of sea water at microwave frequencies," *IEEE Trans. Antennas Propag.*, vol. 25, no. 1, pp. 104–111, Jan. 1977.
[27] L. Dong, M. Wang, J. He, Z. Li, L. He, and W. Wang, "The research for effects of window functions in radio astronomy," in *Proc. 3rd Int. Congr. Image Signal Process.*, 2010, pp. 3064–3073.
[28] Z. Liu, W. Liu, and Y. Peng, "Random function based spectral representation of stationary and non-stationary stochastic processes," *Probabilistic Eng. Mech.*, vol. 45, pp. 115–126, 2016.
[29] Y. Ma, Y. Cao, S. Vrudhula, and J. Seo, "Optimizing the convolution operation to accelerate deep neural networks on FPGA," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 26, no. 7, pp. 1354–1367, Jul. 2018.
[30] Vivado Design Suite User Guide - High-Level Synthesis (UG902), 2017. [Online]. Available: www.xilinx.com
[31] R. Chen, N. Park, and V. K. Prasanna, "High throughput energy efficient parallel FFT architecture on FPGAs," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2013, pp. 1–6.
[32] R. Kastner, J. Matai, and S. Neuendorffer, "Parallel programming for FPGAs," 2018, *arXiv:1805.03648*.

Shuhao Zhang received the B.Sc. degree in electromagnetic wave propagation and antenna, in 2017 from Xidian University, Xi'an, China, where he is currently working toward the Ph.D. degree in radio physics at the School of Physics and Optoelectronic Engineering.

His research interests include sea clutter modeling and simulation, and field programmable gate arrays hardware acceleration.

Jinxing Li was born in Henan, China, in 1993. He received the B.S. degree in EM wave propagation and antenna and the Ph.D. degree in radio physics from Xidian University, Xi'an, China, in 2014 and 2019, respectively.

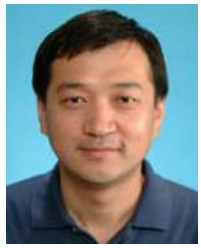
He is currently the University Lecture with the School of Physics and Optoelectronic Engineering, Xidian University, Xi'an, China. His current research interests include computational electromagnetics, EM scattering modeling of complex targets, and sea surfaces and its application.

Yachao Li was born in Jiangxi Province, China in May, 1981. He received the M.S and Ph.D. degrees in 2005 and 2008, respectively, from Xidian University, both in electrical engineering.

He is currently a professor with Xidian University.His current research interests include SAR/ISAR imaging, missile-borne SAR imaging, ground moving target indication, matching and orientation of SAR image, real-time signal processing based on FPGA and DSP technology, and distributed radar.

Pengbo Wei was born in Shaanxi, China, in 1988. He received the B.S. degree in applied physics and Ph.D. degree in radio physics from Xidian University, Xi'an, China, in 2011 and 2016, respectively.

From 2016 to 2019, he was a Postdoctoral Research Fellow in EM field and microwave technique with Xidian University, where he is currently a University Lecturer with the School of Physics and Optoelectronic Engineering. His current research interests include computational electromagnetics, EM theory and its application, theoretical modeling on EM scattering from rough (ground) surface, and terrain and sea clutter analysis and simulation.



Min Zhang (Member, IEEE) received the B.Sc. degree in physics from Shaanxi Normal University, Xi'an, China, in 1990, the M.Sc. degree in radio physics from Xidian University, Xi'an, China, in 1998, and the Ph.D. degree in astrometry and celestial mechanics from Shaanxi Astronomical Observatory, Chinese Academy of Sciences, Xi'an, China, in 2001.

From 2001 to 2003, he was a Postdoctoral Research Fellow in EM field and microwave technique with Xidian University, where he is currently a Full Professor with the School of Physics and Optoelectronic Engineering. From May 2003 to October 2004, he was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His current research interests include EM model design, integral equation techniques, and fast hybrid algorithms for EM scattering, radiation, wave propagation, radar signal processing and SAR imaging.