

METHODS

An Adaptive Word-Length Selection Method to Optimize Hardware Resources for FPGA-Based Real-Time Simulation of Power Converters

FUHAI ZHAO¹, JIANG DU¹, YUNKAI DENG¹, (Member, IEEE),

JIALIN ZHENG², (Student Member, IEEE),

YANGBIN ZENG², (Member, IEEE), AND CHUNHUI QU¹

¹Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China

²Department of Electrical Engineering, Tsinghua University, Beijing 100084, China

Corresponding author: Chunhui Qu (quch@aircas.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grant 52207209, and funded by China Postdoctoral Science Foundation under Grant 2021M701844.

ABSTRACT Power converters have become increasingly important in the power industry. To enhance design reliability and efficiency, real-time simulation is a popular method for testing control units of these devices. FPGA-based real-time simulation is considered a genuine alternative to CPU-based applications as control unit frequencies increase. Although automated tools can map simulation methods from floating-point high-level languages to fixed-point FPGA, manually selecting the fixed-point word length to meet precision requirements involves a significant amount of tedious and repetitive work. Word length selection also impacts FPGA hardware resource usage. This paper proposes an adaptive word length selection method under accuracy constraints to minimize hardware resources. The innovation lies in using error noise models to compute analytical expressions relating simulation model accuracy to word length. The adaptive word length selection process is combined with commercial high-level synthesis tools to achieve high resource optimization performance. A three-phase inverter example demonstrates the benefits of our method in calculating model accuracy and optimizing FPGA hardware resources in simulations.

INDEX TERMS Real-time simulation, field-programmable gate arrays resource optimization, power converter, word-length selection.

I. INTRODUCTION

Power converters are revolutionizing the entire electrical industry, encompassing renewable energy generation [1], high-voltage direct current transmission [2], hybrid AC/DC distribution grids [3], and electric vehicles, among others. Such widespread applications inevitably demand stricter requirements for research and development cycles and operational reliability [4]. Real-time simulation is widely recognized as an economically efficient prototyping method, used in power converter controller testing [5]. This design technique involves implementing the power part of the converter system in digital computing hardware to simulate the actual controller's response throughout the closed-loop system [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos¹.

In recent years, thanks to the development of power semi-conductors, the switching frequency of power converters has become increasingly higher [7]. Consequently, real-time simulation of these systems requires sub-microsecond time steps to obtain gate signals and output precise results accurately [8]. Such small-time steps have prompted a shift in real-time simulation from CPU-based hardware to FPGA-based implementation, as it offers ultra-low transmission latency and high-speed computing capabilities [9].

However, FPGA programming poses a challenging task due to the necessity of designing at the Register Transfer Level (RTL) abstraction, which demands expertise in hardware [10]. Employing RTL's low abstraction level directly for developing FPGA-based real-time simulators considerably increases development cost and time, limiting its applications [11]. To address this issue, FPGA manufacturers

have developed high-level synthesis (HLS) tools, such as Xilinx's Vitis HLS [12], to facilitate FPGA programming with high-level languages. HLS tools offer various optimization directions to better control pipelining, data partitioning, and inlining, thus enhancing hardware performance. Furthermore, efficient simulation algorithms on FPGAs require considering data representation [13]. Given the considerations of computation speed and hardware resource utilization, fixed-point formats, when compared to high-precision floating-point formats, present notable advantages. This makes them a more suitable choice for FPGA implementations due to their efficiency and lower complexity. Currently, mainstream FPGAs utilize fixed-point arithmetic, wherein the data range and computational accuracy are significantly influenced by the word length. A judicious choice of word length is pivotal too short leads to increased simulation errors, too long decreases simulation speed and consumes excess hardware resources. Hence, selecting an appropriate fixed-point format is crucial for the performance of FPGA-based real-time simulators [14].

The fixed-point format selection process involves choosing word length and binary point position for all data, with the primary criteria being the achievable precision and required hardware resources [15]. Shorter word lengths may result in precision loss due to the inability to accurately represent variables, while longer word lengths approach floating-point precision but demand more hardware resources and computation time. To strike a suitable balance between simulation accuracy and resource consumption, it's necessary to choose an optimal word length that ensures a certain degree of precision. In practical engineering, word length selection often relies on empirical or trial-and-error methods, which lack theoretical grounding. The concept of signal-to-noise ratio (SNR) has been introduced to study the relationship between word length, simulation step size, and simulation accuracy, and is used to choose the optimal simulation step size [16]. However, it requires calculation across all predefined word length ranges and entails a large amount of quantitative computation for SNR.

This paper pioneers an automatic adjustment method for real-time power converter simulation to address these challenges. Leveraging a realistic noise model, we enable direct computation of precision metrics based on the simulation signal flow. We integrate this analytical precision assessment with the simulator's high-level synthesis process, facilitating automatic word length selection, accurate resource usage estimation, and performance optimization. Notably, our approach significantly reduces word length selection evaluation time, marking a considerable advancement in the field.

The paper is structured as follows: Section II introduces the real-time simulation architecture of power converters; Section III presents the precision assessment method; Section IV outlines the automatic optimization process combining analytical precision assessment and high-level synthesis; Section V demonstrates the advantages of the

proposed method through case studies; and finally, Section VI concludes the paper.

II. REAL-TIME SIMULATION OF POWER CONVERTERS

A. MODELING OF POWER CONVERTER

Several methods have been developed for modeling power converters, including switch-based models such as the dual resistor model, constant admittance model, and switch function model. These approaches differ in their mathematical descriptions and ability to reproduce switching behavior. Typically, a power converter system needs to be transformed into a matrix form for numerical computation. Two prevalent techniques for generating matrices are state-space and nodal analysis methods. This paper adopts the Modified Nodal Analysis (MNA) method due to its ease of automation. Utilizing the MNA approach, power converters can be represented as,

$$\begin{bmatrix} Y & N_1 \\ N_2 & Z \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} = \begin{bmatrix} i_{src} \\ v_{src} \end{bmatrix} \quad (1)$$

where v and i denote the system nodal voltages and branch currents, while Y and Z correspond to their respective admittance and impedance. N_1 and N_2 are coupling matrices used to establish the relevant nodal or branch equations. i_{src} and v_{src} represent the current sources and voltage sources associated with the respective nodes.

Inductances and capacitances of the converter need to be discretized to avoid time-dependent impedance matrices. Common discretization methods include the Euler method, trapezoidal rule, etc. Using the trapezoidal rule as an example, the differential equation for inductance can be discretized into the following difference equation,

$$\begin{aligned} i_{L,k+1} &= \frac{T}{2L} v_{L,k+1} + i_{L,k} + \frac{T}{2L} v_{L,k} \\ &= G_L v_{L,k+1} + I_{L,k} \end{aligned} \quad (2)$$

where the subscript k denotes the k -th time step. The difference equation for inductance is composed of an admittance G_L and a companion current source. Similarly, the difference equation for capacitance can be represented as an admittance G_C and a companion current source. After discretization, the equivalent models for capacitance and inductance can be depicted as shown in Fig. 1.

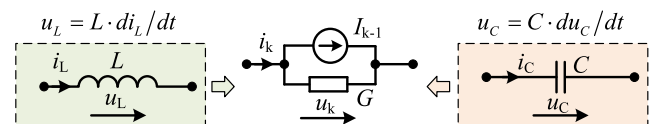


FIGURE 1. Discrete equivalent circuits for capacitors and inductors.

By modeling all energy storage elements in the circuit as admittances and companion current sources using the aforementioned methods, a discretized linear circuit can be obtained, as shown in Eq. (2). Furthermore, to represent switching behavior, this paper adopts a switch function modeling, which models the switch as an equivalent source

determined by the switching coefficient, as shown in Fig. 2. The companion current sources resulting from the discretization of energy storage elements, along with the switch equivalent sources and the original sources, together constitute the sources on the right side of Eq. (2).

$$\begin{cases} v_E(t_n) = k_E v_J(t_{n-1}) \\ i_J(t_n) = k_J i_E(t_{n-1}) \end{cases} \quad (3)$$

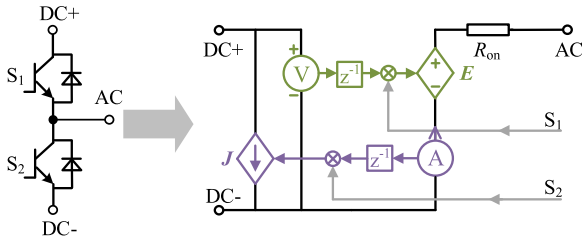


FIGURE 2. Switch leg modeling and equivalent circuit.

B. SOLVING FLOW OF FPGA-BASED SOLVER

A corresponding FPGA-based simulator has been designed, utilizing the power converter circuit modeling methods previously mentioned. Detailed information regarding its simulation algorithm and hardware implementation can be found in the referenced literature [17], [18]. Fig. 3 illustrates the solution process of the simulator, which is comprised of five basic function modules.

1) SWITCH STATE UPDATER

Composed of two parts of hardware logic, it is responsible for updating the switch equivalent source based on gate signals and dynamically updating the vector register of state variables.

2) MATRIX PARAMETER GENERATOR

Generates matrices and source vectors representing the current network equations based on the information from the updater and other fixed data. Matrices are stored in the FPGA's block RAM, while source vectors are saved in registers.

3) MATRIX-VECTOR MULTIPLIER

Performs multiplication of the generated matrix and source vector to compute the new values of state variables. Generally, the FPGA's parallel hardware is utilized to divide the

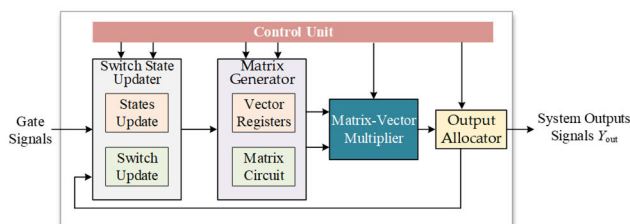


FIGURE 3. Flow of FPGA-based solvers.

matrix-vector multiplication into multiple dot-product operators for parallel computation, reducing calculation latency.

4) OUTPUT ALLOCATOR

Computed results are allocated to different addresses according to design requirements. Historical values are assigned to the switch updater, while values that need to be observed serve as the simulator's output.

5) CONTROL UNIT

Responsible for the simulator's initialization and normal operation, it coordinates the operation of the above four functional units and memory read/write operations through finite state machines and clocks.

C. DATA SCALING

The inputs and outputs of the FPGA-based solver need to be connected to A/D (Analog-to-Digital) and D/A (Digital-to-Analog) converters to form a closed-loop with the actual controller, as shown in Fig. 4. Consequently, it is necessary to scale the solver's inputs and outputs to the range supported by the A/D and D/A converters.

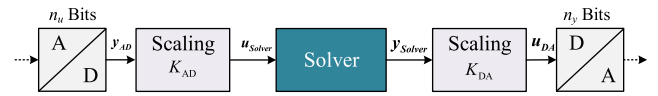


FIGURE 4. Data scaling of solver input and output.

In power converter applications, the solver's inputs are generally gate signals with small value variations, while the outputs comprise various variables with large value variations. Considering the output resolution of the A/D and D/A converters, it is necessary to scale down the output results of the A/D converter to reduce hardware resource usage and scale down the output results of the solver to fit the capabilities of the D/A converter. The overall data scaling method is as follows,

$$\begin{aligned} u_{\text{Solver}} &= K_{\text{AD}} \cdot y_{\text{AD}}, \\ u_{\text{DA}} &= K_{\text{DA}} \cdot y_{\text{Solver}}. \end{aligned} \quad (4)$$

where K_{AD} and K_{DA} are diagonal matrices composed of the corresponding input and output scaling factors, respectively. The scaling factors can be obtained through the following methods,

$$\begin{aligned} K_{\text{AD},i} &= \frac{\max(|u_{\text{Solver,max}}|, |u_{\text{Solver,min}}|)}{2^{n_{\text{AD}}-1} - 1}, \\ K_{\text{DA},i} &= \frac{\max(|y_{\text{Solver,max}}|, |y_{\text{Solver,min}}|)}{2^{n_{\text{DA}}-1} - 1}. \end{aligned} \quad (5)$$

where i denotes the index of the corresponding element in the vector. n_{AD} and n_{DA} represent the output bit counts of the A/D and D/A converters, respectively.

III. SNR-BASED PRECISION ASSESSMENT METHOD

To minimize computational latency and hardware resource consumption, fixed-point implementation is commonly recommended for FPGA-based simulation methods. When selecting the fixed-point word length, it is essential to ensure that the implemented precision meets the required accuracy constraints. SNR serves as a widely used standard for evaluating fixed-point precision. However, obtaining SNR through simulation methods can be time-consuming. This paper proposes an analytical precision assessment method based on SNR, specifically tailored for power converter simulations, to enhance evaluation efficiency.

A. NOISE MODEL

When quantizing data using fixed-point format, quantization errors are inevitably introduced. A quantized value $\hat{x}(n)$ can be represented by the true value $x(n)$ before quantization and the quantization error $b(n)$, as follows:

$$\hat{x}(n) = x(n) - b(n) \quad (6)$$

This quantization error can be equivalently modeled as a uniform distribution of white noise that is independent of the signal $\hat{x}(n)$, along with other noise components. Thus, the impact of quantization can be described through a statistical model of the quantization error. This model has been proven effective in evaluating quantization errors during data format conversions.

B. ANALYTIC NOISE POWER EXPRESSION

To analyze the impact of noise models, we consider a linear time-invariant (LTI) system with n inputs $x_i(n)$ and one output $y(n)$. In this system, the input $x_i(n)$ and output $y(n)$ are transformed into $X_i(z)$ and $Y(z)$ through the z-transform. We define $H_i(z)$ as the transfer function between $Y(z)$ and $X_i(z)$. By applying the inverse z-transform, we obtain the corresponding impulse response $h_i(n)$. Consequently, the output $y(n)$ can be represented as follows,

$$y(n) = \sum_{i=0}^{k-1} h_i(n) * x_i(n) \quad (7)$$

When considering a fixed-point representation of the system, we take $x_i(n)$ as the input to the LTI system. Employing fixed-point methods results in actual output $\hat{y}(n)$, with its difference from the ideal output $y(n)$ defined as the output computational error $b_y(n)$. According to the noise model, this computational error primarily stems from two error sources. The first source of error is input propagation noise $e_{x,i}(n)$, which originates from the propagation of input noise $u_{x,i}(n)$ associated with the fixed-point input $\hat{x}_i(n)$ in the system. The second source of error is constant propagation noise $e_{c,i}(n)$, caused by the quantization of internal system constants. The propagation function between $e_{c,i}(n)$ and input signal $x_i(n)$ is $\Delta H_i(z)$, corresponds to the difference between $H_i(z)$ and $\hat{H}_i(z)$. To simplify the expression, we assume that all x_i are

white noise. In this case, the output noise can be represented as shown in Fig. 5.

$$e_y = \sum_{i=0}^{k-1} \Delta h_i * x_i + \sum_{i=0}^{k-1} h_i * e_{x_i}^* \quad (8)$$

Indeed, the output noise by represents an LTI system excited by multiple individual input noise sources. To simplify, $e_{y,i}$ is used to denote a single output noise, u_i corresponds to its input. Consequently, the power associated with the variance of output noise $e_{y,i}$ can be expressed through statistical parameters,

$$E(e_{y,i}^2) = \left(\mu(u_i)H_i(e^{j0}) \right)^2 + \sigma^2(u_i) \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} |H_i(e^{j\Omega})|^2 d\Omega \quad (9)$$

Furthermore, the power of the output noise by can be calculated by aggregating these individual output noise contributions,

$$E(e_y^2) = \sum_{i=0}^{K-1} E(e_{y,i}^2) + \sum_{a=0}^{K-1} \sum_{b=0}^{K-1} \varphi(a, b) \quad (10)$$

where $\varphi(a, b)$ equals the product of the expectations of $e_{y,a}$ and $e_{y,b}$, as the quantization noise sources are uncorrelated. Eqs. (10) and (10) demonstrate that the power of the output quantization noise depends on the expectations and variances of the quantization noise sources and the system inputs, as well as the transfer function of the entire system.

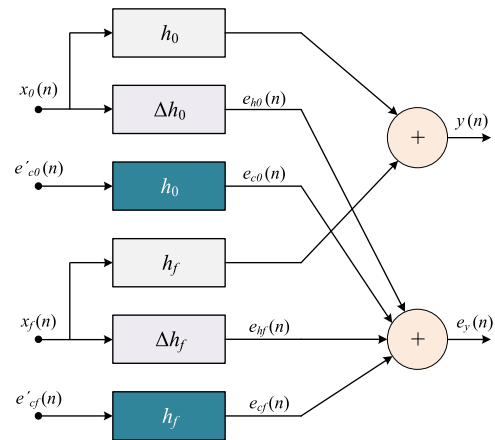


FIGURE 5. Noise model.

C. SNR CALCULATION METHOD

Utilizing the analytic noise power expressions, the SNR of real-time simulator outputs is calculated to evaluate accuracy. The specific steps are as follows: firstly, the simulator is divided into five sections based on functionality, with details provided in Section II-B. Once the parameters for each functional block have been determined, designated fixed-point formats are assigned as inputs for different blocks, allowing optimal word length settings to be established. Subsequently,

the transfer functions between all inputs and outputs are assessed. Lastly, the SNR of the simulator output is calculated using the frequency response of different transfer functions and the statistical parameters of quantization noise.

Specifically, the analysis of signal flow graphs and required operations for different functional blocks is performed initially, based on the fixed-point signal word length, signal type, constant values, etc. Subsequently, the two noise sources defined in Section III-B are detected and added to the signal flow graphs. The statistical parameters of these noise sources can be derived from ideal noise sources or collected from the real world. Furthermore, to represent data and operators at the noise level, each data node in the signal flow graph is divided into signal nodes and noise nodes, while each operator is replaced with a modified operator featuring noise propagation. The data conversion process in the signal flow is illustrated in Fig. 6, and the modified multiplication and addition operators are shown in Fig. 7. S and N represent the ideal signal and noise, respectively.

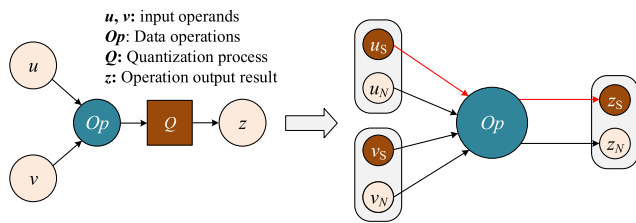


FIGURE 6. Data conversion process.

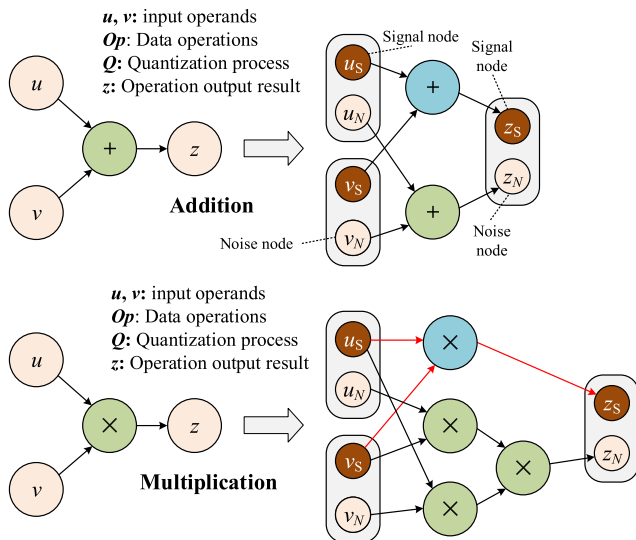


FIGURE 7. Modified multiplication and addition operators.

The modified signal flow graph is further employed to determine transfer functions. Firstly, traversing the signal flow graph from input to output is necessary to obtain the linear function of each signal flow, and the corresponding transfer functions are computed through Z-transform. If there are cyclic structures within the signal flow graph, a graph

decomposition method proposed in [18] is used to convert the cyclic graph into a directed acyclic graph.

After obtaining the transfer functions of different functional blocks, the frequency responses of various blocks can be easily acquired. Finally, the power of the output noise signal is calculated using Eqs. (10) and (12), and compared with the desired signal-to-noise ratio to evaluate whether the simulation output accuracy meets the requirements.

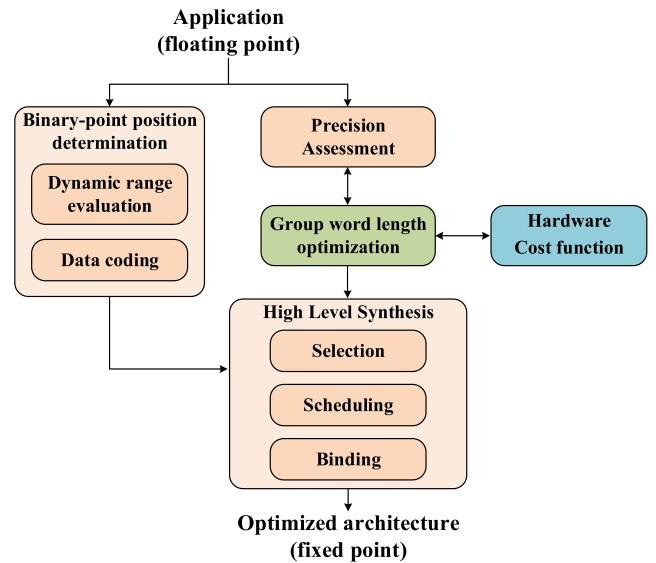


FIGURE 8. Multi-word-length optimization architecture.

IV. WORD-LENGTH OPTIMIZATION WITH PRECISION CONSTRAINT

A. FLOATING-POINT TO FIXED-POINT CONVERSION PROCESS

Fig. 8 illustrates the proposed automatic conversion process of floating-point algorithms to fixed-point algorithms. This method relies on the proposed analytical accuracy assessment technique and high-level synthesis (HLS) tools. The proposed approach consists of two steps. Firstly, determine the binary point position for all fixed-point data to ensure that the integer part of each fixed-point number does not overflow. Assess the variation range of each data and define the binary point position accordingly. Then, perform multi-word-length optimization with the goal of minimizing hardware resource usage while satisfying accuracy constraints. Analytical SNR-based assessment methods can be used for accuracy evaluation, and hardware resources can be obtained through HLS tools.

B. BINARY POINT POSITION DETERMINATION

To prevent overflow, or to maintain a low probability of overflow, after converting data to a fixed-point format, it is necessary to determine the binary point position (BPP) for each data. To ensure the appropriateness of the found position, first, evaluate the data range of each variable, using floating-point simulator calculations as a reference. For input

data, the dynamic range can be directly determined, while for intermediate and output data, the dynamic range should be determined in conjunction with the input data's dynamic range and specified operators. Next, based on the determined data dynamic range, determine the minimum length of each fixed-point number's integer part. With this foundation, considering fixed-point computation rules, properly shift the integer bit count to adapt the data format to different operation formats.

C. MULTI-WORD-LENGTH OPTIMIZATION ARCHITECTURE

After determining the integer word length for each fixed-point data, the next step is to determine the fractional word length. The primary goal of data format conversion is to meet accuracy constraints, followed by minimizing hardware resource usage. Therefore, the proposed multi-word-length optimization architecture is achieved through iterative accuracy assessment and resource synthesis. This architecture assigns dedicated word lengths for each functional block in real-time simulation, and if time permits, even for each computational step. Then, use the proposed analytical output noise power calculation method to evaluate accuracy. Finally, if the output noise power is below the defined threshold, the multi-word-length optimization problem becomes a chip resource minimization problem.

$$\min_{e_k \in \mathbb{Z}^+} (S(e_k)) \text{ such as } SNR(e_k) = 10 \log_{10} (P_y/P_{e_y}) \leq SNR_{min} \quad (11)$$

For hardware resource evaluation, this paper adopts commercial HLS tools to obtain accurate hardware resource usage information. A simplified chip area evaluation model can also be used to further improve optimization time. The hardware area of each adder and multiplier in the simulator is,

$$S_{add} = K_{add} \cdot e_k, \\ S_{mult} = K_{mult} \cdot (e_k)^2, \quad (12)$$

This hardware resource minimization problem can be solved through constrained nonlinear minimization methods, where $e_k \in \mathbb{N}^+$. In our research, we specifically used the Sequential Quadratic Programming (SQP) method to address this issue. Moreover, we used scripts to call the HLS tool for synthesis, returning the achieved SNR to SQP to get the word length format for the next iteration. This process is repeated until the accuracy requirement is met. Moreover, combined optimization algorithms can be employed to find the global optimal solution in the discrete search space while avoiding the inefficiency of exhaustive searches.

V. CASE STUDY: THREE-PHASE TWO-LEVEL INVERTERS

A. STUDY SETUP

In the following, we present our proposed method using a three-phase two-level inverter, as depicted in Fig. 9, which is widely employed in renewable energy systems, electric vehicles, and uninterruptible power supplies. Studying real-time

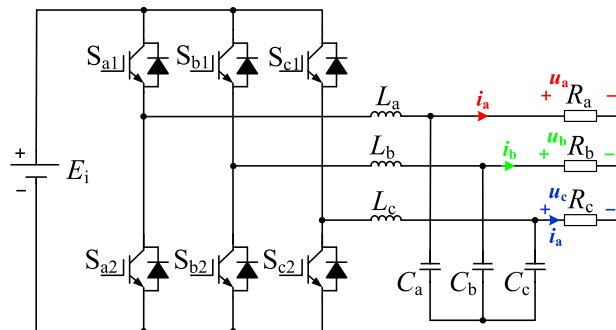


FIGURE 9. Three-phase two-level inverter topology.

TABLE 1. System parameters of studied case.

Parameters	Value
V_{DC}/V	700
f_{sw}/kHz	5
L/H	1e-6
C/F	680e-6
R/Ω	39
R_{on}/Ω	1e-3

simulation of such inverters contributes to enhancing design efficiency and reliability in these practical applications. System parameters are provided in Table 1.

Initially, the inverter is modeled based on the switch model described in Section II, and the matrix generator is formulated using the trapezoidal discretization method. Detailed procedures and underlying algorithms for these processes can be found in the previous works [17], [18]. This generator can obtain the matrix form Eq. (1) automatically according to the inverter's connectivity, alleviating manual generation efforts.

A personal computer with an Intel® i7-10700 CPU is utilized for conducting offline simulation tests and FPGA word-length optimization selection, as well as high-level synthesis. The word-length optimization method is implemented in C++, and Xilinx® HLS is employed as a resource assessment tool. The synthesized RTL is then downloaded to an FPGA evaluation board using the Vivado® tool for real-time simulation. The FPGA evaluation board employed is the Xilinx® VC707, equipped with a Virtex®-7 485T chip. This board is accompanied by TI®'s DAC and ADC boards for signal input and output.

B. ACCURACY VALIDATION

Subsequently, floating-point and fixed-point simulations are performed in the MATLAB® environment. The simulation scenario is designed to assess the response capacity of inverter controller parameters during a sudden voltage drop. In the interval from 0 to 0.25s, the DC bus of the inverter maintains a rated voltage of 700V. However, at the 0.25s mark, the voltage abruptly drops to 600V due to the failure of other sources and loads connected to the DC bus. The inverter adopts the classical Proportional-Resonant (PR) control. Regarding data format, the fixed-point simulation uses

TABLE 2. Hardware resources optimization with unified word-length.

Integer word-length	Fractional word-length	SNR (dB)	LUTs	DSP48	Flip-Flops
24	18	88.3	12939	345	11588
24	20	92.7	14185	376	13544
24	22	96.1	15530	427	14975
24	24	100.5	16988	468	16024
24	26	102.2	18337	534	17685
24	28	103.5	19910	597	18415

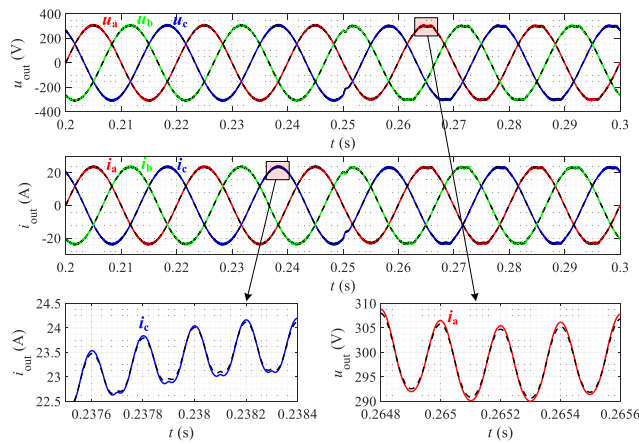


FIGURE 10. Simulation results.

a 128-bit word length, double that of the 64-bit floating-point simulation. Despite having a larger word length, the fixed-point simulation doesn't provide the same precision due to its inability to dynamically adjust the decimal point. The noticeable difference between the two results in Fig. 10 results from these format differences, not from injected noise. Floating-point operations, while more precise, are slower on an FPGA, necessitating a trade-off between precision and computation speed in real-time simulations. Both simulations employed a step size of 100 ns, as depicted in Fig. 10. In this context, u_a , u_b , and u_c represent the three-phase voltages on loads, while i_a , i_b , and i_c denote the three-phase currents on loads. The definitions of phases a , b , and c are referenced in Fig. 9. Following this voltage drop, PR control aids the inverter in quickly overcoming the DC bus voltage drop. However, as the PR control parameters are designed according to the rated conditions, slight distortion occurs under undervoltage circumstances. It can be observed that the simulation outcomes are highly similar for both load voltage and current. Even the ripple details of the capacitor voltage exhibit a high degree of consistency. However, achieving such high precision requires twice the computational word length, which results in substantial resource usage and is not conducive to scaling up the simulation size.

C. WORD-LENGTH SELECTION OPTIMISATION AND RESOURCE COMPARISON

First, the word-length optimization is considered under the condition of a globally uniform word length, with the objective of minimizing the required hardware resources. An accuracy constraint of an SNR of 100 dB is set for the purpose of this research, to demonstrate the potential of the proposed method in achieving high precision. However, it should be noted that an SNR of 100dB may not be universally applicable in all scenarios and could potentially lead to overengineering. It's important to note that this is a benchmark established for this study, and the target SNR can certainly be adjusted according to the specific requirements of different applications in practical settings. In this case, for each optimization step, the integer part of the word length is chosen to be the maximum value of all data variations, which is 24 in this example. The fractional part is determined based on whether it meets the SNR constraint, and then the optimal result is found according to the resource usage synthesized by HLS. Typical values for a uniform word length are shown in Table 2.

It can be observed that, with the integer word length ensuring no data overflow, the SNR increases as the fractional word length increases, implying enhanced precision. When the fractional length grows to 24, the SNR approaches stability. However, as the fractional word length increases, hardware resources such as LUTs, DSP48, and FFs also expand. The rapid growth of hardware resources can limit the simulation scale of the real-time simulator.

To manage the increasing demand on hardware resources and maintain the efficiency of our real-time simulator, we have taken a strategic approach, further optimized based on the simulation workflow and calculation characteristics, as detailed in Section III-C. The workflow of the simulator is divided into five computational parts, each with distinct data requirements. If we applied a uniform fixed-point number format across all these parts, it would have to comply with the most stringent data format, leading to unnecessary resource consumption. In order to optimize hardware usage, we have assigned unique data formats to each of the five parts, allowing each part to use the format that best suits its requirements.

TABLE 3. Hardware resources optimization with multi-group word-length.

N_z	word-length					SNR (dB)	LUTs	DSP48	Flip-Flops
	b_1	b_2	b_3	b_4	b_5				
1	(24,26)	(24,26)	(24,26)	(24,26)	(24,26)	102.2	18337	534	17685
2	(14,14)	(24,26)	(24,26)	(24,26)	(24,26)	101.9	16745	495	17246
3	(14,14)	(16,20)	(16,20)	(26,28)	(16,20)	102.1	15859	476	17067
3	(14,14)	(14,20)	(14,20)	(24,26)	(24,26)	99.4	14985	462	16981
4	(14,12)	(16,20)	(14,20)	(26,28)	(16,20)	101.8	15048	468	16887
5	(14,12)	(16,20)	(14,20)	(26,28)	(14,20)	101.7	14963	462	16843
5	(14,12)	(16,20)	(14,20)	(26,28)	(14,16)	101.5	14825	448	16758

Despite the computations within each function being similar, further subdivision would not significantly conserve hardware resources and would necessitate more adaptive adjustment work. Therefore, we have allocated the five parts of the simulation to the b_1 - b_5 groups, each adopting a distinct data format. Necessary conversions are performed between groups with different data formats. The data format and conversion mode for each group can be managed using the adaptive word length optimization method proposed in this paper. This approach aids in striking a balance between precision and resource consumption, thus enhancing the overall efficiency of the real-time simulator.

The design space N_z is explored by gradually increasing the number of different variables “ b_k ” in Eq. 11, and the automatic word-length method is applied to each b_k , as previously mentioned. Table 3 presents the results of the multi-word length optimization.

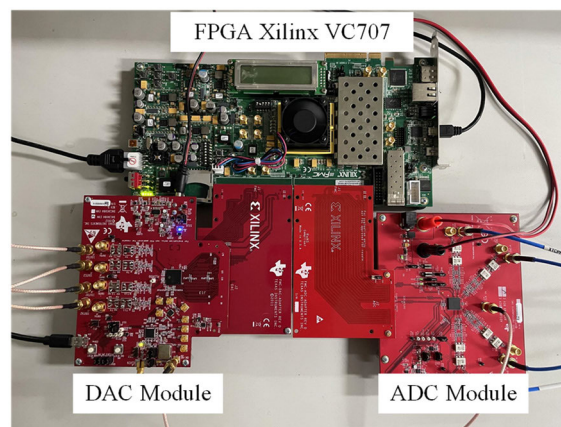
In the first experiment, all signals have the same word length. The third group of experiments provides the best accuracy, while the seventh group is optimal for hardware resource utilization. It can be observed that by increasing the design space, the word length required for each functional block can be more flexibly chosen, thereby maximizing hardware resource usage under the given accuracy constraint. With the automatic word-length selection method proposed in this paper, hardware resources can be reduced by up to 25.5% under the accuracy constraint. This implies that the simulation scale can be increased by 33% using this word-length selection method.

Building on this point, our design, as detailed in Table 3, utilized the Virtex-7 FPGA and its DSP blocks, which traditionally operate with 25×18 bit multipliers. However, we did not strictly adhere to this 25×18 limit. This deviation was influenced by three considerations: accuracy requirements, functionality demands, and design flexibility. Firstly, our application’s accuracy requirements necessitated wider data paths to maintain result precision. Secondly, complex mathematical operations in our design required larger bit widths. Finally, we aimed for design flexibility to accommodate date future upgrades without extensive redesign. Despite exceeding the 25×18 limit, the adaptive word-length selection method proposed in this paper allowed us to

optimize FPGA hardware resources while meeting the accuracy demands of real-time simulation applications. This approach demonstrates a practical application of our findings from the experiments, balancing resource utilization and simulation accuracy.

D. FPGA REAL-TIME SIMULATION RESULTS

The optimized word lengths are applied to different functional blocks of the real-time simulator, and the studied three-phase inverter is implemented on an FPGA, as shown in Fig. 11. The FPGA board is used for simulation computation, receiving external control signals through the ADC module and outputting simulation results via the DAC module.

**FIGURE 11.** FPGA hardware platform.

The simulation circuit continues to use the inverter shown in Fig. 9, which is in its nominal operating state at the start, with the controller using the classic PR control strategy. In addition, to display the results correctly on the DAC, it’s necessary to scale the simulation results. For instance, for voltage, the nominal value is 380V, while the DAC’s output capability is $\pm 2V$. Considering a certain margin, the scaling ratio is set as $2/400=1/200$. On the other hand, to display the correct results on the oscilloscope, this scaling ratio needs to be applied to the sampling results. The results after inverse scaling are shown in Fig. 12. It presents the waveform results

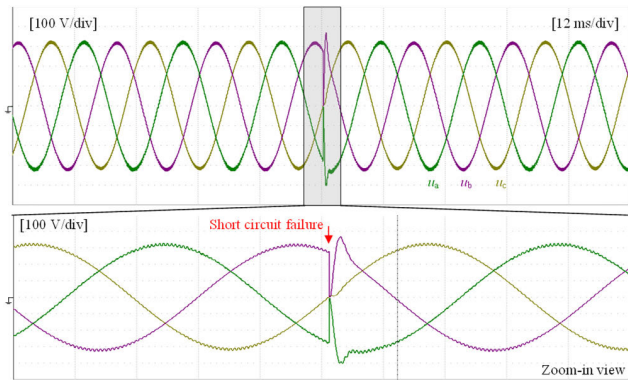


FIGURE 12. FPGA-based real-time HIL simulation results.

of the three-phase inverter output voltage and output current during a short-circuit fault test, where a sudden short circuit occurs at the input end at 0.25 seconds and is cleared after 0.5 ms. It can be observed that the inverter operates under such extreme power conditions, and these operating states can be used for control optimization in the later stage. This real-time simulation testing method can significantly mitigate the safety risks associated with power experiments under extreme conditions.

E. COMPREHENSIVE COMPARISON

In the sphere of real-time power electronic system simulations, the efficiency of word length selection methods is integral. This part spotlights a comparison between the proposed method and other prevalent methodologies.

1. The manual selection approach, the most common, involves manual adjustments based on simulation results. While simple, the method demands a significant time commitment, marking its primary disadvantage.
2. The operation-specific SNR computation method provides a more technical approach [7], [8], [9]. This method computes the SNR for each operation and employs a brute-force strategy to explore all potential results within a given range [19], [20]. Despite its ability to uncover optimal solutions within the range, this method is both time and resource-intensive, particularly as the scale of the case expands. Moreover, the manual creation of SNR expressions for each operation introduces potential errors and extends the time requirement.
3. The proposed method significantly improves upon these traditional approaches. Leveraging HLS tools, negates the need for manual SNR expression creation, reducing both time investment and the potential for error. Furthermore, it reframes the word length selection problem as a constrained resource optimization problem, with the SNR value acting as the constraint and the hardware resource as the variable.

This transformation enables the use of optimization algorithms, resulting in a more efficient resolution.

In summary, the proposed method in this paper offers substantial improvements over both manual and other automated techniques. It not only minimizes time and computational resources but also boosts the precision and reliability of the word length selection process.

F. DISCUSSIONS

1) FLEXIBILITY

The proposed method differs significantly from using commercial computational software like MATLAB's HDL Coder. While HDL Coder can convert Simulink models into hardware description languages like VHDL, the process involves a two-step conversion that can decrease efficiency. Furthermore, the HDL Coder performs redundant operations to ensure universal code operability, which can also reduce efficiency. In contrast, our method deploys customized simulation algorithms directly on FPGAs, offering flexible word length adjustment for each part of the simulation computation. This flexibility broadens the optimization space and allows us to utilize hardware resources more effectively.

2) EFFICIENCY

Meanwhile, the proposed method leverages existing commercial HLS tools, an approach that allows us to optimize computational efficiency and reduce complexity without having to develop new HLS tools from scratch. Compared to other FPGA methods [7], [8], [9], our approach offers the advantage of automatic word length selection, thereby eliminating the tedious task of manually selecting and evaluating word lengths. Furthermore, our method fundamentally differs from the approach in [21], which optimizes word length within HLS software but is not generally accessible. We externally optimize with existing HLS software, targeting the simulation process of power electronics by decomposing it into parts. Our technique optimizes five types of word lengths, offering faster results despite not having per operation selection. Given its operational readiness and practicality, our method presents a more viable alternative to [21] for real-time power electronics simulation.

3) AUTOMATION

Building on this, our method stands out from existing practices in its ability to automatically determine optimal word length within the fixed-point number format. Unlike traditional methods that require manual adjustments and in-depth involvement, our approach divides the simulation process into different word length groups, calculates their SNR based on theory, and finally adapts the word length based on the SQP algorithm. This innovative process not only enhances efficiency but also maintains precision without overusing computational resources. In contrast, traditional methods often struggle with rigid word length selection, either requiring exhaustive manual trials or settling for

single-precision floating-point numbers to avoid making word length choices [22], limiting their performance and increasing computation time. Our method effectively addresses these limitations, offering a more flexible, automated, and efficient solution for word length determination in FPGA-based real-time simulation.

4) COST-EFFECTIVENESS

Beyond technological merits, cost-effectiveness is key for our solution's viability. Our framework's creation and setup costs center on computational resources and time investment. Hardware for running simulations is required, but our tests confirm a personal computer suffices. We use Xilinx's HLS tools and specific mathematical libraries for optimization. Superior resources can hasten processes, but our method's strength lies in minimizing syntheses, enhancing efficiency, and automation. The setup requires substantial initial effort encompassing coding and integrating the optimization algorithm, but this is a one-time commitment. Once the framework is in place, it facilitates a streamlined approach to word length selection for real-time power electronic simulations.

Moreover, it is critical to note that typical HLS compilers can rearrange expressions and may optimize parts of the code. While these optimizations are beneficial in many respects, they can introduce discrepancies between the SNR calculated by our model and the actual SNR. Regardless, our model delivers a faster SNR approximation than direct computation through HLS, enabling efficient derivation of a general range for word lengths. To achieve more precise results within this range, we incorporate commercial HLS tools. By comparing the computational results of fixed-point and double-precision floating-point formats, we use the latter as a benchmark to calculate the SNR for all variables, yielding a more accurate assessment.

G. FUTURE WORK

In this study, we have adopted a strategy of grouping word lengths by functional blocks for streamlining optimization. Preliminary results indicate the efficacy of this approach. Yet, refining this technique through more specific, granular grouping might enhance optimization and promote efficient resource utilization.

Future investigations will experiment with more nuanced grouping strategies for word lengths. Potential approaches might include subdividing within functional blocks or implementing innovative grouping methods. These advancements could necessitate more proficient and efficient optimization algorithms, given the additional complexities introduced by more detailed grouping.

Therefore, our future work aims to design bespoke optimization algorithms, tailored specifically for word length optimization. These algorithms will account for the intricate dynamics of finer grouping, offering a more fine-tuned pathway toward efficient optimization.

VI. CONCLUSION

This paper proposes an adaptive word length selection method under accuracy constraints to optimize hardware resources. The novelty lies in using error noise models to derive analytical expressions that relate simulation model accuracy to word lengths. By combining the adaptive word length selection process with commercial high-level synthesis tools, high resource optimization performance is achieved. The benefits of this method are demonstrated through a three-phase inverter example, showcasing improved model accuracy and optimized hardware resource utilization in simulations. The proposed method has the potential to greatly reduce evaluation time and improve the efficiency of real-time simulations for power converter control units. By addressing the challenges associated with fixed-point format selection, this approach offers a promising solution to enhance the performance of FPGA-based real-time simulators in the rapidly evolving power industry.

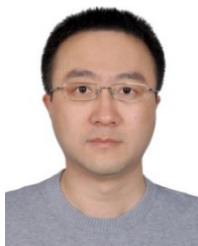
REFERENCES

- [1] O. Ellabban, H. Abu-Rub, and F. Blaabjerg, "Renewable energy resources: Current status, future prospects and their enabling technology," *Renew. Sustain. Energy Rev.*, vol. 39, pp. 748–764, Nov. 2014, doi: [10.1016/j.rser.2014.07.113](https://doi.org/10.1016/j.rser.2014.07.113).
- [2] C. Dufour, W. Li, X. Xiao, J.-N. Paquin, and J. Bélanger, "Fault studies of MMC-HVDC links using FPGA and CPU on a real-time simulator with iteration capability," in *Proc. 11th IEEE Int. Conf. Compat., Power Electron. Power Eng. (CPE-POWERENG)*, Apr. 2017, pp. 550–555, doi: [10.1109/CPE.2017.7915231](https://doi.org/10.1109/CPE.2017.7915231).
- [3] K. Ma, J. Wang, X. Cai, and F. Blaabjerg, "AC grid emulations for advanced testing of grid-connected converters—An overview," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 1626–1645, Feb. 2021, doi: [10.1109/TPEL.2020.3011176](https://doi.org/10.1109/TPEL.2020.3011176).
- [4] S. K. Mazumder et al., "A review of current research trends in power-electronic innovations in cyber-physical systems," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 5, pp. 5146–5163, Oct. 2021, doi: [10.1109/JESTPE.2021.3051876](https://doi.org/10.1109/JESTPE.2021.3051876).
- [5] H. Bai, C. Liu, E. Breaz, K. Al-Haddad, and F. Gao, "A review on the device-level real-time simulation of power electronic converters: Motivations for improving performance," *IEEE Ind. Electron. Mag.*, vol. 15, no. 1, pp. 12–27, Mar. 2021, doi: [10.1109/MIE.2020.2989834](https://doi.org/10.1109/MIE.2020.2989834).
- [6] Y. Zeng, J. Zheng, Z. Zhao, W. Liu, S. Ji, and H. Li, "Real-time digital mapped method for sensorless multi-timescale operation condition monitoring of power electronics systems," *IEEE Trans. Ind. Electron.*, early access, May 10, 2023, doi: [10.1109/TIE.2023.3273259](https://doi.org/10.1109/TIE.2023.3273259).
- [7] H. Chalanger, T. Ould-Bachir, K. Sheshyekani, and J. Mahseredjian, "A direct mapped method for accurate modeling and real-time simulation of high switching frequency resonant converters," *IEEE Trans. Ind. Electron.*, vol. 68, no. 7, pp. 6348–6357, Jul. 2021, doi: [10.1109/TIE.2020.2998746](https://doi.org/10.1109/TIE.2020.2998746).
- [8] M. Yushkova, A. Sanchez, and A. de Castro, "Oversampling techniques to improve the accuracy of hardware-in-the-loop switching models," *IEEE Trans. Power Electron.*, vol. 38, no. 5, pp. 6024–6035, May 2023, doi: [10.1109/TPEL.2023.3243702](https://doi.org/10.1109/TPEL.2023.3243702).
- [9] H. Chalanger, T. Ould-Bachir, K. Sheshyekani, and J. Mahseredjian, "Methods for the accurate real-time simulation of high-frequency power converters," *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9613–9623, Sep. 2022, doi: [10.1109/TIE.2021.3114706](https://doi.org/10.1109/TIE.2021.3114706).
- [10] W. Meeus, K. Van Beeck, T. Goedemé, J. Meel, and D. Stroobandt, "An overview of today's high-level synthesis tools," *Des. Autom. Embedded Syst.*, vol. 16, no. 3, pp. 31–51, Sep. 2012, doi: [10.1007/s10617-012-9096-8](https://doi.org/10.1007/s10617-012-9096-8).
- [11] F. Montano, T. Ould-Bachir, and J. P. David, "An evaluation of a high-level synthesis approach to the FPGA-based submicrosecond real-time simulation of power converters," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 636–644, Jan. 2018, doi: [10.1109/TIE.2017.2716880](https://doi.org/10.1109/TIE.2017.2716880).

- [12] *Vitis HLS*. Accessed: Jun. 28, 2023. [Online]. Available: <https://china.xilinx.com/products/design-tools/vitis/vitis-hls.html>
- [13] M. Milton and A. Benigni, "ORTIS solver codegen: C++ code generation tools for high performance, FPGA-based, real-time simulation of power electronic systems," *SoftwareX*, vol. 13, Jan. 2021, Art. no. 100660, doi: [10.1016/j.softx.2021.100660](https://doi.org/10.1016/j.softx.2021.100660).
- [14] J. Xu, K. Wang, P. Wu, Z. Li, Y. Liu, G. Li, and W. Zheng, "FPGA-based submicrosecond-level real-time simulation of solid-state transformer with a switching frequency of 50 kHz," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 4, pp. 4212–4224, Aug. 2021, doi: [10.1109/JESTPE.2020.3037233](https://doi.org/10.1109/JESTPE.2020.3037233).
- [15] M. S. Martínez-García, A. de Castro, A. Sanchez, and J. Garrido, "Word length selection method for HIL power converter models," *Int. J. Electr. Power Energy Syst.*, vol. 129, Jul. 2021, Art. no. 106721, doi: [10.1016/j.ijepes.2020.106721](https://doi.org/10.1016/j.ijepes.2020.106721).
- [16] X. Guo, J. Yuan, Y. Tang, and X. You, "Hardware in the loop real-time simulation for the associated discrete circuit modeling optimization method of power converters," *Energies*, vol. 11, no. 11, p. 3237, Nov. 2018, doi: [10.3390/en11113237](https://doi.org/10.3390/en11113237).
- [17] J. Zheng, Y. Zeng, Z. Zhao, W. Liu, H. Xu, and S. Ji, "A semi-implicit parallel leapfrog solver with half-step sampling technique for FPGA-based real-time HIL simulation of power converters," *IEEE Trans. Ind. Electron.*, vol. 71, no. 3, pp. 2454–2464, Mar. 2024, doi: [10.1109/TIE.2023.3265042](https://doi.org/10.1109/TIE.2023.3265042).
- [18] Z. Yu, Z. Zhao, B. Shi, Y. Zhu, and J. Ju, "An automated semi-symbolic state equation generation method for simulation of power electronic systems," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 3946–3956, Apr. 2021, doi: [10.1109/TPEL.2020.3025785](https://doi.org/10.1109/TPEL.2020.3025785).
- [19] J. Yuan, X. Guo, C. Wang, and X. You, "FPGA resource optimization method for hardware in the loop real-time simulation of power converters," in *Proc. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, Mar. 2019, pp. 2849–2854.
- [20] P. Lamo, G. A. Ruiz, F. J. Azcondo, A. Pigazo, and C. Brañas, "Impact of the noise on the emulated grid voltage signal in hardware-in-the-loop used in power converters," *Electronics*, vol. 12, no. 4, p. 787, Feb. 2023.
- [21] D. Menard and O. Sentieys, "Automatic evaluation of the accuracy of fixed-point algorithms," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, Mar. 2002, pp. 529–535, doi: [10.1109/DATE.2002.998351](https://doi.org/10.1109/DATE.2002.998351).
- [22] J. Zheng, Z. Zhao, Y. Zeng, B. Shi, and Z. Yu, "An event-driven real-time simulation for power electronics systems based on discrete hybrid time-step algorithm," *IEEE Trans. Ind. Electron.*, vol. 70, no. 5, pp. 4809–4819, May 2023, doi: [10.1109/TIE.2022.3187594](https://doi.org/10.1109/TIE.2022.3187594).



FUHAI ZHAO received the B.E. degree in electronic information science and technology from Shandong University, Shandong, China, in 2012, and the M.E. degree in information and communication engineering from the Beijing Institute of Technology, Beijing, China, in 2015. He is currently an Assistant Researcher with the Aerospace Information Research Institute, Chinese Academy of Sciences. His research interest includes high-speed digital circuit design.



JIANG DU received the B.E. degree in electronics and information engineering and the M.E. degree in information and communication engineering from Xidian University, Xi'an, China, in 2005 and 2008, respectively.

In 2008, he joined the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, where he was an Associate Research Fellow with the Department of Space Microwave Remote Sensing Systems. He has participated in

the design and development of several spaceborne SAR system projects. His research interests include digital signal processing (DSP) and high-speed digital design.



YUNKAI DENG (Member, IEEE) received the M.E. degree in electrical engineering from the Beijing Institute of Technology, Beijing, China, in 1993. He is a member of the Scientific Board. He was a recipient of several prizes, including the First and Second Class Rewards of the National Defense Science and Technology Progress, in 2007, the First Class Reward of the National Scientific and Technological Progress, in 2008, the Achievements of the Outstanding Award of the CAS, in 2009, and the First Class Reward of the Army Science and Technology Innovation, in 2016, for his outstanding contribution in SAR field.



JIALIN ZHENG (Student Member, IEEE) received the B.S. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical Engineering, Tsinghua University, Beijing. His research interest includes real-time simulation techniques.



YANGBIN ZENG (Member, IEEE) received the B.E. degree in building electrical and intelligent engineering from Xiangtan University, Xiangtan, China, in 2015, and the Ph.D. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2021. He is currently a Postdoctoral Researcher with the Department of Electrical Engineering, Tsinghua University, Beijing. His current research interest includes real-time simulation techniques.



CHUNHUI QU received the B.E. and M.E. degrees in information and communication engineering from the Xi'an University of Electronic Technology, Xi'an, China, in 2006 and 2009, respectively. He is currently an Associate Researcher with the Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests include high-speed digital circuit design, digital system design, and radar system design.

...