ЕЛЕКТРОНИКА (ELECTRONICS)

***

Publisher

Faculty of Electrical Engineering
University of Banja Luka
Patre 5, 78000 Banja Luka
Bosnia and Herzegovina

Phone: +387 51 211-820
Fax: +387 51 211-408
Email: info@etf.unibl.org
Web: www.etf.unibl.org

Publication frequency: Two issues a year
ISSN: 1450-5843 (Print), 2831-0128 (Online)
UDC: 621.38
Number of printed copies: 50

ELECTRONICS, VOL. 29, NO. 2, DECEMBER 2025

Editor’s Column

Mladen Knezic

> It is hard for an individual to properly assess his intellectual abilities. Even a fool considers himself smart.
>
> *Milutin Milanković, Memories*

Editorial Letter
DOI: 10.53314/ELS2529041K

WE are pleased to present this new issue of the journal featuring four compelling papers that highlight significant advancements across integrated circuit simulation, high-efficiency digital signal processing architectures, reliable memory design for healthcare devices, and machine learning applications in precision agriculture.

The first paper, entitled “Virtual Simulation of Integrated Circuits Combining AP with DE Algorithm,” authored by Peipei Hu, proposes a simulation method that combines the Affinity Propagation (AP) algorithm for circuit fault diagnosis with the Differential Evolution (DE) algorithm for circuit parameter optimization. The fusion of these algorithms aims to provide a more efficient and accurate virtual simulation method for Integrated Circuit (IC) design, helping to predict and solve problems early in the design phase, thus reducing manufacturing costs and shortening the research and development cycle. Experimental results confirm that the combined AP and DE approach achieved a precision of 94.26%, a recall rate of 93.41%, and a mean F1 value of 88.59%. Furthermore, the convergence speed was measured at 56.77 seconds, with a stability of 93.17%. The proposed fusion algorithm demonstrates better performance across these metrics when compared to several optimization methods, including standalone DE and AP, Genetic Algorithm (GA), Simulated Annealing Algorithm (SAA), Ant Colony Algorithm (ACA), Particle Swarm Optimization (PSO), and Artificial Neural Network (ANN).

The second contribution, “High-Performance and Resource-Efficient Squaring Architecture for FPGA Platforms,” by Burhan Khurshid, introduces a novel approximate squaring architecture based on the CORDIC algorithm. By operating the CORDIC algorithm in linear mode, the architecture is modified to emulate the squaring function using only simple shift and add operations. The core design is an unfolded (pipelined) feed-forward architecture, converting the sequential iterative structure to a parallel one suited for high-speed Digital Signal Processing (DSP) applications. A Pareto analysis conducted on the 8-bit CORDIC square architecture found that 8 stages are optimal, achieving an Error Rate (ER) of 5%, meaning 95% of the results are accurate. This proposed architecture demonstrated superior efficiency, reporting a 25% reduction in Power-Delay-Area Product (PDAP) on ASIC platforms and a 38% reduction in PDAP on FPGA platforms, compared to the next best existing design. The design also recorded the lowest Mean Error Distance (MED) and Normalized Mean Error Distance (NMED) among the various compared 8-bit square architectures.

Next, the paper “Variation Tolerant SRAM with Enhanced Stability for Wearable Healthcare Devices,” authored by M. Kavitha, S. Ramani, and P. K. Janani, addresses the crucial need for ultra-low power and robust memory cells in battery-powered wearable healthcare devices. The authors propose an asymmetrical nine-transistor Carbon Nanotube Field Effect Transistor (CNTFET) SRAM cell (PA9T SRAM) designed for enhanced stability and variation tolerance. The PA9T SRAM achieves power minimization by incorporating a multi-threshold voltage technique (using High Threshold Voltage (HVT) transistors in the cross-coupled inverters) and a stacking effect (via an HVT sleep transistor shared by all read ports). Compared to the conventional structure (CA9T SRAM), the proposed design showed remarkable improvements: power reduction during read and write operations was enhanced by 4.7x and 9.9x, respectively. Furthermore, the reduction in read delay and Power Delay Product (PDP) was improved by 10x. Stability metrics indicated that the hold, read, and write stability were enhanced by 1.4x, 1.2x, and 4.1x, respectively.

The final paper, “A GCN-Attention Model for Precision Irrigation Evaluation,” by Ying Huang and Meng Liu, proposes the innovative and computationally efficient UFO-GCN-SPANet architecture for precise irrigation evaluation, specifically for resource-constrained precision agriculture. The architecture sequentially integrates a linear-complexity Unit Force Operated Vision Transformer (UFO-ViT) for efficient global spatio-temporal feature extraction (overcoming the $O(N^2)$ bottleneck of standard Transformers), Graph Convolutional Networks (GCNs) for spatial dependencies, and a Salient Positions-based Attention Network (SPANet) employing a Significant Position Selection (SPS) algorithm to focus computation on the most informative features. By transforming the soil water content prediction problem into a classification problem, the model guides dynamic irrigation strategy. Experimental results demonstrate the model’s superior predictive performance on a collected tea plantation dataset, achieving an overall accuracy of 0.9596, average precision of 0.9205, average recall of 0.9410, and average F1-Score of 0.9239. This represents a significant improvement, including a 12.07% higher accuracy compared to the GCNConv model.

Finally, I thank the authors for their contributions to this issue of the journal and extend great appreciation to all the reviewers who participated in the editorial process by providing valuable comments in a timely manner to the editors and authors.

Virtual Simulation of Integrated Circuits Combining AP with DE Algorithm

Peipei Hu

Abstract—**As computers develops, virtual simulation technology becomes an important means of integrated circuit design. Therefore, based on the demand for virtual simulation of integrated circuits, a simulation method combining affinity propagation and differential evolution algorithm was proposed. By applying the affinity propagation to circuit fault diagnosis and combining it with differential evolution algorithm, circuit parameters optimization was carried out. These experiments confirm that the fusion of affinity propagation and differential evolution algorithm has a precision of 94.26%, recall of 93.41%, mean F1 of 88.59%, convergence speed of 56.77 seconds, and stability of 93.17%. The affinity propagation performs well in clustering. Especially without pre-defining the classes, it can identify the position and number of class centers automatically. The simulation of integrating affinity propagation and differential evolution algorithm has broad application prospects in virtual simulation of integrated circuits. It can improve simulation effectiveness and performance, providing effective support for circuit design and testing.**

Index Terms—**AP algorithm, differential evolution algorithm, integrated circuits, virtual simulation.**

Original Research Paper
DOI: 10.53314/ELS2529043H

I. Introduction

As technology advances, integrated circuit design faces complex challenges. While meeting performance, power consumption, and area requirements, it is also necessary to consider various issues that may arise during the manufacturing process. Virtual simulation, as an important part of integrated circuit design, can effectively predict and solve these problems [1]. Integrated circuit virtual simulation is an important method for circuit design and testing, which can improve design efficiency and reduce testing costs. However, due to the complexity of integrated circuits and the diversity of faults, traditional simulation methods face certain challenges in terms of efficiency and accuracy [2]. Affinity Propagation (AP) is a commonly used clustering method that can divide the sample set into different categories for circuit fault diagnosis. Differential Evolution (DE) is an optimization algorithm that optimizes circuit parameters to improve circuit performance and stability [3]. The study combines these two algorithms to diagnose circuit faults through AP, and then optimizes circuit parameters through DE to achieve efficient and accurate circuit virtual simulation. The study provides a more efficient and accurate virtual simulation method for integrated circuit design. The research can provide a more efficient and accurate virtual simulation method for IC design, so as to reduce the design cost, shorten the development cycle, and contribute to the development of IC industry. The paper is organized as follows: In the first section, the research background and purpose are proposed, the research content is introduced, and the research status of AP clustering algorithm, differential evolution algorithm and integrated circuit virtual simulation at home and abroad are discussed and analyzed. In the second section, the integrated circuit virtual simulation combining AP clustering algorithm and differential evolution algorithm is proposed. The circuit parameters of differential evolution algorithm are optimized, and the subpopulation is divided by AP clustering algorithm. In the third section, the effectiveness and performance of the algorithm and the virtual simulation circuit are verified by experiments. In the fourth section, the research results are summarized.

II. Related Works

In recent years, AP has gradually become a popular clustering algorithm and has been widely applied. DE is an evolutionary optimization algorithm that has received widespread attention. The information fusing method is an important analyzing means. Zhao et al. proposed a multiple fusing method combing AP and other methods. This study confirmed that it had high classifying accuracy, showing good performance [4]. To provide a reasonable solution for improving the monitoring reliability and economic benefits of the entire industrial process, Ma et al. put forward a distributed quality fault detecting framework with AP. These results confirmed the excellent fault detecting performance of this framework [5]. To improve the classical clustering algorithm, Tang et al. proposed a strategy to extend AP in non-spherical clustering by constructing class similarity of objects. These results confirmed that the adaptive spectral affinity propagation algorithm was superior to benchmark algorithms [6]. How to simultaneously locate multiple high-quality equivalent positioning systems and obtain a uniformly distributed probability density function remains a challenging task. Regarding this, Zhou et al. proposed a method for solving multi-modal multi-objective DE. These results confirmed the superiority of this model, which could locate multi-

Manuscript received on December 29th, 2023. Received in revised form on July 15th and July 25th, 2024. Accepted for publication on July 25th, 2024.

Peipei Hu. Author is with the Office of Academic Affairs, Wuxi Vocational College of Science and Technology, Wuxi, China phone: +86 18961759064; e-mail: 18961759064@163.com.

ple equivalent positioning systems and obtain a uniformly distributed probability density function [7]. To obtain more energy and reduce total costs for offshore wind turbines, Wang et al. proposed a novel accelerated multi-objective DE to globally optimize the device layout. These results confirmed that the optimized hybrid wind farm had a 37.75% increase in power generation compared to pure wind farms, and a 43.65% reduction in wind turbine foundation wave loads [8]. In many classic DE, population size is usually determined by users based on empirical values. And it remains unchanged during the evolution process, which greatly affects the performance of differential evolution. Li et al. proposed a dynamic population reduction DE that combined linear and nonlinear strategy piece-wise functions. These results confirmed that dynamic population reduction DE performed well in overall performance and was significantly superior to other algorithms [9].

In recent years, virtual simulation of integrated circuits have been continuously developing and applied. To improve the linearity of presynaptic current, Cha et al. proposed a neuron circuit with a capacitive cross impedance amplifier integrator. These results confirmed that the proposed neural circuit successfully improved output linearity at a wide range of input current levels [10]. To propose, design, and implement a novel narrowband pass filter based on half mode substrate integrated wave-guide, Mahant et al. proposed using Ansoft high-frequency structure simulator for simulation. These results confirmed that at a center frequency of 11.2 GHz, the return loss was 20.39 dB, the insertion loss was 1.59 dB, and the 3dB fractional bandwidth was 2.58% [11]. Finding a balance between solar thermal gain and solar transmittance is an appropriate strategy for achieving energy efficiency when designing photovoltaic shading systems. Gao et al. proposed using parameterized script modeling to integrate thermal and lighting performance, and utilizing multi-objective optimization to optimize the balance between them. These results confirmed that the integrated framework was feasible and could be extended to the design of other advanced shading systems or building integrated photovoltaics [12]. In the absence of a multiplier circuit, the differential trans-conductance amplifier exhibits two operating modes: incremental and decreasing. Vista et al. proposed a floating memristor model using a single voltage difference trans-conductance amplifier and grounded passive components. These results confirmed the performance of the proposed memristor by examining the characteristics of all possible temperature and process angles through the Cadence Virtuoso layout [13]. When connected to the grid, photovoltaic power generation systems can improve the quality of electricity. Rajagopal et al. proposed an adaptive neural fuzzy inference system based on an improved moth flame optimization algorithm. These results confirmed that the designed controller could maintain the exchange of active and reactive power, regulate the DC bus voltage, grid voltage, and grid current [14]. Using an optimized circuit breaker model can estimate the energy absorption requirements during fault current suppression. Torwelle et al. proposed a universal fault current calculating means with equations. These results confirmed that this method could represent temporary blockages over a large time after a fault occurs, facilitating the reactors and breakers' determination in the power grid [15].

To sum up, the complexity of IC design requires that virtual simulation technology not only has efficient diagnostic capability, but also has accuracy in parameter optimization. Existing research, such as the work of Lee S and Ali Y, has made progress in battery performance simulation, but there is still room for improvement in the accuracy and efficiency of circuit simulation [16]. In addition, Yan Y et al. 's research on field effect transistors has promoted the development of integrated circuits, but has not fully solved the challenges of simulation technology in fault diagnosis and parameter optimization [17]. In view of these shortcomings, a combination of AP clustering algorithm and differential evolution algorithm is proposed to improve the efficiency and accuracy of IC virtual simulation. The innovation of this research lies in the development of a new simulation method by integrating the adaptive clustering ability of AP clustering algorithm and the global optimization characteristics of DE algorithm. This method shows higher precision, recall and F1 mean in circuit fault diagnosis and parameter optimization, and has faster convergence speed and higher stability. The study also explores the combination of multi-objective optimization algorithms and neural network techniques to provide new perspectives and tools for building energy efficient design, and this approach has been validated in multiple climate regions, showing its broad applicability and overall performance improvement.

III. Optimization of Circuit Parameters and Sub-population Partitioning by Integrating AP

In this study, AP is used for clustering analysis of components and sub-circuits. By measuring the similarity between components, they can be divided into different clusters. DE is used to optimize the parameter configuration of this circuit. By searching and optimizing in the parameter space, the optimal circuit configuration can be found to meet design and performance requirements.

A. Circuit Parameters Optimization based on Differential Evolution Algorithm

In the field of integrated circuit design, we are faced with the dual challenges of fault diagnosis and parameter optimization. To address this challenge, the study proposes a novel approach that combines the advantages of differential evolution and adaptive particle swarm clustering. The differential evolution algorithm, known for its global search capability, iterates to find the optimal circuit parameter configuration by simulating the process of natural selection. The adaptive particle swarm clustering algorithm, with its unique self-organizing characteristics, can automatically identify the similarities of components in the circuit and divide them into different categories, so as to assist fault diagnosis. The workflow of this fusion method can be simplified into the following steps: Firstly, the circuit components are classified by the adaptive particle clustering algorithm, and the possible faulty component groups are identified; The differential evolution algorithm is then optimized for the parameters of these groups to improve the overall performance and stability of the circuit. This process not only improves the accuracy of fault detection, but also speeds up the speed of problem solving by optimizing circuit parameters, thus playing an important role in integrated circuit design and testing. In this way, problems can be found and solved at the design stage, reducing manufacturing costs and shortening the research and development cycle. DE is an optimizing algorithm used to solve continuous optimization problems. This algorithm originates from genetic algorithms and introduces differential operations in evolution to achieve efficient exploration and optimization of the search space. DE seeks the optimal solution by simulating the process of biological evolution. It is usually used to handle continuous optimization problems. In virtual simulation of integrated circuits, DE can find circuit parameters' optimal configuration to improve circuit performance or reduce power consumption. By continuously adjusting the parameters of the circuit, such as resistance, capacitance, and inductance, better circuit design and performance optimization can be achieved [18]. Fig. 1 shows the advantages and disadvantages of DE.

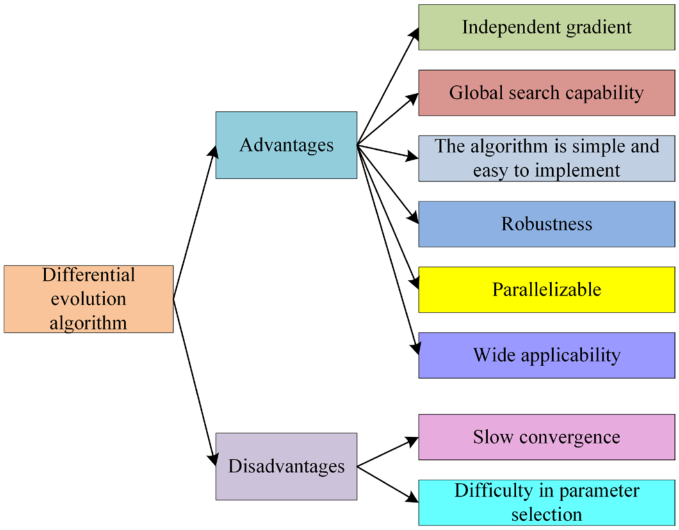


Fig. 1. Advantages and disadvantages of differential evolution algorithm.

DE does not require gradient information for solving problems. This enables DE to be applied to various types of optimization problems, including nonlinear, non-convex, and nonsmooth problems. DE has strong global search ability and can find better solutions in optimization problems of multi-dimensional, multi-modal, and non-convex functions. By randomly generating mutated individuals and global search strategies, it can effectively seek the global optimal solution. The basic principle of DE is relatively simple, easy to understand and implement. It only requires three basic operations: mutation, crossover, and selection. Fig. 2 shows the mutation operation.

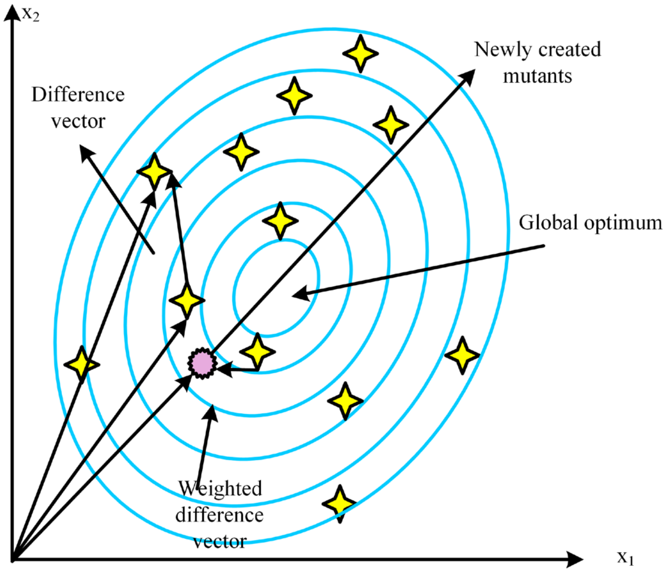


Fig. 2. Mutation operation.

DE is relatively insensitive to the selection of initial solutions and population size, and has strong robustness. Therefore, it has good adaptability and application flexibility in practice. DE can be easily parallelized, allowing for the use of multi-core or distributed computing resources to accelerate algorithm execution. The parameter selection in DE has a significant impact on its performance and effectiveness. Choosing appropriate parameter values can improve algorithm's converging speed and quality, but it also increases the difficulty of parameter tuning. DE is used to optimize the parameter configuration of the circuit. By searching and optimizing in the parameter space, the optimal circuit configuration can be found to meet design and performance requirements. DE uses floating-point vectors to encode the population and generate individuals in the population. DE is optimized by selecting two individuals from the parent and performing a difference to obtain the difference vector. Secondly, differential vectors are used to accumulate another individual to obtain a subject. Then the parents and experimental individuals are crossed to produce new offspring. On this basis, by screening the parents and children, offspring that meet the conditions are selected [19]. Equation (1) is the assumed optimization model.

$$\min f(x_1, x_2, ..., x_D), x_j^L \le x_j \le x_j^U, j = 1, 2, ..., D \tag{1}$$

In equation (1), D is solution space's dimension. x_j^L, x_j^U represent the critical values of the j-th component x_j, respectively. The initialization population is represented by equation (2).

$$x_i(0) \mid x_{j.i}^L \le x_{j,i}(0) \le x_{j.i}^v \tag{2}$$

In equation (2), $i = 1, 2, ..., NP; j = 1, 2, ..., D$. $x_i(0)$ stands for the 0th generation's i-th individual. $x_{j,i}(0)$ means the 0th generation individual i's j-th gene. NP stands for the population size. The next step is to randomly generate various groups of individuals, represented by equation (3).

$$x_{j,i}(0) = x_{j,i}^L + rand(0,1) \cdot x_{j,i}^U - x_{j,i}^L \tag{3}$$

In equation (3), $rand(0,1)$ stands for a uniformly distributed random value within [0, 1]. The commonly used differential strategy in DE is represented by equation (4).

$$v_i(g+1) = x_{i1}(g) + F \cdot (x_i(g) - x_{ij}(g)) \tag{4}$$

In equation (4), $i \neq r_1 \neq r_2 \neq r_3$. F stands for a scaling factor. $x_i(g)$ refers to the g -th generation's individual i. The limiting condition for the g -th generation population is represented by equation (5).

$$x_i(g) \,|\, x_{j.i}^{L} \leq x_{j,i}(g) \leq x_{j.i}^{v} \tag{5}$$

The mutated intermediate is represented by equation (6).

$$x_i(g+1) \,|\, x_{j.i}^{L} \leq x_{j,i}(g+1) \leq x_{j.i}^{v} \tag{6}$$

Next, the g -th generation population and its variant intermediates are used for crossover in equation (7).

$$u_{j,i}(g+1) = \begin{cases} v_{j,i}(g+1), rand(0,1) \leq CR, j = j_{rand} \\ x_{j,i}(g) \end{cases} \tag{7}$$

In equation (7), CR stands for the crossover probability. j_{rand} represents a random integer within [1, D]. The selection operation adopts the greedy algorithm in equation (8).

$$x_i(g+1) = \begin{cases} u_i(g+1), f(u_i(g+1)) \leq f(x_i(g)) \\ x_i(g) \end{cases} \tag{8}$$

Fig. 3 shows the basic steps of DE.

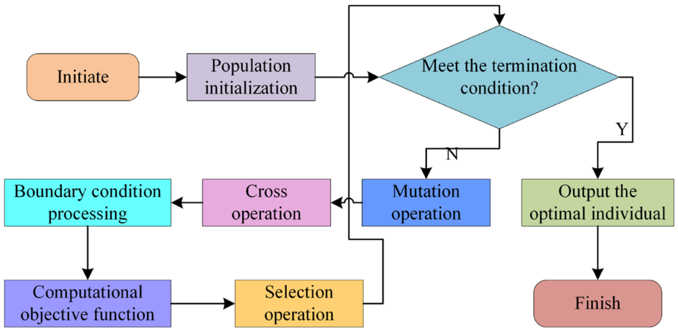

Initiate
Population initialization
Meet the termination condition?
N
Y
Boundary condition processing
Cross operation
Mutation operation
Output the optimal individual
Computational objective function
Selection operation
Finish

Fig. 3. Basic steps of differential evolution algorithm.

Firstly, the population is initialized, and some individuals are randomly generated to form the initial population. A fitness function can evaluate each individual' fitness values, and some individuals are selected as parent individuals. For each parent individual, mutated individuals are generated through differential operations. Differential operation involves selecting two other individuals from the population and calculating the differential individuals through linear combination. For each mutated individual, crossover operation is performed to generate offspring individuals. A fitness function can evaluate each offspring individual's fitness values, and some individuals are selected as the parents of the next generation. These three operations are repeated until the termination conditions are met. However, DE faces some challenges, such as low search efficiency for high-dimensional problems and the need for appropriate parameter adjustments to improve algorithm performance.

B. Combining Differential Evolution Algorithm with AP for Sub-Population Partitioning

To optimize the performance of DE, researchers have proposed many improvements and variants, such as adaptive DE, multi-objective DE, etc. These improvements and variants can be selected and applied according to the characteristics of specific problems to improve its search efficiency and convergence speed [20]. Research chooses to merge AP to partition the sub-populations of DE. AP is based on the similarity and accessibility of data points. It allocates data points to different clusters through iterative calculation, making the similarity of data points within the same cluster large and the difference of different clusters' data points large. Fig. 4 shows the AP flowchart.

The responsibility and availability in AP are indicators used to measure the degree of similarity and affinity between data points. In AP, each data point calculates its own responsibility and availability based on the similarity and affinity with other data points. Responsibility indicates the suitability to which a data point selects other data points as its pattern, while availability indicates the suitability to which other data points choose that data point as their pattern [21]. Specifically, for a data point i, its responsibility $r(i,k)$ refers to the suitability of data point k for i as its pattern. It is equal to the sum of the

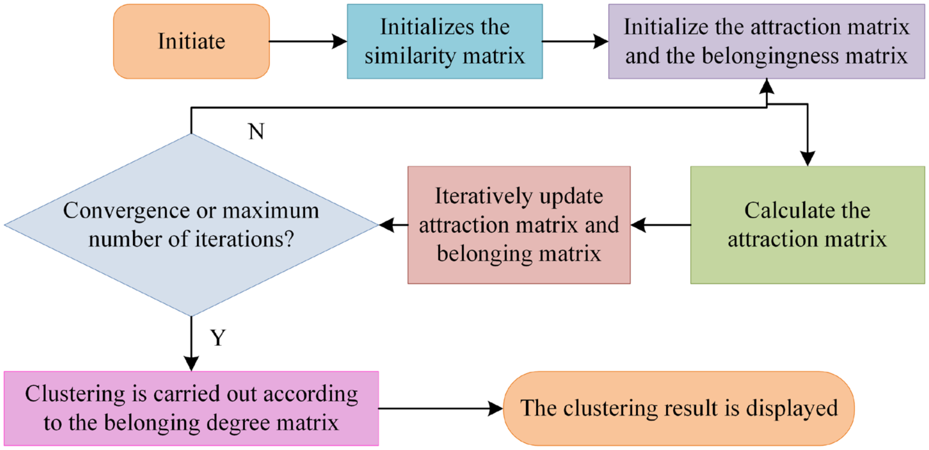

Initiate
Initializes the similarity matrix
Initialize the attraction matrix and the belongingness matrix
N
Convergence or maximum number of iterations?
Iteratively update attraction matrix and belonging matrix
Calculate the attraction matrix
Y
Clustering is carried out according to the belonging degree matrix
The clustering result is displayed

Fig. 4. Flow chart of AP algorithm.

degree to which k chooses i as its attribution and the degree to which k chooses other data points j as its attribution minus the sum of the degree to which j chooses k as its pattern and j chooses other data point l as its attribution in equation (9).

$$r\left(i,\ k\right)=s\left(i,\ k\right)-max\{a\left(j,\ k\right)+s(j,\ k)\} \tag{9}$$

In equation (9), $s(i,k)$ represents i and k's similarity. $a(j,k)$ represents the degree to which j chooses k as its mode availability. Availability $a(i,k)$ indicates the suitability of k's choice of i as its mode. It is equal to the sum of the responsiveness of i selecting k as its mode and i selecting j as its mode minus the maximum sum of the responsiveness of k selecting j as its mode and k selecting other data point l as its mode in equation (10).

$$a\left(i,k\right)=min\{0,r\left(k,k\right)+sum(max\{0,r(j,k)\})\} \tag{10}$$

In equation (10), sum represents the sum of all data points j. To better utilize the information from the previous iteration while reducing data fluctuations, during the information transmission process, the value from the previous iteration will be multiplied by $1-\lambda$ times and the current value will be multiplied by λ times. The change in responsiveness is represented by equation (11).

$$r(i,k)=(1-\lambda)\times r(i,k)+r(i,k)_{last} \tag{11}$$

Equation (12) represents the change of availability.

$$a(i,k)=(1-\lambda)\times a(i,k)+a(i,k)_{last} \tag{12}$$

By iteratively calculating responsibility and availability, AP divides data points into multiple clusters and selects one data point from each cluster as its pattern. In AP, self-availability is a special case of availability. It represents the degree to which data points choose themselves as their patterns. Self-availability is calculated by subtracting the sum of the responsibilities of data points selecting themselves as their patterns and other data points selecting themselves as their patterns from the sum of the responsibilities of other data points selecting themselves as their patterns, as shown in equation (13).

$$a\left(i,i\right)\ =min\{0,r\left(i,\ i\right)+sum(max\{0,r(j,i)\})\} \tag{13}$$

In equation (13), sum represents the sum of all data points j. In AP, self-availability plays a role in self-evaluation, indicating the degree to which the data point is considered suitable for its own cluster. If the self-availability is positive, it indicates that the data point is considered a good pattern and becomes a representative of its cluster. If the self-availability is negative, it indicates that the data point is not suitable as a pattern and will not be selected as a representative of clustering. The combination of self-availability and membership helps AP achieve adaptive clustering. By iteratively calculating responsiveness and belonging, data points will automatically adjust their relationships with each other, forming stable clustering results. Fig. 5 shows the sub-population partitioning based on AP.

The sub-population partitioning method based on AP can be applied to various optimization problems, helping to solve complex and high-dimensional optimizing problems, and improving the efficiency and accuracy of optimization algorithms. Meanwhile, this method can also be combined with other optimization algorithms to further improve optimization performance. Backpropagation Neural Networks (BPNN) are used to optimize and train AP clustering algorithm. By comparing the predicted output of the output layer with the target value, the network achieves fine adjustment of the weight, so as to reduce

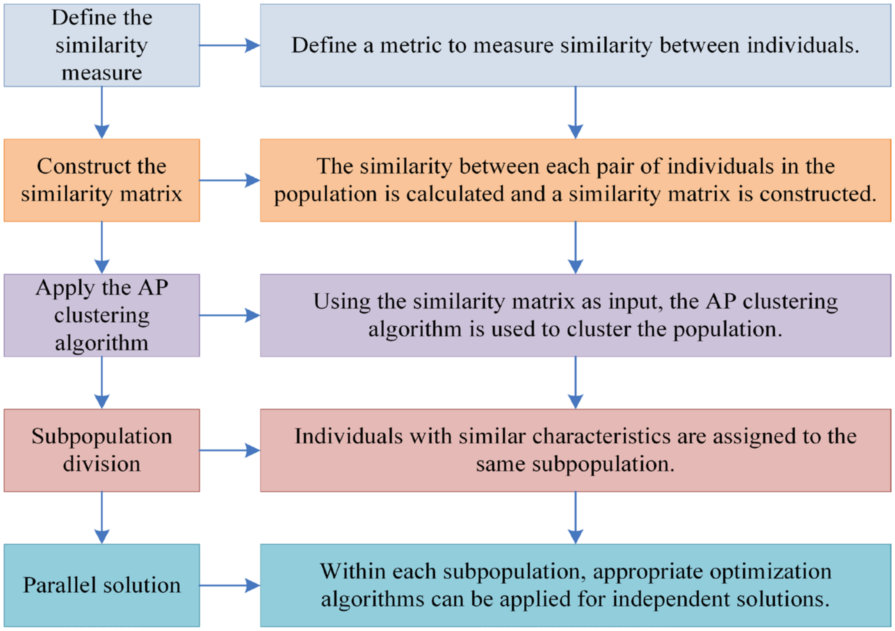

Define the similarity measure
Define a metric to measure similarity between individuals.
Construct the similarity matrix
The similarity between each pair of individuals in the population is calculated and a similarity matrix is constructed.
Apply the AP clustering algorithm
Using the similarity matrix as input, the AP clustering algorithm is used to cluster the population.
Subpopulation division
Individuals with similar characteristics are assigned to the same subpopulation.
Parallel solution
Within each subpopulation, appropriate optimization algorithms can be applied for independent solutions.

Fig. 5. Sub-population division process.

the prediction error. This process involves several iterations until the network output reaches the desired level of accuracy. The transmission of data in the network first passes through the input layer, then into the hidden layer, and finally generates the prediction result at the output layer. The error is evaluated by comparing the predicted value of the network with the actual observed value, and then using the gradient descent algorithm to optimize the weight and bias of the network. This circular process of forward and backpropagation continues until a preset termination condition is met. The number of nodes in the input layer is directly related to the feature dimension of the input data, while the number of nodes in the hidden layer is the optimal solution obtained through experimental adjustment, which is usually determined by trial and error, as described in formula (14) [22].

$$m = \sqrt{n + l} + a \tag{14}$$

In equation (14), a is the constant from 1 to 10, m is the number of nodes in the hidden layer, l is the number of nodes in the output layer, and n is the number of nodes in the input layer. Through this method, BPNN can learn the complex mapping relationship between input data and output results, and provide an accurate prediction model for building energy efficiency optimization design. In the neural network model of this study, the number of nodes in the output layer is determined according to the number of target variables involved in the research problem. The nonlinear activation function is used in the output layer to enhance the prediction ability of the model. Among the many nonlinear activation functions, hyperbolic functions are widely used because of their ability to handle outputs in the range [-1, 1], which makes them particularly suitable for nonlinear transformations of hidden layers. The specific expression of hyperbolic function is given in formula (15).

$$f(x) = \frac{1}{1 + e^{x}} \tag{15}$$

In Formula (15), through this setup, the network can more effectively capture and learn complex nonlinear relationships between input data and output results, thereby improving prediction accuracy and model performance. There is a direct correlation between the sample size used in the experiment and the accuracy of the model's predictions. Increasing the number of samples can improve the response accuracy of the model, but this improvement will gradually become saturated with the increase of the sample size. When the sample size reaches a certain threshold, the accuracy of the model will stabilize at a specific level, and further increasing the sample size will not lead to significant accuracy improvement. At the same time, the expansion of the scale of the network will lead to the increase of the complexity of the mapping relationship. When initializing network weights, there are usually two strategies: one is to give the weights a small enough initial value to avoid saturation of the activation function; The second is to ensure that the weights are initialized to a balanced number of positive and negative values. This approach helps to ensure that the selected model has the best generalization ability and prediction accuracy when dealing with real-world problems. In integrated circuits, AP is widely used in circuit fault diagnosis and testing. Integrated circuits are complex circuit systems composed of various electronic components, which may have faults or defects. Through AP, faults in circuits can be identified and classified. Fig. 6 shows the specific steps.

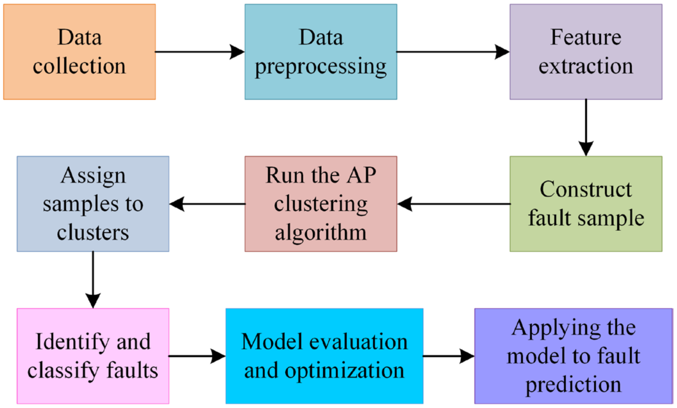

Data collection
Data preprocessing
Feature extraction
Construct fault sample
Run the AP clustering algorithm
Assign samples to clusters
Identify and classify faults
Model evaluation and optimization
Applying the model to fault prediction

Fig. 6. Fault identification and classification steps in a circuit.

In Fig. 6, in the fault diagnosis of integrated circuits, the AP clustering algorithm is applied through a continuous iterative process, starting from data collection and feature extraction, to build a similarity matrix to quantify the similarity between components. After initializing the algorithm parameters, the damping coefficient controls the propagation of information among the circuit nodes, and the priority affects the number of clusters. In the iterative calculation, the responsiveness and belongingness of each node are updated to assess their suitability as cluster centers. With the progress of the algorithm, the center node of the cluster is gradually determined, and other samples are allocated to the corresponding cluster based on similarity. Finally, the clustering results are verified to ensure accuracy, provide basis for fault analysis, identify fault causes, and propose solutions.

IV. Performance Analysis of Virtual Simulation of Integrated Circuits Integrating AP and Differential Evolution Algorithm

Two 2.5DIC benchmark sets were selected from dies in IWLS ‘05 and Opencores benchmarks, assuming 20 scan chains per Die and a test frequency f set to 50MHz. In this experiment, the algorithm is verified by using Matlab R2010a software on an Intel computer equipped with 4G memory. For differential evolution algorithm and adaptive particle swarm clustering algorithm, after a series of parameter tuning experiments, the following optimal parameter combinations are determined to ensure the optimal performance of the algorithm on test scheduling problems. The differential evolution algorithm selects a scaling factor F of 0.8, a crossover probability Cr of 0.9, and a population size of 50. For the adaptive particle cluster clustering algorithm, the damping coefficient is determined to be 0.6, and the priority p is 1.5. Such parameter configuration helps the algorithm to automatically identify the cluster center of circuit components without the need to define the number of classes in advance, thus effectively supporting the fault diagnosis process. The study first analyzed the performance of AP and DE, and then conducted virtual simulation and verification of the optimized circuit configuration. By analyzing and comparing simulation results, the performance and reliability of the circuit can be evaluated, and necessary adjustments and improvements can be made.

A. Performance Analysis of Integrating AP and Differential Evolution Algorithm

To test the performance of integrating AP and DE, a comparative analysis was conducted between separate DE, separate AP, Ant Colony Algorithm (ACA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Neural Network (ANN), Simulated Annealing Algorithm (SAA). These algorithms are all optimization algorithms, but they differ when dealing with optimization problems. The following is a comparative analysis of these algorithms, and Table I is the results of different indicators.

In Table I, the algorithm combination combining DE and AP has the best performance, its accuracy reaches 94.26%, recall rate is 93.41%, F1 value is 88.59%, convergence speed is 56.77 seconds, and stability is 93.17%. This shows that the fusion algorithm has high accuracy and reliability in identifying circuit faults and optimizing parameters. When using DE or AP alone, performance is slightly lower than fusion algorithms, but still better than GA, SAA, ACA, PSO, and ANN. In particular, the convergence speed of the fusion algorithm is the fastest, which means that it can find a solution in a short time, which is especially important for the circuit design and testing process requiring fast iteration and real-time feedback, indicating that the fusion AP clustering algorithm and differential evolution algorithm have higher accuracy and better performance. To further test the performance of fusion AP and DE, the precision, recall, and fitness of fusion AP and DE were compared and analyzed with PSO in Fig. 7.

Fig. 7(a) shows the fusion of AP and DE, with a relatively high precision rate above 0.9, and a recall above 0.8. Fig. 7(b) shows the accuracy and recall of PSO, ranging from 0.7 to 0.8. This indicates that the fusion of AP and DE has higher performance. To verify the clustering effect of AP, a visualization method was used to analyze and compare the clustering results in Fig. 8.

TABLE I
DIFFERENT INDEX RESULTS OF SIX ALGORITHMS

Algorithm index	Precision(%)	Recall(%)	F1 value(%)	Rate of convergence(s)	Stability(%)
DE+AP	94.26	93.41	88.59	56.77	93.17
DE	83.27	81.53	80.26	76.39	88.23
AP	82.73	88.97	83.69	78.52	83.64
GA	73.68	82.38	76.30	70.67	87.31
SAA	86.11	77.25	76.25	39.71	84.15
ACA	69.34	69.58	70.38	48.26	72.64
PSO	81.27	82.34	82.55	52.34	83.55
ANN	78.59	76.53	79.69	47.19	80.42

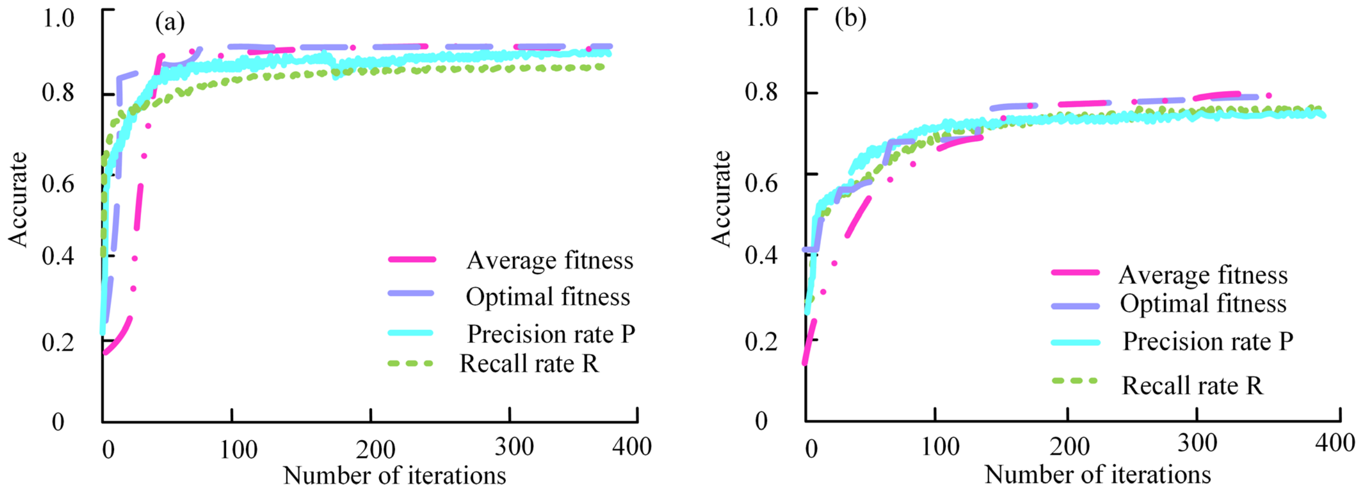


Fig. 7. Comparison of accuracy, recall and fitness of different algorithms: (a) AP+DE; (b) particle swarm optimization algorithm.

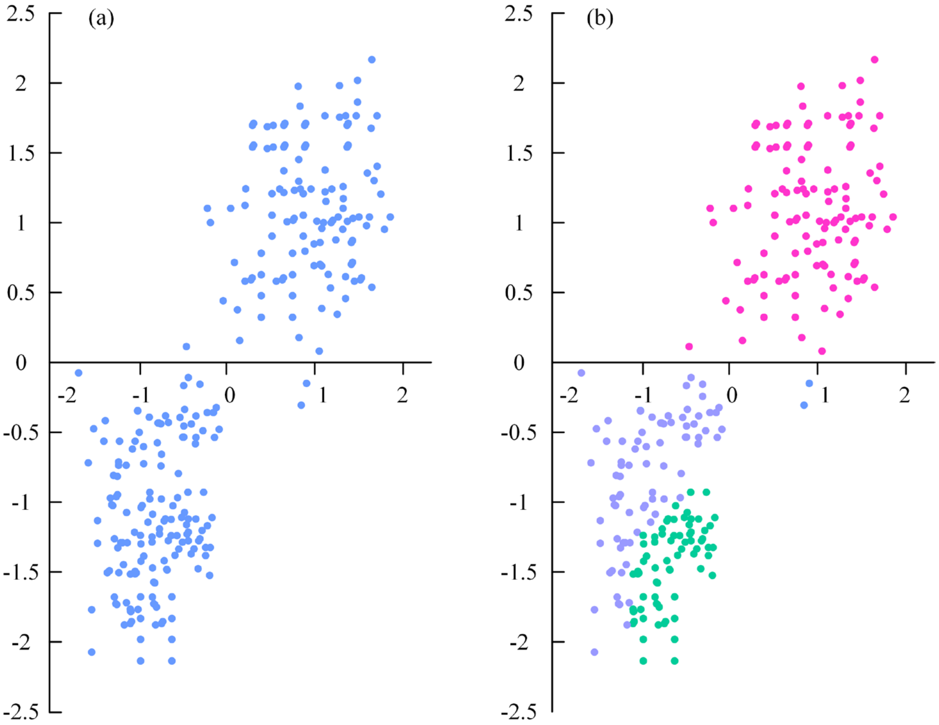
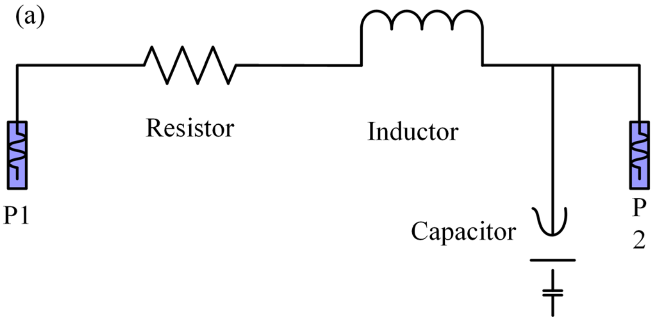


Fig. 8 Clustering effect of AP algorithm: (a) raw data; (b) clustering result.

Fig. 8(a) presents the original integrated circuit data. Fig. 8(b) shows the integrated circuit data after clustering. AP clusters data points on the similarity matrix, with the goal of minimizing the distance between data points and their class representative points. AP performs well in clustering, especially without pre-defining the classes. It can identify the position and number of class centers automatically, maximizing the sum of similarities between all data points and the nearest class representative point.

B. *Virtual Simulation Performance Analysis of Integrated Circuits*

In the developed circuit simulation tool, the method of integrating AP and DE is used. The circuit is built in the circuit simulation tool, with a scanning frequency range of 0.1 GHz-1 GHz and a gradient of 0.1 GHz. The S parameter results are compared in Fig. 9.



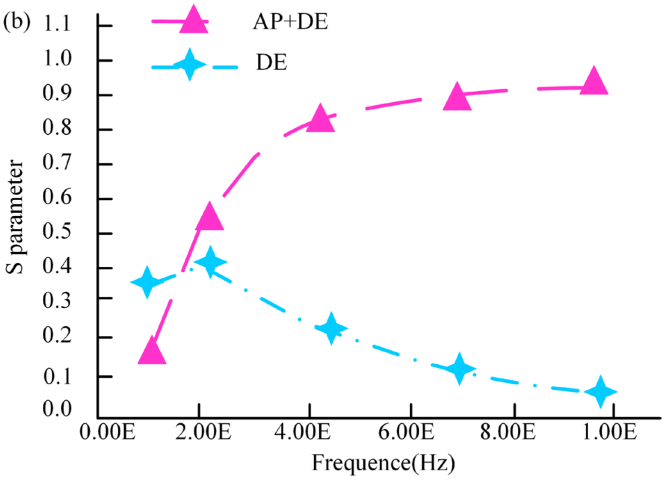


Fig. 9. Integrated circuit simulation diagram and S parameter: (a) integrated circuit virtual simulation diagram; (b) sparameter.

Fig. 9(a) shows the virtual simulation of the integrated circuit, with the scanning frequency set to 0.1 GHz-1 GHz. Fig. 9(b) shows a comparison of the results of fusing AP and DE, as well as individual DE. The points corresponded to each frequency coincide, and the results were consistent, indicating that using the fusion AP and DE to calculate the S parameter was feasible. Further analysis is conducted on the performance of integrating AP and DE in dealing with complex circuit problems, and the accuracy and efficiency of PSO in dealing with complex circuit problems are compared in Fig. 10. In the context of circuit simulation and fault diagnosis, accuracy is a measure of an algorithm's ability to correctly identify faults or optimize parameters. It is determined by calculating the percentage of instances in which the algorithm correctly predicts a failure or finds an effective solution to the total number of instances. Efficiency refers to the speed at which an algorithm solves a problem, including the use of computational resources and the time it takes to solve the problem. In circuit design, efficient algorithms can complete simulation and optimization in a relatively short time, thus speeding up the design cycle.

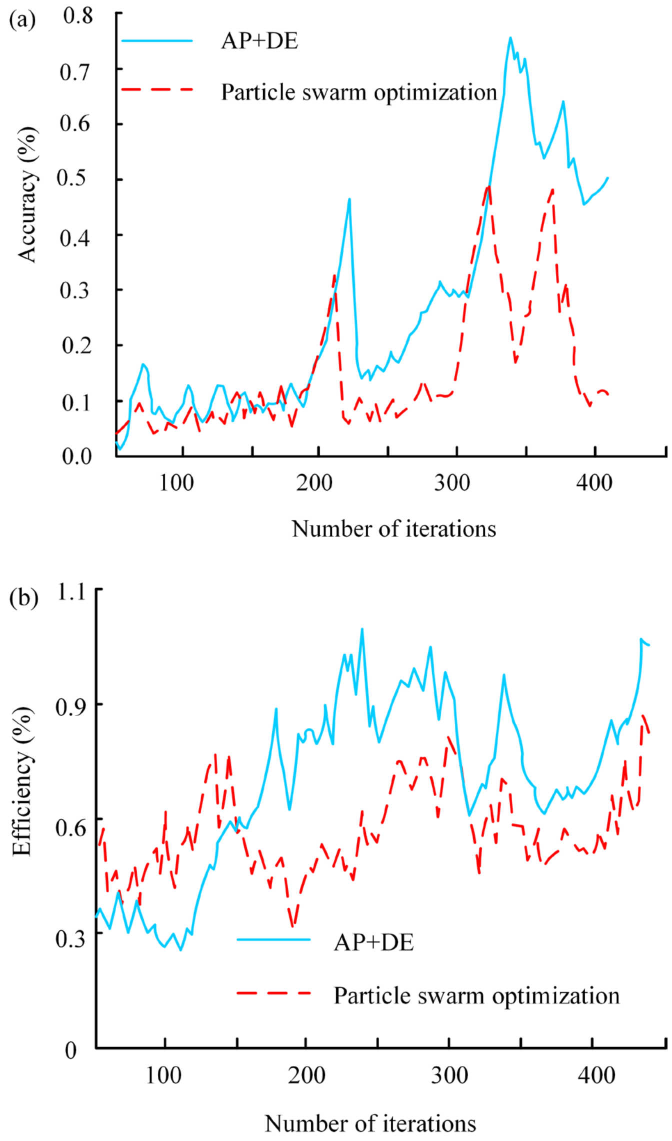


Fig. 10. Accuracy and efficiency in dealing with complex circuit problems: (a) the accuracy; (b) the efficiency.

In Fig. 10, the fusion of AP and DE had higher accuracy and efficiency. After 300 iterations, the accuracy reached 89% and the efficiency reached 100%, indicating that the fusion of AP and DE performed better when dealing with complex integrated circuit problems. To observe the mean features of the sequence before and after change point detection, the following is a graph of the mean features before and after change point detection in Fig. 11.

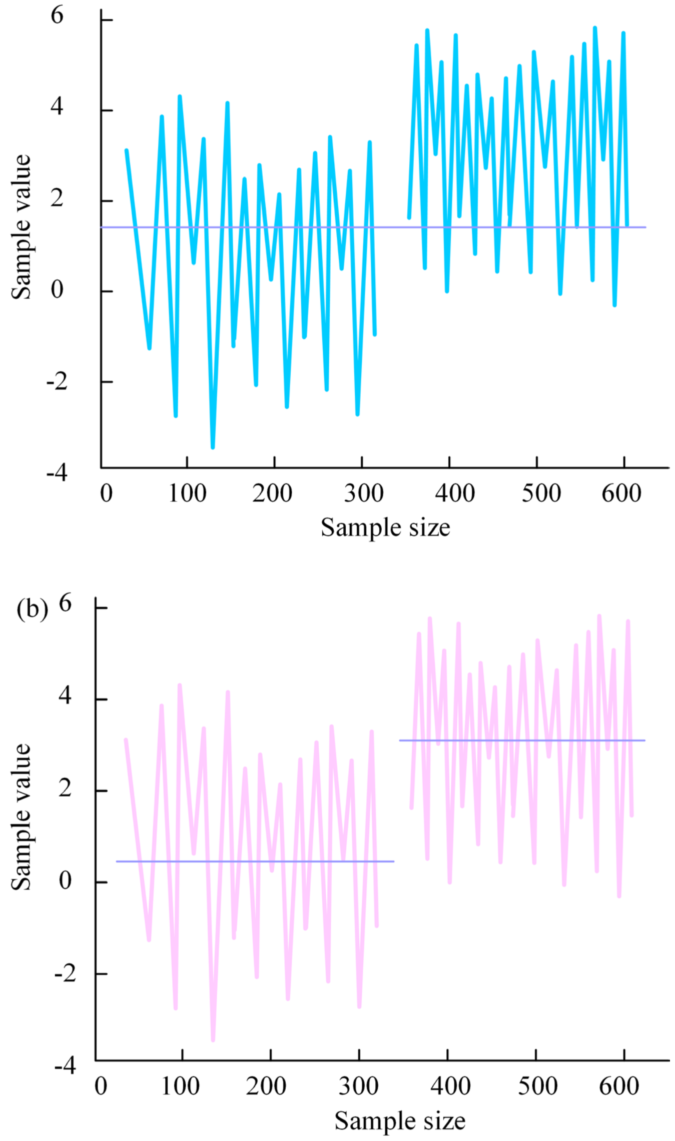


Fig. 11. The mean value feature map before and after change point detection: (a) mean value of sample point features before detection; (b) mean of the features of the sample points after detection.

The straight line in Fig. 11(a) represents the data mean feature of the unchanged point model. The straight line in Fig. 11(b) represents the data mean feature under the change point model, which divided the data into two segments. Each segment characterized the mean of these data, which was more in line with the true data. This indicates that the proposed DE can detect the position of change points and better explain the characteristics of normal distribution means.

V. Conclusion

Virtual simulation provides an effective means for integrated circuit design to simulate circuit operation on a computer. It helps to detect and solve problems early in the design phase, thereby reducing manufacturing costs and shortening research and development cycles. The study adopted a simulation method that integrates AP and DE, aiming to improve the accuracy of circuit fault diagnosis and the efficiency of circuit parameter optimization. These experiments confirm that the fusion method of AP and DE performs well in terms of precision, ultimately stabilizing above 0.9. Its recall remains above 0.8. Specifically, after 300 iterations, the accuracy of this method reaches 89% and the efficiency reaches 100%. Compared to traditional simulation methods, this fusion algorithm exhibits higher accuracy and efficiency in circuit fault diagnosis and parameter optimization. However, this method of integrating AP and DE is complex, involving multiple parameters and operations, resulting in increased computational difficulty and time cost. To improve the efficiency and reliability of virtual simulation of integrated circuits, future research can explore more optimization methods.

References

[1] S. Chander, P. Vijaya, R. Fernandes, A. P. Rodrigues, and R. Maheswari, "Dolphin-political optimized tversky index based feature selection in spark architecture for clustering big data," *Adv. Eng. Softw.*, vol. 176, no. 1, pp. 103331-1-13, Feb. 2023, DOI: 10.1016/j.advengsoft.2022.103331.

[2] G. H. Gao, P. Wang, and H. Wang, "Follow-the-leader motion strategy for multi-section continuum robots based on differential evolution algorithm," *Ind. Robot.*, vol. 48, no. 4, pp. 589-601, Jun. 2021, DOI: 10.1108/IR-01-2021-0001.

[3] X. Zhang and X. Wang, "Hybrid-adaptive differential evolution with decay function applied to transmission network expansion planning with renewable energy resources generation," *IET Gener. Transm. Distrib.*, vol. 16, no. 14, pp. 2829-2839, Oct. 2022, DOI: 10.1049/gtd2.12296.

[4] H. Zhao, J. Mi, and M. Liang, "A multi-granularity information fusion method based on logistic regression model and dempster-shafer evidence theory and its application," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 10, pp. 3131-3142, Jun. 2022, DOI: 10.1007/s13042-022-01584-w.

[5] L. Ma, M. Wang, J. Dong, and K. Peng, "A novel distributed detection framework for quality-related faults in industrial plant-wide processes," *Neurocomputing*, vol. 492, no. 1, pp. 126-136, Jul. 2022, DOI: 10.1016/j.neucom.2022.04.014.

[6] L. Tang, L. Sun, C. Guo, and Z. Zhang, "Adaptive spectral affinity propagation clustering," *J. Syst. Eng. Electron.*, vol. 33, no. 3, pp. 647-664, Jun. 2022, DOI: 10.23919/JSEE.2022.000060.

[7] T. Zhou, Z. Hu, Q. Su, and W. Xiong, "A clustering differential evolution algorithm with neighborhood-based dual mutation operator for multimodal multiobjective optimization," *Expert Syst. Appl.*, vol. 216, pp. 119438.1-119438.18, Apr. 2023, DOI: 10.1016/j.eswa.2022.119438.

[8] Y. Wang, Z. Liu, and H. Wang, "Proposal and layout optimization of a wind-wave hybrid energy system using GPU-accelerated differential evolution algorithm," *Energ.*, vol. 239, pp. 121850.1-121850.16, Jan. 2022, DOI: 10.1016/j.energy.2021.121850.

[9] K. Li, X. Fu, F. Wang, and H. Jalil, "A dynamic population reduction differential evolution algorithm combining linear and nonlinear strategy piecewise functions," *Concurr. Comput.: Pract. Exp.*, vol. 34, no. 6, pp. 1-14, Dec. 2022, DOI: 10.1002/cpe.6773.

[10] Y. J. Cha, K. H. Park, J. H. Lee, and B. G. Park, "An integrate-and-fire neuron with capacitive trans-impedance amplifier for improving linearity in spiking neural networks," *Solid-State Electron.*, vol. 197, pp. 108435.1-108435.5, Nov. 2022, DOI: 10.1016/j.sse.2022.108435.

[11] K. Mahant, H. K. Mewada, A. V. Patel, A. Vala, and J. Chaudhari, "HMSIW based highly selective filter for radar applications," *Circuit World*, vol. 47, no. 2, pp. 206-212, Jul. 2021, DOI: 10.1108/CW-03-2020-0039.

[12] Q. Gao, Y. Yang, and Q. Wang, "An integrated simulation method for PVSS parametric design using multi-objective optimization," *Front. Archit. Res.*, vol. 11, no. 3, pp. 509-526, Jun. 2022, DOI: 10.1016/j.foar.2021.11.003.

[13] J. Vista and A. Ranjan, "Flux controlled floating memristor employing VDTA: Incremental or decremental operation," *IEEE Trans. Comput. -Aid. Des. Integr. Circuit. Syst.: Publ. IEEE Circ. Syst. Soc.*, vol. 40, no. 2, pp. 364-372, Jun. 2021, DOI: 10.1109/TCAD.2020.2999919.

[14] S. Rajagopal and P. Umapathy, "An investigation of grid-integrated photovoltaic system intended for hybrid moth flame optimization using anfis techniques," *J. Intell. Fuzzy Syst.: Appl. Eng. Technol.*, vol. 42, no. 3, pp. 2505-2519, Feb. 2022, DOI: 10.3233/JIFS-211748.

[15] P. Torwelle, A. Bertinato, B. Raison, T. D. Le, and, M. Petit, "Fault current calculation in MTDC grids considering MMC blocking," *Electr. Power Syst. Res.*, vol. 207, pp. 107662.1-107662.12, Jun. 2022, DOI: 10.1016/j.epsr.2021.107662.

[16] S. Lee and Y. Ali, "An integrated experimental and modeling study of the effect of solid electrolyte interphase formation and Cu dissolution on CuCo2O4-based Li-ion batteries," *Int. J. Energ. Res.*, vol. 46, no. 3, pp. 3017-3033, Oct. 2022, DOI: 10.1002/er.7359.

[17] Y. Yan, Y. Zhao, and Y. Liu, "Recent progress in organic field-effect transistor-based integrated circuits," *J. Polym. Sci.*, vol. 60, no. 3, pp. 311-327, Aug. 2022, DOI: 10.1002/pol.20210457.

[18] C. Van Kien, H. P. H. Anh, and N. N. Son, "Adaptive inverse multilayer fuzzy control for uncertain nonlinear system optimizing with differential evolution algorithm," *Appl. Intell.*, vol. 51, no. 1, pp. 527-548, Aug. 2021, DOI: 10.1007/s10489-020-01819-9.

[19] K. C. Lin and C. C. Liao, "Simulation of slurry residence time during chemical-mechanical polishing using 3-D computational fluid dynamics," *Chem. Eng. Res. Des.*, vol. 191, no. 1, pp. 375-386, Jan. 2023, DOI: 10.1016/j.cherd.2023.01.025.

[20] S. Zhang, G. Li, Y. Zheng, and Y. Liu, "Research on enterprise terminal network planning under the new retail format," *Int. J. Simulat. Process Model.*, vol. 18, no. 3, pp. 200-210, Nov. 2022, DOI: 10.1504/IJSPM.2022.126889.

[21] S. Choudhuri, S. Adeniye, and A. Sen, "Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation," *Artif. Intell. Appl.*, vol. 1, no. 1, pp. 43-51, Oct. 2023, DOI: 10.47852/bonviewAIA2202524.

[22] E. Wang, X. Liu, and J. Wan, "An improved filtering algorithm for indoor localization based on DE-PSO-BPNN," *J. Intell. Fuzzy Syst.*, vol. 44, no. 6, pp. 9513-9525, Jun. 2023, DOI: 10.3233/JIFS-230178.

High-Performance and Resource-Efficient Squaring Architecture for FPGA Platforms

Burhan Khurshid

Abstract—**Hardware-intensive signal processing has seen tremendous growth over the last few decades, owing to the advances in VLSI technology. This has resulted in a significant paradigm shift, wherein different computational functionalities are increasingly implemented using different hardware platforms. The squaring function is one such operation that finds its application in many signal-processing tasks. Since squaring is a specific case of multiplication, traditional multiplication algorithms can be adapted to create high-performance squaring architectures. In this paper, we present a squaring architecture that is based on the CORDIC algorithm. The hardware efficiency of the CORDIC algorithm enables it to compute different mathematical functions using only shift and add operations. By operating the algorithm in linear mode, the CORDIC computations can be modeled to emulate the squaring function. Our 8-bit CORDIC-based squaring architecture shows a 25% and 38% reduction in PDAP over the existing best design for ASIC and FPGA platforms, respectively.**

Index Terms— **CORDIC algorithm, DSP, Fixed-point arithmetic, Pipelining, Vedic Mathematics.**

Original Research Paper
DOI: 10.53314/ELS2529053K

I. Introduction

Multiplication based functions like Multiply-Add (MAD), Multiply-Accumulate (MAC), Convolution, etc., form the core of many digital signal processing (DSP) applications [1], [2]. Image Processing, Neural Networks, Audio-Video Processing, and Machine Learning are a few of the many areas that have multiplication-inspired computations as an integral part of their functionality. The squaring function is one such operation that finds its application in many signal-processing tasks [3], [4]. The squaring operation can be seen as a specific instance of multiplication where the multiplicand and multiplier have the same value [5]. Evidently, most of the existing work related to squaring architectures is actually based on the optimization of the multiplication operation for the specific case of identical operands [6]. The performance bottleneck for a multiplier unit is typically limited by the partial product reduction (PPR) stage [7]. Traditionally, Wallace and Dadda trees have been used to reduce the partial product matrix (PPM) [8], [9]. Numerous modifications to these approaches have been proposed. Prominent among them are the approximation techniques that focus on designing compressors that reduce the complexity of the PPR stage. A 4:2 compressor-based multiplier based on the correlation of input data is presented in [10].

Approximate compressor trees with reduced accumulation layers are presented in [11], [12]. An approximate multiplier based on an inexact 4:2 compressor that shares the logic required for sum and carry is presented in [13]. Similarly, 4:2 compressor-based multipliers based on a constant approximation and probability-based error correction are presented in [14], [15]. Recursive approximate multipliers based on the divide and conquer strategy are presented in [16]. These show substantial improvements in error metrics when compared to traditional compressor-based multipliers. For field programmable gate array (FPGA) based designs, generalized parallel counters (GPC) are used to construct the compressor tress [17]-[19]. These map well on FPGAs and are often used to replace traditional Wallace and Dadda trees. However, given that squaring is a specific type of multiplication where both operands are identical, the number of partial products generated is generally fewer [5]. This reduces the computational complexity of the PPR stage. Evidently, a squaring architecture is often much faster and consumes fewer resources compared to an equivalent multiplier [20].

Recently, Vedic mathematics has been used to improve the performance of multiplier architectures in general [21], [22] and squaring architectures in specific [5], [6], [20], [23], [24]. In [23], a squaring architecture based on the duplex property of *Dwandwayoga* sutra is proposed and implemented using 14nm FinFET technology. A squaring architecture based on the *Urdhva Tiryagbhyam* sutra from Vedic mathematics is proposed in [24]. The concept is further modulated in [5], wherein the authors use a combination of *Urdhva Tiryagbhyam* sutra and *Karatsuba-Ofman* algorithm to design a recursive squaring architecture. In [20], the authors present a square architecture based on the *Anurupye* sutra of Vedic mathematics. The architecture is implemented on Kintex-7 FPGA and reports a subsequent improvement in performance over some earlier Vedic-based designs [24], [25]. However, it fails to match the performance of the design reported in [5]. The authors in [26] report a squaring architecture based on the *Ekadhikena Purvena* sutra of Vedic mathematics. The resulting structure is multiplier-less but fails to match the performance of designs in [5] and [20]. Similar squaring architectures based on the *Yavadunam* sutra of Vedic mathematics are reported in [6], [27].

Manuscript received on August 22nd, 2024. Received in revised form on January 6th and February 11th, 2025. Accepted for publication on March 10th, 2025.

Burhan Khurshid is with the Department of Electronics & Communication Engineering, National Institute of Technology Srinagar, J&K, India (e-mail: burhan@nitsri.ac.in).

This paper proposes a novel approximate squaring architecture based on the CORDIC algorithm. In the past, CORDIC-based computations have been employed to calculate various trigonometric and non-linear transcendental functions. We operate the CORDIC algorithm in linear mode and modify the computations to evaluate the squaring operation. We also conduct a Pareto analysis for an 8-bit CORDIC-based squaring architecture to determine the optimum number of stages that justify the accuracy-performance trade-offs for the proposed architecture.

The paper proceeds with the following structure: Section II briefly discusses the basics of the CORDIC algorithm. Squaring architecture based on CORDIC is presented in Section III. Error analysis is done in Section IV. Synthesis, implementation, and performance comparison are carried out in Section V. Section VI concludes the paper and points out any scope for future work. References are listed at the end.

II. Cordic Algorithm

CORDIC stands for Coordinate Rotation Digital Computer. The algorithm iteratively rotates a vector through arbitrary angles within linear or non-linear coordinate systems [28]-[30]. The basic rotation equations are given as [31]:

$$\begin{bmatrix}\mathbf{x}_m\\ \mathbf{y}_m\end{bmatrix} = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\mathbf{x}_0\\ \mathbf{y}_0\end{bmatrix} \tag{1}$$

($\mathbf{x}_0$, $\mathbf{y}_0$) are the starting coordinates of the vector, and ($\mathbf{x}_m$, $\mathbf{y}_m$) are the final coordinates of the vector. Since its induction, different algorithm modifications have been proposed to enable the algorithm to calculate a wider range of mathematical functions [32]. The basic operating modes of the algorithm are the rotation and the vectoring mode. While the former is characterized by the reduction of the value of the residual angle with every iteration [33], the latter performs a finite number of micro-rotations to compute the angle that aligns the final vector parallel to the x-axis. A single set of iterative unified CORDIC equations that encapsulate both modes is given as [34]:

$$\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{y}_{i+1}\end{bmatrix} = \mathbf{k}_i\begin{bmatrix}1 & -\mu\boldsymbol{\sigma}_i2^{-i}\\ \boldsymbol{\sigma}_i2^{-i} & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix} \tag{2}$$

$$\mathbf{z}_{i+1} = \begin{cases}\mathbf{z}_i - \boldsymbol{\sigma}_i\tanh^{-1}(2^{-i}), & \mu = -1\\ \mathbf{z}_i - \boldsymbol{\sigma}_i(2^{-i}), & \mu = 0\\ \mathbf{z}_i - \boldsymbol{\sigma}_i\tan^{-1}(2^{-i}), & \mu = 1\end{cases} \tag{3}$$

Where μ = -1, 0, and 1 for hyperbolic, linear, and circular coordinates, respectively.

$$\boldsymbol{\sigma}_i = \begin{matrix}\text{sign}(\mathbf{z}_i), \text{for rotation mode}\\ -\text{sign}(\mathbf{y}_i), \text{for vectoring mode}\end{matrix}$$

i and *n* represent the current and the total number of iterations, respectively. The rotations are pseudo rotations and tend to increase the length of the rotating vector with every iteration. The gain factor $\boldsymbol{k}_i$ is, therefore, used to maintain the original length of the rotating vector. k_i is calculated as:

$$\mathbf{k}_i = \prod_{i=0}^{n-1}\sqrt{1+\mu2^{-2i}} \tag{4}$$

III. Proposed Squaring Architecture

CORDIC computations can be modeled to emulate the squaring operation by performing rotations in a linear coordinate system. From equations (2) and (3), we have:

$$\mathbf{x}_{i+1} = \mathbf{k}_i[\mathbf{x}_i - \mu\boldsymbol{\sigma}_i2^{-i}\mathbf{y}_i]$$

$$\mathbf{y}_{i+1} = \mathbf{k}_i[\boldsymbol{\sigma}_i2^{-i}\mathbf{x}_i + \mathbf{y}_i]$$

$$\mathbf{z}_{i+1} = \mathbf{z}_i - \boldsymbol{\sigma}_i2^{-i}$$

For a linear coordinate system, μ = 0. The value of the gain factor $\boldsymbol{k}_i$ will, therefore, be unity. The above equations will be modified as:

$$\mathbf{x}_{i+1} = \mathbf{x}_i$$

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \boldsymbol{\sigma}_i2^{-i}\mathbf{x}_i$$

$$\mathbf{z}_{i+1} = \mathbf{z}_i - \boldsymbol{\sigma}_i2^{-i}$$

Or

$$\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{y}_{i+1}\end{bmatrix} = \begin{bmatrix}1 & 0\\ \boldsymbol{\sigma}_i2^{-i} & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix} \tag{5}$$

$$\mathbf{z}_{i+1} = \mathbf{z}_i - \boldsymbol{\sigma}_i2^{-i} \tag{6}$$

The above equations are iterative. Note that depending upon the direction of rotation $\boldsymbol{\sigma}_i$ may be either +1 or -1. If the initial values of **x**, **y**, and **z** vectors are $\mathbf{x}_s$, $\mathbf{y}_s$, and $\mathbf{z}_s$, respectively, then the equations after the first iteration (i = 0) would be:

$$\mathbf{x}_1 = \mathbf{x}_s$$

$$\mathbf{y}_1 = \mathbf{y}_s \pm \mathbf{x}_s$$

$$\mathbf{z}_1 = \mathbf{z}_s \mp 1$$

After second iteration (i = 1):

$$\mathbf{x}_2 = \mathbf{x}_1 = \mathbf{x}_s$$

$$\begin{aligned}\mathbf{y}_2 &= \mathbf{y}_1 \pm 2^{-1}\mathbf{x}_1\\ &= \mathbf{y}_s \pm \mathbf{x}_s \pm 2^{-1}\mathbf{x}_s\end{aligned}$$

$$\begin{aligned}\mathbf{z}_2 &= \mathbf{z}_1 \mp 2^{-1}\\ &= \mathbf{z}_s \mp 1 \mp 2^{-1}\end{aligned}$$

After third iteration (i = 2):

$$\mathbf{x}_3 = \mathbf{x}_2 = \mathbf{x}_1 = \mathbf{x}_s$$

$$\begin{aligned}\mathbf{y}_3 &= \mathbf{y}_2 \pm 2^{-2}\mathbf{x}_2\\ &= \mathbf{y}_s \pm \mathbf{x}_s \pm 2^{-1}\mathbf{x}_s \pm 2^{-2}\mathbf{x}_s\end{aligned}$$

$$\begin{aligned}\mathbf{z}_3 &= \mathbf{z}_2 \mp 2^{-2}\\ &= \mathbf{z}_s \mp 1 \mp 2^{-1} \mp 2^{-2}\end{aligned}$$

After *m* iterations:

$$\mathbf{x}_m = \mathbf{x}_{m-1} \dots = \mathbf{x}_2 = \mathbf{x}_1 = \mathbf{x}_s$$

$$\begin{aligned}\mathbf{y}_m &= \mathbf{y}_{m-1} \pm 2^{-m+1}\mathbf{x}_{m-1}\\ &= \mathbf{y}_s \pm \mathbf{x}_s \pm 2^{-1}\mathbf{x}_s \pm 2^{-2}\mathbf{x}_s \pm \cdots \pm 2^{-m+1}\mathbf{x}_s\\ &= \mathbf{y}_s \pm \mathbf{x}_s[1 \pm 2^{-1} \pm 2^{-2} \pm \cdots \pm 2^{-m+1}]\end{aligned}$$

$$\mathbf{z}_m = \mathbf{z}_{m-1} \mp 2^{-m+1}$$

$$= \mathbf{z_s} \mp 1 \mp 2^{-1} \mp 2^{-2} \mp \cdots \mp 2^{-m+1}$$

Since the algorithm is being operated in rotation mode after m iterations, the value of the residual angle ($\mathbf{z}_m$) will be extremely small and may be equated to 0. Therefore, after *m* iterations:

$$\mathbf{z_m} = 0$$

$$\mathbf{z_s} \mp 1 \mp 2^{-1} \mp 2^{-2} \mp \cdots \mp 2^{-m+1} = 0$$

Or

$$\mathbf{z_s} = 1 \pm 2^{-1} \pm 2^{-2} \pm \cdots \pm 2^{-m+1}$$

Or

$$\mathbf{y_m} = \mathbf{y_s} \pm \mathbf{x_s z_s}$$

The final equations after *m* iterations would, therefore, be:

$$\mathbf{x_m} = \mathbf{x_s} \tag{7}$$

$$\mathbf{y_m} = \mathbf{x_s z_s} + \mathbf{y_s} \tag{8}$$

$$\mathbf{z_m} = 0 \tag{9}$$

Equivalently, the above set of equations may be written as:

$$\begin{bmatrix} \mathbf{x_m} \\ \mathbf{y_m} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \mathbf{z_s} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x_s} \\ \mathbf{y_s} \end{bmatrix} \tag{10}$$

$$\mathbf{z_m} = 0 \tag{11}$$

In equation (8), if $\mathbf{x}_s = \mathbf{z}_s$, and $\mathbf{y}_s = 0$, then $\mathbf{y}_m$ will compute the square of the input $\mathbf{x}_s$. A direct mapping of the set of equations in (5) and (6) will result in an iterative (folded) architecture, as shown in Fig. 1. The iterative architecture is sequential and incurs a lot of latency, which is unsuitable for high-speed DSP applications. The iterative architecture can be unfolded to map each iteration on a separate computing stage. This results in a parallel structure, as shown in Fig. 2. The parallel architecture can be easily pipelined by placing registers after every computation stage. Thus, a sequential iterative architecture is converted into a combinational (pipelined) feed-forward architecture.

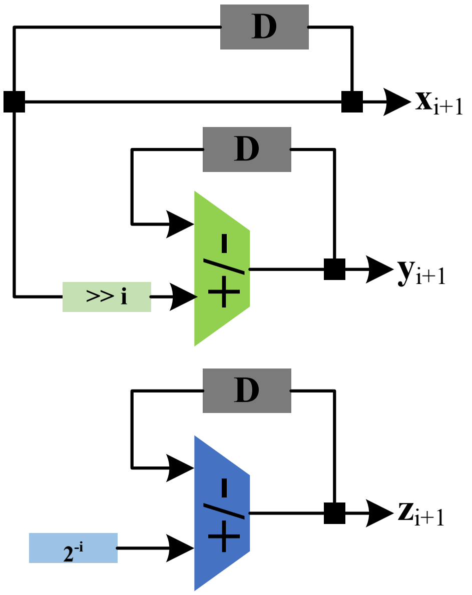


Fig. 1. Sequential CORDIC-based Square architecture

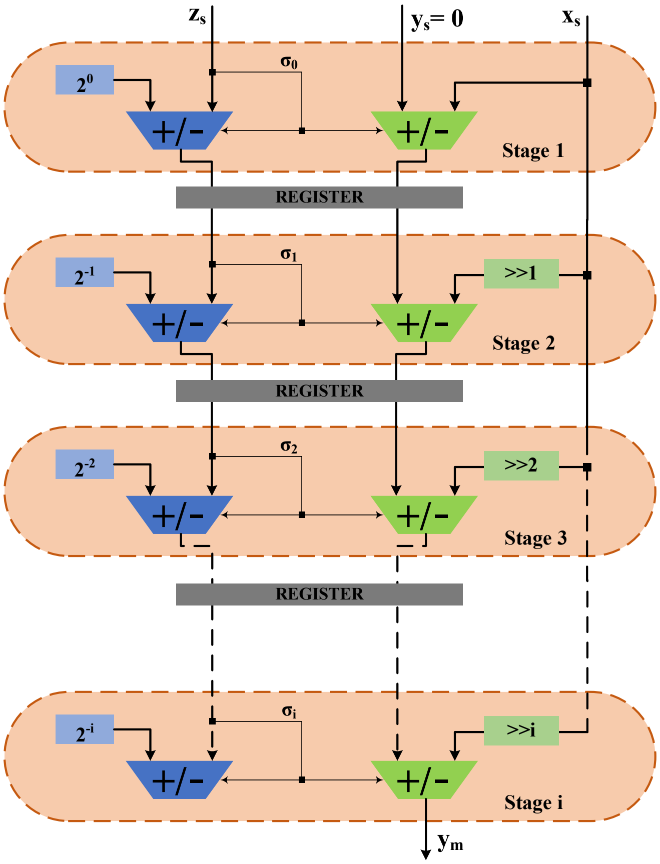


Fig. 2. Unfolded CORDIC-based Square architecture

The square architecture of Fig. 2 is based on simple shift and add operations. For larger magnitude operands, the algorithm will require a lot of stages to converge to an acceptable level of accuracy. This will incur a lot of hardware costs and a degradation in performance and accuracy. To avoid this, the input is scaled down such that its value lies in the range [0,1). This is done by scaling the **x** vector by 2^n. The new input is thus represented in fixed-point 2's complement format as:

$$\mathbf{x'_s} = \mathbf{x'_n} \,.\, \mathbf{x'_{n-1}}\ \mathbf{x'_{n-2}}\ \mathbf{x'_{n-3}} \ldots \ldots\ \mathbf{x'_1}\ \mathbf{x'_0} \tag{12}$$

The value of the input is given as:

$$\mathbf{x'_s} = \textstyle\sum_{i=1}^{n} \mathbf{x'_{n-i}} 2^{-i} \tag{13}$$

The result is then obtained by scaling the output from the last stage by 2^{2n}. Note that CORDIC-based computations are only approximate. The number of iterations will, therefore, determine the accuracy of the final result. We, therefore, conducted a detailed Pareto analysis to determine the number of optimal stages that will justify the accuracy-performance trade-offs. Our analysis considers the 8-bit CORDIC square architecture. The added advantage of similar inputs in the squaring operation lowers the permutations that can exist for the inputs. A total of 256 input combinations exist for an 8-bit square architecture, and all such combinations are considered Pareto points. The Pareto analysis focuses on the variation of error with the number of computation stages. Specifically, we have used error rate (ER) and mean error distance (MED) to quantify the error in our computations. ER is defined as the percentage of outputs in error. MED is defined as the average of error distance (ED) values for a set of input-output samples. Mathematically,

$$\mathrm{MED} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}|\mathrm{ED}_{\mathrm{i}}| \tag{14}$$

Where N is the number of patterns. The results of the Pareto analysis are presented in Figs. 3 and 4, where the MED and ER for all the input combinations are plotted as a function of the number of CORDIC stages, respectively. A MED of 7.8 is achieved with a CORDIC square architecture designed in 8 stages; thereafter, any increase in the number of stages does not result in any significant reduction in MED. In fact, as the number of stages is further increased, the correctly obtained outputs are subjected to further computations, thereby introducing errors in the results. This is evident from Fig. 3, where the MED slightly increases after 10 stages due to over-computation of the results. Similarly, an ER of 5% is achieved with a CORDIC square architecture designed in 8 stages, which signifies that 95% of results are accurate. Based on the fixed-point representation of equation (12) and the results from the Pareto analysis, an illustration of the stage-wise computations for the CORDIC squarer is presented in Table I. The modified square structure is shown in Fig. 5.

TABLE I
STAGE-WISE COMPUTATIONS FOR AN 8-BIT CORDIC SQUARE ARCHITECTURE.

Initial Inputs	$= \mathbf{01101110_2}$ $= \mathbf{110_{10}}$	$= \mathbf{00000000_2}$ $= \mathbf{0_{10}}$	$= \mathbf{01101110_2}$ $= \mathbf{110_{10}}$
Inputs and scaled down by 2^8	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.00000000_2$ $= 0.0_{10}$	$= 0.01101110_2$ $= 0.4296875_{10}$
1st Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.00000000_2 + 0.01101110_2$ $= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.01110001_2 - 1.00000000_2$ $= 1.01101110_2$ $= -0.570313_{10}$
2nd Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.011011100_2 - 0.001101110_2$ $= 0.001101110_2$ $= 0.21484375_{10}$	$= 1.01101110_2 + 0.10000000_2$ $= 1.11101110_2$ $= -0.070313_{10}$
3rd Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.0011011100_2 - 0.0001101110_2$ $= 0.0001101110_2$ $= 0.107421875_{10}$	$= 1.11101110_2 + 0.01000000_2$ $= 0.00101101_2$ $= 0.179687_{10}$
4th Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.00011011100_2 + 0.00001101110_2$ $= 0.00101001010_2$ $= 0.1611328125_{10}$	$= 0.00101101_2 - 0.00100000_2$ $= 0.00001101_2$ $= 0.054687_{10}$
5th Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.001010010100_2 + 0.000001101110_2$ $= 0.001100000010_2$ $= 0.18798828125_{10}$	$= 0.00001101_2 - 0.00010000_2$ $= 0.00000010_2$ $= -0.007813_{10}$
6th Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.0011000000100_2 - 0.0000001101110_2$ $= 0.0010110010110_2$ $= 0.174560546875_{10}$	$= 0.00000010_2 + 0.00001000_2$ $= 0.00000101_2$ $= 0.023437_{10}$
7th Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.00101100101100_2 + 0.00000001101110_2$ $= 0.00101110011010_2$ $= 0.1812744140625_{10}$	$= 0.00000101_2 - 0.00000100_2$ $= 0.00000001_2$ $= 0.007812_{10}$
8th Stage	$= 0.01101110_2$ $= 0.4296875_{10}$	$= 0.001011100110100_2 + 0.000000001101110_2$ $= 0.001011110100010_2$ $= 0.18463134765625_{10}$	$= 0.00000001_2 - 0.00000010_2$ $= 1.11111111_2$ $= -0.0000005_{10}$
Multiplier Output after 8 stages (Bit adjusted and scaled up by 2^{16})		$= \mathbf{0001011110100010 0_2}$ $= \mathbf{12100_{10}}$	

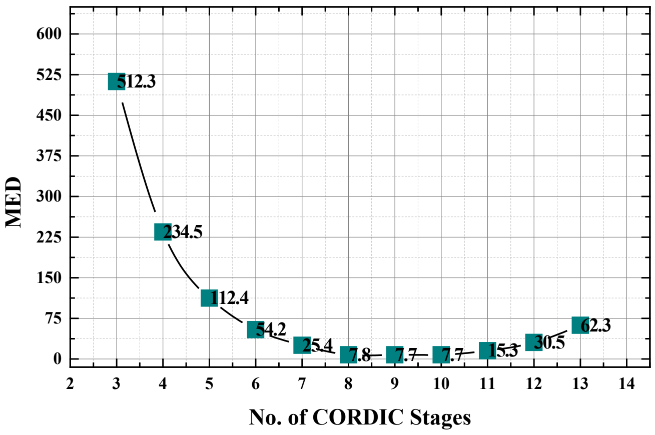


Fig. 3. Variation of MED with the number of computation stages for CORDIC-based 8-bit Square architecture.

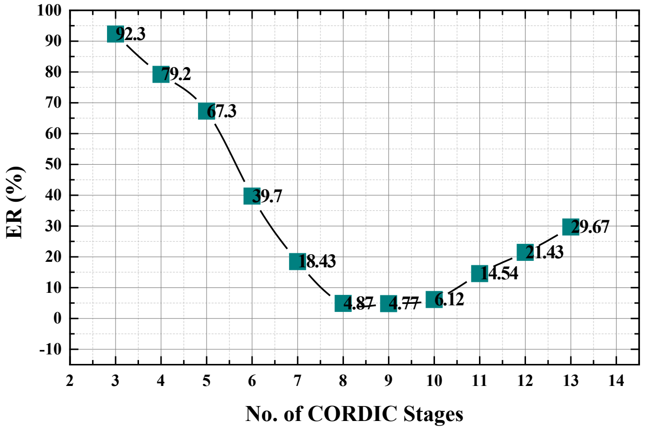


Fig. 4. Variation of ER with the number of computation stages for CORDIC-based 8-bit Square architecture.

IV. Error Analysis

The accuracy of the proposed architecture is compared against some approximate multiplier-based square architectures in terms of MED and normalized mean error distance (NMED). MED is given by equation (14). NMED is computed from MED as:

$$\text{NMED} = \frac{\text{MED}}{\text{R}_{\text{max}}} \tag{15}$$

Where R_{max} is the maximum exact result of the computation. The error metrics are computed using the Vivado 23.1 simulator. The input patterns are provided as an input text file, and the calculated output of different approximate designs is also written to a separate text file. The computed outputs are then compared to the exact outputs for each pattern, and ED for each pattern is calculated. MED is then computed as the average of EDs. Table II provides a comparison of different 8-bit square architectures. It is observed that the CORDIC-based square architecture has the least MED and NMED among the various designs.

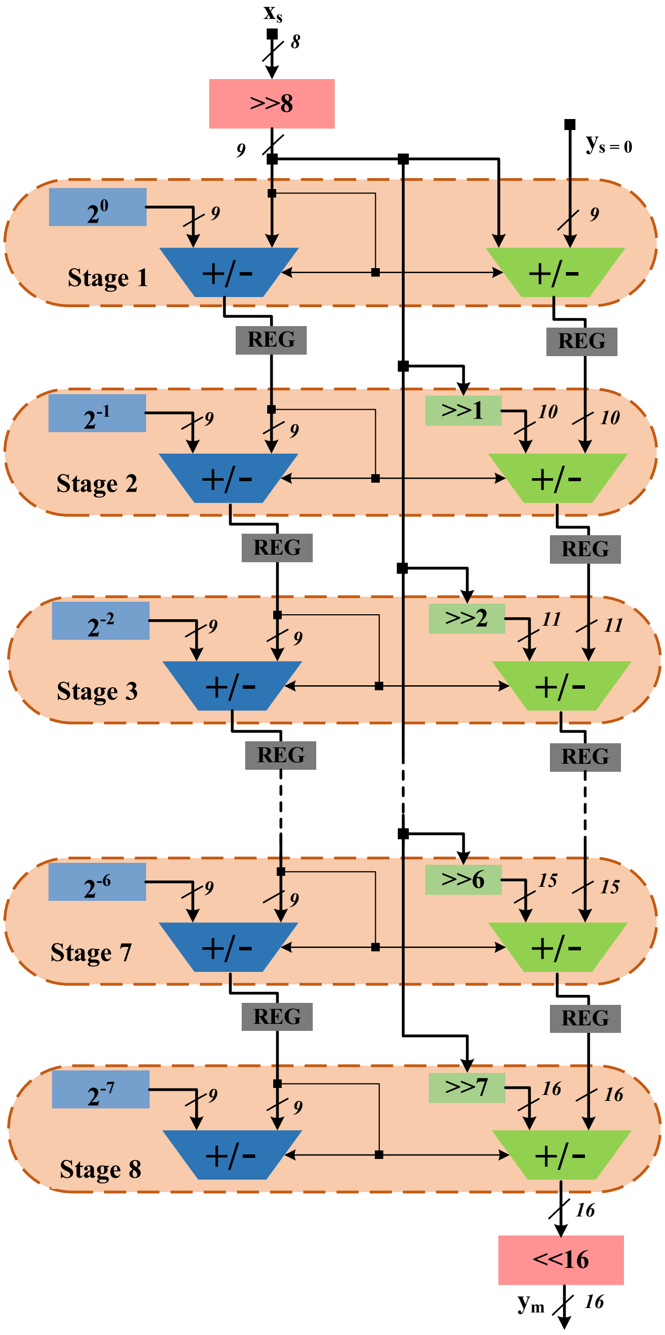


Fig. 5. CORDIC-based Square architecture with scaled inputs

TABLE II
Accuracy Analysis of Different Square Architectures for 8-bit Operands

Design	MED	NMED
[11]	21.8	0.00034
MCom 4:2 [12]	204.1	0.00314
MFA1 [12]	67.45	0.00104
MFA2 [12]	45.12	0.0007
[13]	137.3	0.0021
Normal [14]	16.4	0.00025
Hybrid [14]	11.3	0.00017
MUL1 [15]	121.32	0.00186
MUL2 [15]	115.7	0.00178
RRAM-I [16]	72.8	0.00112
RRAM-II [16]	9.32	0.00014
Proposed	7.8	0.00012

V. Synthesis & Implementation

The proposed square architecture is implemented both on ASIC and FPGA platforms. The ASIC implementation is based on the TSMC 180 nm CMOS technology using the Synopsys design tool. The simulations are conducted at an operating voltage of 1.8 V with an operating temperature of 25º C and a target clock frequency of 100 MHz. The area, critical path delay, and dynamic power dissipation are reported for each design. For FPGA implementation, the architecture in Fig. 5 is coded using VHDL and implemented using Xilinx Vivado 23.1, targeting the 7th generation xc7k325tffg900c FPGA device from the Kintex-7 family. The analysis involves resources, timing, and dynamic power dissipation as parameters of interest. Implementation is done under a constrained environment for both ASIC and FPGA platforms, with both physical and timing constraints being duly provided to emulate the actual operating conditions. The metrics are reported after the complete placement and routing of the design. Table III compares the performance of the proposed square architecture against different exact and approximate square architectures reported in the literature. The analysis is done for an operand word length of eight bits. The analysis reveals that the CORDIC-based square architecture shows an improved performance when compared to the existing exact and approximate designs. For ASIC implementation, a few designs report slightly better metrics than our proposed architecture. However, our design reports the best metrics for FPGA platforms, with only MUL1-based square architecture reported in [15] showing reduced LUT count and dynamic power dissipation. However, the square architecture based on the MUL1 design is a serial architecture and has an extremely large critical path delay. To get a better perspective of the performance improvement, Table IV shows the power-delay-area product (PDAP) for different designs on ASIC and FPGA platforms. Our design reports a 25% and 38% improvement over the next best design for ASIC and FPGA platforms, respectively. Due to the rigorous pipelining, our design implementation requires more registers than the other reported. However, each logic slice in Kintex-7 FPGAs supports eight flip-flop registers; thus, using these registers does not incur any extra hardware cost. Including these registers in the implementation process considerably reduces the critical paths. This has two advantages; first, the structure can be clocked at higher frequencies, enhancing speed and throughput. Second, the capacitances to be charged and discharged in a single clock cycle are reduced, reducing dynamic power dissipation. Further analysis focuses on the achievable accuracy versus performance trade-offs using the proposed square architecture. This is shown in Figs. 6 and 7, where power-delay-product (PDP) and resource usage are plotted against the NMED for different exact and approximate square architectures for FPGA and ASIC platforms, respectively. For a given NMED, our proposed square architecture has the least PDP and utilizes the least resources. Similar results are reflected in Figs. 8 and 9, plotting PDAP against the NMED for

TABLE III
Performance Analysis of Different 8-bit Square Architectures on ASIC and FPGA Platforms

Design	ASIC (180 nm CMOS)			FPGA (xc7k325tffg900c)		
	Area (μm^2)	Delay (ns)	Power (mW)	LUT	Delay (ns)	Power (mW)
Accurate						
[05]	1910.32	2.34	0.062	56	4.879	28.2
[20]	2115.44	2.65	0.067	62	8.6	29
[23]	1978.96	2.44	0.062	58	5.6	29
[24]	4503.84	2.98	0.075	132	9.8	28.2
[25]	3616.72	2.71	0.071	106	9.4	28.2
[26]	3343.76	2.66	0.071	98	8.9	20
[27]	2653.21	2.93	0.069	68	9.11	21
Approximate						
[11]	3421.12	1.612	0.066	59	4.1	21.33
MCom 4:2 [12]	2114.32	1.495	0.052	48	3.4	23.43
MFA1 [12]	3461.65	1.567	0.063	64	4.02	21.88
MFA2 [12]	4094.22	1.588	0.071	66	4.44	22.11
[13]	2312.16	1.441	0.0591	52	3.33	20.67
Normal [14]	2117.65	1.431	0.0583	50	3.76	21.56
Hybrid [14]	2216.71	1.442	0.0586	52	3.81	21.6
MUL1 [15]	1412.32	3.112	0.0411	32	7.8	18.75
MUL2 [15]	1721.11	3.211	0.0431	39	8.01	18.89
RRAM-I [16]	2410.61	1.621	0.056	53	3.01	20.87
RRAM-II [16]	2441.11	1.562	0.0552	54	3.334	20.81
Proposed	1522.01	1.44	0.0561	36	2.929	19.43

different exact and approximate square architectures for FPGA and ASIC platforms, respectively.

The proposed square architecture is also tested by using it to increase the intensity of the output pixels in an image edge-detection application, thereby leading to better contrast enhancement. Edge detection is performed by applying Gaussian noise filtering followed by horizontal and vertical filters to detect the edges. The proposed square architecture is then used to improve the contrast of the final filtered image by multiplying it with itself. The quality of the output image is reported in terms of the peak signal-to-noise ratio (PSNR) calculated with respect to the image obtained using the exact square architecture. Fig. 10 shows the output images obtained using different square architectures. Our proposed square architectures report an improvement of 20% in PSNR over the next best architecture.

TABLE IV
PDAP Analysis of Different 8-bit Square Architectures on ASIC and FPGA Platforms

Design	ASIC (180 nm CMOS)	FPGA (xc7k325tffg900c)
	PDAP (fJm^2)	PDAP (nJA)
[05]	0.277149	7.704917
[20]	0.375596	15.4628
[23]	0.299377	9.4192
[24]	1.006608	36.47952
[25]	0.695893	28.09848
[26]	0.631503	17.444
[27]	0.5363	13.009
[11]	0.36398	5.159727
MCom 4:2 [12]	0.164367	3.823776
MFA1 [12]	0.341738	5.629286
MFA2 [12]	0.461615	6.479114
[13]	0.196911	3.579217
Normal [14]	0.17667	4.05328
Hybrid [14]	0.187315	4.279392
MUL1 [15]	0.18064	4.68
MUL2 [15]	0.238191	5.901047
RRAM-I [16]	0.218826	3.329391
RRAM-II [16]	0.210478	3.746549
Proposed	0.122954	2.048777

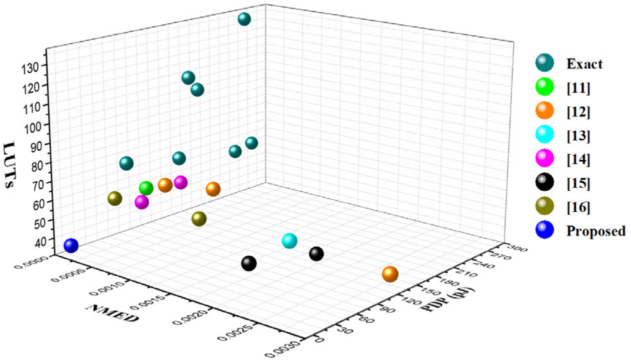


Fig. 6. PDP-LUT-NMED plot for different Square architectures implemented on the FPGA platform.

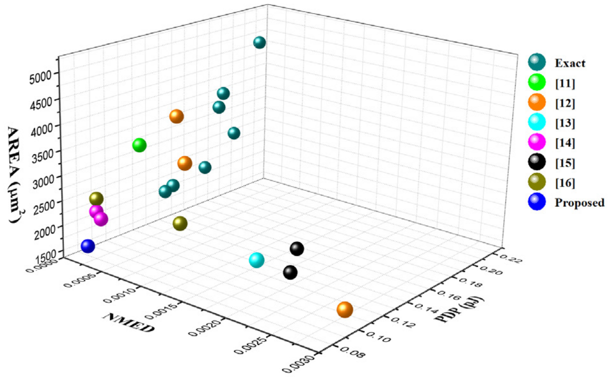


Fig. 7. PDP-AREA-NMED plot for different Square architectures implemented on the ASIC platform.

V. CONCLUSION

In this paper, we propose a square architecture based on the operation of the CORDIC algorithm in a linear coordinate system. While CORDIC has frequently been used in literature to evaluate different trigonometric and transcendental functions, it is very rarely used to evaluate linear functions. Being hardware-efficient, the computations are simple, enabling the inherent algorithm to be translated into diverse architectures to suit the application demands. In this paper, we focussed on unfolded (pipelined) architectures. Our analysis with ASIC and 7th-generation FPGAs reported a substantial performance improvement evaluated in terms of PDAP. Further, the accuracy-performance trade-offs achievable with our proposed square architecture outperform all the existing approximate multiplier-based square architectures. Heuristically, the convergence of the architecture shares a linear relationship with the operand word length. For large operand word-lengths, therefore, there will be an exponential increase in the resource utilized. Our future endeavors will focus on speeding up the convergence of the computations through the use of radix-4 arithmetic. This will reduce the number of iterative stages, resulting in lesser LUT utilization.

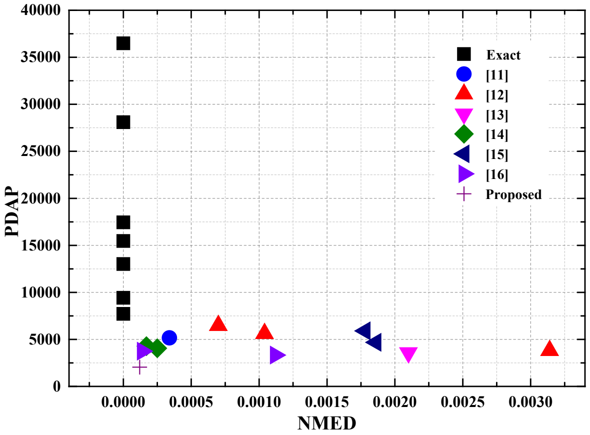


Fig. 8. PDAP-NMED plot for different Square architectures implemented on the FPGA platform.

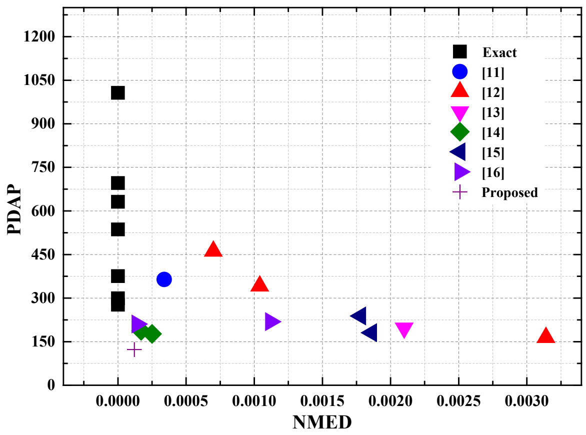


Fig. 9. PDAP-NMED plot for different Square architectures implemented on the ASIC platform.

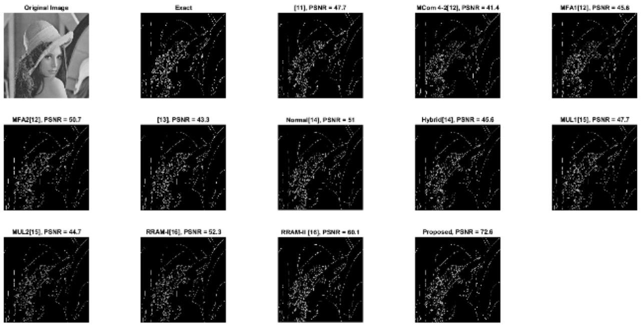


Fig. 10. Contrast enhancement using different square architectures for an edge-detection application.

References

[1] P. Jebashini, R. Uma, P. Dhavachelvan and H. K. Wye, "A Survey and Comparative Analysis of Multiply-Accumulate (MAC) block for Digital Signal Processing Application on ASIC and FPGA," *Journal of Applied Sciences,* July 2015; 15(7): 934-946, doi:10.3923/jas.2015.934.946.

[2] A. Deepa and C. N. Marimuthu, "A High Speed VLSI Architecture of a Pipelined Reed Solomon Encoder for Data Storage in Communication Systems," *Asian Journal of Research in Social Sciences and Humanities,* Feb. 2017; 7(2): 228–238, doi:10.5958/2249-7315.2017.00085.5.

[3] J. Pihl and E. J. Aas, "A multiplier and squarer generator for high-performance DSP applications," in *Proceedings of the 39th Midwest Symposium on Circuits and Systems*, Ames, IA, USA, 1996, pp. 109-112, doi:10.1109/MWSCAS.1996.594049.

[4] M. Kaveh, M. Khishe and M. R. Mosavi, "Design and implementation of a neighborhood search biogeography-based optimization trainer for classifying sonar dataset using multi-layer perceptron neural network," *Analog Integrated Circuits and Signal Processing* Nov. 2018; 100: 405-428, doi: 10.1007/s10470-018-1366-3.

[5] J. Bajaj and B. Jajodia, "Efficient Hardware Implementation of High-Speed Recursive Vedic Squaring Architecture on FPGA," in *Proceedings of the International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Cape Town, South Africa, Dec. 2021, pp. 1-6, doi:10.1109/ICECET52533.2021.9698774.

[6] A. Deepa and C. N. Marimuthu, "Design of a high speed Vedic multiplier and square architecture based on *Yavadunam Sutra*." *Sadhana*, Indian Academy of Sciences Aug. 2019; 44:197: 1–10, doi:0.1007/s12046-019-1180-3.

[7] P. J. Edavoor, S. Raveendran and A. D. Rahulkar, "Approximate Multiplier Design using Novel Dual-Stage 4:2 Compressors," *IEEE Access*, Mar. 2020; 8: 48337-51, doi: 10.1109/ACCESS.2020.2978773.

[8] S. Asif and Y. Kong, "Design of an Algorithmic Wallace Multiplier using High-Speed Counters," in *Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES)*, Cairo, Egypt; Dec. 2015, pp. 133-138, doi: 10.1109/ICCES.2015.7393033.

[9] W. J. Townsend, E. E. Swartzlander and J. A. Abraham, "A Comparison of Dadda and Wallace Multiplier Delays," in *Proc. SPIE Annual meeting Opt. Sci. Technol.,* San Diego, CA, USA, Dec. 2003, pp. 552-560, doi: 10.1117/12.507012.

[10] H. Xiao, H. Xu, X. Chen, Y. Wang, Y. Han, "Fast and high-accuracy approximate MAC unit design for CNN computing," *IEEE Embedded Systems Letters* 14 (3) September 2022 155–158, doi: 10.1109/LES.2021.3137335.

[11] T. Yang, T. Ukezono, T. Sato, "Low-power and high-speed approximate multiplier design with a tree compressor," in: *Proceedings of the 35th International Conference on Computer Design (ICCD),* Boston, MA, USA, November 2017, pp. 89–96, doi:10.1109/ICCD.2017.22.

[12] B. Rashidi, "Efficient and low-cost approximate multipliers for image processing," *Integration, the VLSI Journal* Jan. 24; 94(102084), pp. 1-13, doi: 10.1016/j.vlsi.2023.102084.

[13] S. K. Beura, B. P. Devi, P. K. Saha and P. K. Meher, "Design of a Novel Inexact 4:2 Compressor and Its Placement in the Partial Product Array for Area, Delay, and Power-Efficient Approximate Multipliers," *Circuits, Systems, and Signal Processing* March 2024 43: 3748-3774, doi: 10.1007/s00034-024-02630-4.

[14] M. Zhang, S. Nishizawa and S. Kimura, "Area Efficient Approximate 4-2 Compressor and Probability-Based Error Adjustment for Approximate Multiplier," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, Vol. 70, No. 5, May 2023, doi: 10.1109/TCSII.2023.3257852.

[15] L. Sayadi, S. Timarachi, A.S. Akbari, Two efficient approximate unsigned multipliers by developing new configuration for approximate 4:2 compressors, IEEE Transactions on Circuits and Systems-I Feb 70 (4) (2023) 1649–1659, doi: 10.1109/TCSI.2023.3242558.

[16] V. Tammineni, S. K. Beura, M. V. H. B. Murthy, S. Majumdar and P. Saha, "Optimized recursive approximate multipliers for edge detection and image smoothing applications," *Microsystem Technologies*, Nov. 2024, doi: 10.1007/s00542-024-05810-z.

[17] T. Matsunaga, S. Kimura and Y. Matsunaga, "Multi-Operand Adder Synthesis on FPGAs using Generallized Parallel Counters," in *Proc. of 15th Asia and South Pacific Design Automation Conference (ASP-DAC),* Taipei, Taiwan, Jan 2010, pp. 337-342, doi: 10.1109/ASPDAC.2010.5419871.

[18] H. P. Afshar, P. Brisk and P. Ienne, "Efficient Synthesis of Compressor Trees on FPGAs," in *Proc. of 2008 Asia and South Pacific Design Automation Conference,* Seoul, South Korea, Mar. 2008, pp. 138-143, doi: 10.1109/ASPDAC.2008.4483927.

[19] H. P. Afshar, A. Neogy, P. Brisk and P. Ienne, "Compressor Tree Synthesis on Commercial High-Performance FPGAs," *ACM Transactions on Reconfigurable Technology and Systems*, Dec. 2011; 4(4): 1-19, doi: 10.1145/2068716.2068725.

[20] B. N. K. Reddy, "Design and implementation of high performance and area efficient square architecture using Vedic Mathematics," *Analog Integrated Circuits and Signal Processing,* July 2019; 102: 501-506, doi: 10.1007/s10470-019-01496-w.

[21] P. Saritha, J. Vinitha, S. Sravya, V. Vijay and E. Mahesh, "4-Bit Vedic Multiplier with 18 nm FinFET Technology," in *Proceedings of the International Conference on Electronics and Sustainable Communication System (ICESC)*, Coimbatore, India, 2020, pp. 1079-84, doi: 10.1109/ICESC48915.2020.9155707.

[22] V. Bianchi V and I. D. Munari, "A modular Vedic multiplier architecture for model-based design and deployment on FPGA platforms," *Microprocessors and Microsystems,* July 2020; 76: 103106: 1-9, doi: 10.1016/j.micpro.2020.103106.

[23] S. Shetkar and S. Kohli, "Squaring Circuit Using 14nmFinFET Technology with Vedic Mathematics Approach." *IETE Journal of Research,* May 2024, 1-9, doi: 10.1080/03772063.2024.2355662.

[24] P. S. Kasliwal, B. P. Patil and D. K. Gautam, "Performance evaluation of squaring operation by Vedic Mathematics," *IETE Journal of Research*, 2011; 57(1), 39-41, doi: 10.4103/0377-2063.78327.

[25] H. Thapliyal, S. Kotiyal and M. B. Srinivas, "Design and analysis of a novel parallel square and cube architecture based on ancient Indian Vedic Mathematics" in *Proceedings of 48th Midwest Symposium on Circuits and Systems,* Covington, KY, USA, Aug. 2005, pp. 1462–1465, doi: 10.1109/MWSCAS.2005.1594388.

[26] D. K. Yadav and R. R. Lal, "Analysis of Vedic Mathematics Ekadhikena Purvena Sutra in Squaring and Multiplication," in *8th International Conference on Communication and Electronics Systems (ICCES 2023)*, Coimbatore, India, June 2023, pp. 12–19, doi: 10.1109/ICCES57224.2023.10192851.

[27] J. Sravana, K. S. Indrani, S. Mahurkar, M. Pranathi, D. R. Reddy and V. Vallabhuni, "Optimized VLSI Design of Squaring Multiplier Using Yavadunam Sutra Through Deficiency Bits Reduction," *Advances in Signal Processing and Communication Engineering. Lecture Notes in Electrical Engineering*, vol 929: 387-399. Springer, Singapore, Dec. 2022, doi: 10.1007/978-981-19-5550-1_36.

[28] B. Khurshid and J. J. Khan, "An Efficient Fixed-Point Multiplier based on CORDIC Algorithm," *Journal of Circuits, Systems and Computers,* 2021; 30(05): 1-19, doi: 10.1142/S0218126621500808.

[29] J. S. Walther, "A Unified Algorithm for Elementary Functions," in *Proceedings of the AFIPS Spring Joint Computer Conference*, New York, USA, May 1971, pp. 379-385, doi: 10.1145/1478786.1478840.

[30] J. S. Walther, "The story of Unified CORDIC," *Journal of VLSI Signal Processing systems for signal, image and video technology,* June 2000; 25(2): 107-112, doi: 10.1023/A:1008162721424.

[31] P. K. Meher, J. Valls, T. B. Juang and K. Sridhavan, "50 years of CORDIC: Algorithms, Architectures and Applications," *IEEE Transactions on Circuits and Systems-I*, 2009; 56(9): 1893-1907, doi: 10.1109/TCSI.2009.2025803.

[32] B. Lakshmi and A. Dhar, "CORDIC Architectures: A Survey," *VLSI Design*, 2010; 794891: 1-19, doi: 10.1155/2010/794891.

[33] S. M. Mohamed, W. S Sayed, A. G. Radwan and L. A. Said, "FPGA Implementation of Reconfigurable CORDIC Algorithm and a Memristive Chaotic System with Transcendental Non-Linearities," *IEEE Transactions on Circuits and Systems-I,* Regular paper, July 2022; 69(7): 2885-92, doi: 10.1109/TCSI.2022.3165469.

[34] G. Raut, S. Rai, S. K. Vishvakarma and A. Kumar, "RECON: Resource-Efficient CORDIC-based Neuron Architecture," *IEEE Open Journal of Circuits and Systems,* Jan 2021; 2: 170-181, doi: 10.1109/OJCAS.2020.3042743.

Variation Tolerant SRAM with Enhanced Stability for Wearable Healthcare Devices

M.Kavitha, S.Ramani, and P.K.Janani

Abstract—**As technology scales down CNTFET (Carbon Nano Tube Field Effect Transistor) circuits has gained importance in VLSI design due to exacerbation of process parameter variations in CMOS. Particularly design of SRAM cell needs more attention as it occupies the larger space in CPU of the battery powered wearable devices. Hence it is a challenging task to the chip designer because the power, energy, speed and stability of the memory cell has a greater impact on system CPU efficiency. A variation tolerant nine transistor CNTFET SRAM cell is proposed in this work. Metrics considered for investigating the proposed SRAM performance is power, delay, power delay product (PDP) and static noise margin (SNM). Stanford University 32 nm CNTFET model and HSPICE tool is utilised for the simulation. In proposed SRAM the read and write power reduction is improved by 4.7x and 9.9x respectively, while the read delay and PDP reduction is improved by 10x than conventional SRAM. The hold, read and write stability of proposed memory cell is enhanced by 1.4x, 1.2x and 4.1x respectively compared to conventional structure.**

Index Terms—**VLSI, SRAM, Low power, Stability, Delay, PDP.**

Original Research Paper
DOI: 10.53314/ELS2529062K

I. Introduction

Embedded memory systems influence the overall performance, efficiency and cost of CPU of modern-day applications. Key design parameters to design an improved SRAM vary according to applications. For modern artificial neural networks, graphics processors, servers etc, high performance is of highest priority than energy [1]. Applications like image processing, Internet of Things (IoT), wearable devices demand reliability, stability and energy efficiency [2]. Wearable devices have occupied a significant place in health monitoring as they help the patients to self-monitor their health conditions. Few examples are wearable fitness trackers and smart health watches, help to track the patient's physical activity and heart rate as they are equipped with sensors synced to their smart phones. Wearable ECG monitors help to measure electrocardiograms and blood pressure and other vitals. The most commonly used device is the wearable blood pressure monitor which, apart from measuring blood pressure, also indicates the calories burnt which help the patients to maintain their health condition under control. Wearable devices in medical field demand ultra-low power and robust SRAM to extend the device battery life and to get reliable results respectively [3]-[5]. CNTFET is the future alternative of MOSFET circuit designs. With respect to carrier mobility, scalability, short channel effects etc, CNTFET performance is better than its CMOS counterpart. Semiconducting channel in MOSFET is replaced with cylinder shaped carbon nano tubes (CNT) tubes. CNT tubes function as semiconductor or metallic depending on the chirality vector. Number of CNT tubes used in the channel region, diameter of CNT tubes, chirality, pitch, gate dielectric, oxide thickness etc. has a greater impact on CNTFET characteristics [6]-[9].

Conventional 6T SRAM has four transistors in cross coupled inverter for storing one bit data and two write access transistors to enable write operation [10]. Read stability of the conventional 6T SRAM is improved by adding a read port. Additional read port provides a separate path for read operation and isolates the storage nodes from bit lines during read operation to increase the read stability [11]. By sharing write bit line and read bit lines in a column, a low power and high stable 9T SRAM can be realized since separate write word lines and drivers improve the write stability, and a dedicated read access transistor along with read control transistor improves the read stability [12]. 9T SRAM with supply feedback minimizes power. A P-type transistor is employed between the supply and load transistors and connecting its gate to the storage node. This feedback makes the cell flip even with strong load transistors by weakening the pull up path [13]. Power gating the SRAM circuit decreases its power consumption and improves stability. Power gating structure puts the SRAM in sleep, drowsy and active mode depending on the control signals [14]-[15].

9T SRAM with read control signal has 6T SRAM cell in upper sub circuit while the lower sub circuit comprises of read circuitry with bit line access transistors along with read access transistor. Read access transistor is enabled by RD signal during read operation and it is disabled during write operation. Due to separate read circuit, the data stability of this SRAM is increased [16]. 9T SRAM with separate read buffer provides read disturb free operation as the read path is decoupled from the storage nodes which improve the stability of the memory cell [17]. Conventional asymmetrical nine transistor SRAM (CA9T

Manuscript received on February 27th, 2025. Received in revised form on July 22nd, 2025. Accepted for publication on August 11th, 2025.

M.Kavitha is with Department of Electronics and Communication Engineering, Government College of Engineering-Bargur, Tamilnadu, India-635104. email: kavithaengr@gmail.com

S.Ramani is with Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India-501301. Email: dr.ramani2017@gmail.com

P.K.Janani is Research Scholar, Faculty of Science and Humanities, Anna University-Chennai, Tamilnadu, India-600025. E-mail: pkjanani2003@gmail.com

SRAM) given in Fig. 1 improves the read and write stability. Read stability is improved by the read buffer (M8 and M9) by isolating the storage nodes from bit lines while the writing ability is increased by the write assist transistor M7 [18]. In this paper a low power, high speed CNTFET based SRAM with enhanced write stability is presented and its performance is compared with conventional structure shown in Fig. 1.

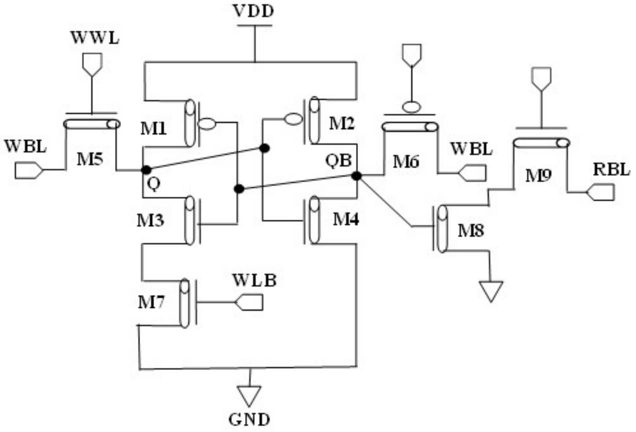


Fig. 1. CA9T SRAM Cell

II. Proposed SRAM Structure

The proposed asymmetrical nine transistor CNTFET SRAM (PA9T SRAM) with enhanced stability is depicted in Fig. 2. PA9T SRAM incorporates two techniques for minimizing power, namely stack and multi-threshold voltage technique. Transistors in cross coupled inverters (M1, M2, M3 and M4) are designed as high threshold transistors (HVT) according to (1).

$$V_{th} = \frac{0.43}{D_{CNT}(nm)} \tag{1}$$

where V_{th} - CNTFET threshold voltage and D_{CNT} -CNT tube diameter [7]. Transistors M6, M7, M8 and M9 are designed as low voltage transistors (LVT). Advantage of utilizing both LVT and HVT in circuit design is that it suppresses power consumption and improves stability without increasing the layout area. Transistor M8 and M9 are the read bit line access transistor and read transistor respectively. The HVT sleep transistor (NS) at the source of the read transistor alleviates the read bit line leakage during idle states. The sleep transistor is connected to all the sources of the read transistors i.e NS is shared by all the read transistors in the SRAM array.

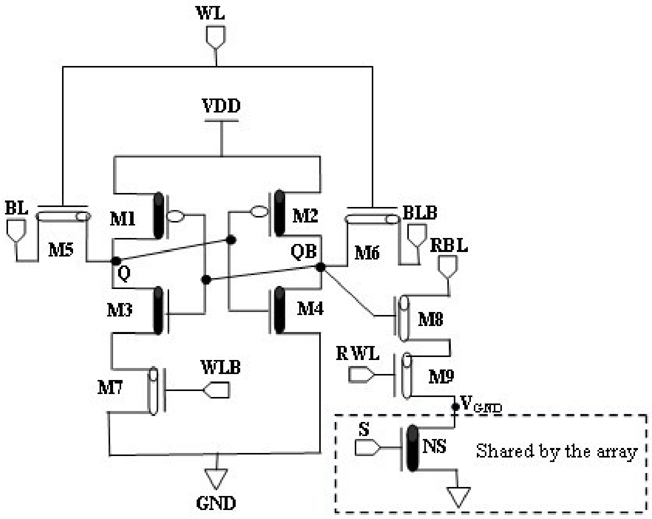


Fig. 2. PA9T SRAM Cell

Fig. 3 reveals that the proposed SRAM has a natural stacking effect, i.e the transistor M9 gets naturally stacked with the sleep transistor NS. Stacking effect reduces the currents through stacked pair transistors if more than one transistor is turned '*off*' [19]. Equations (1) to (8) prove the effect of stacking transistors on leakage. Let I_{off} is the subthreshold current with $V_{GS} = V_{BS} = 0$, $V_{DS} = V_{DD}$, and if $V_{DS} > 3kT/q$, the MOS transistor leakage is

$$I_{leak} = I_{off} 10^{-\frac{1}{S}[\Delta V_G + \lambda_d \Delta V_D + k_\gamma \Delta V_B]} \tag{2}$$

where S - subthreshold swing, λ_d - DIBL factor and k_γ - body effect co-efficient. Referring (2) the leakage of M9 and NS can be written as

$$I_{stack-M9} = W_9 I_{off} 10^{\frac{-(1+\lambda_d+k_\gamma)V_X}{S}} \tag{3}$$

$$I_{stack-NS} = W_{NS} I_{off} 10^{\frac{-\lambda_d(V_{DD}-V_X)}{S}} \tag{4}$$

where W_9, W_{NS} - width of the transistors M9 and NS respectively.

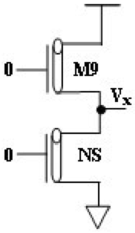


Fig. 3. Transistor Stack

The intermediate node voltage V_x reaches an equilibrium level when the leakage through M9 and NS are equal, and it can be determined from (3) and (4).

$$V_x = \frac{\lambda_d V_{DD} + S\, log \frac{W_9}{W_{NS}}}{1 + k_\gamma + 2\lambda_d} \tag{5}$$

$k_\gamma << 1+2\lambda_d$ in short channel devices and hence (5) becomes

$$V_x \approx \frac{\lambda_d V_{DD} + S\, log \frac{W_9}{W_{NS}}}{1 + 2\lambda_d} \tag{6}$$

Leakage in stacked M_9 and NS is found by substituting V_x in either (3) or (4).

$$I_{stack} = W_9{}^{\alpha} W_{NS}{}^{1-\alpha} I_1 10^{\frac{-\lambda_d V_{DD}}{S}(1-\alpha)} \tag{7}$$

where $\alpha = \frac{\lambda_d}{1+2\lambda_d}$. Leakage reduction in transistor stack with widths W_9 and W_{NS} with respect to a single transistor with width ‘W’ is

$$X = \frac{I_{single}}{I_{stack}} = \frac{W}{W_9{}^{\alpha} W_{NS}{}^{1-\alpha}} 10^{\frac{\lambda_d V_{DD}}{S}(1-\alpha)} \tag{8}$$

When W_9=W_{NS}=W, the stack factor X can be written as

$$X = 10^{\frac{\lambda_d V_{DD}}{S}(1-\alpha)} \text{ Or } X = 10^{\frac{\lambda_d V_{DD}}{S}\left(\frac{1+\lambda_d}{1+2\lambda_d}\right)} = 10^{U} \tag{9}$$

where *‘U’*- universal stack exponent of two transistors. It is apparent from (9) that leakage is greatly reduced in stacked transistors compared to a single transistor. Equations (2) to (9) are applicable to CNFETs also, hence proving the stacking has considerable effect in CNFET circuits also [20]-[21]. Hence the proposed SRAM can provide high power reduction due to stacking effect as well as the multi-threshold voltage design.

PA9T SRAM supports single ended read operation. RBL is charged to VDD prior to read operation and then the read control signal RWL is asserted to read the data in storage nodes. The word line WL and sleep signal ‘S’ are maintained at 0V and VDD respectively. In case if ‘1’ is stored at QB, RBL gets discharged through the read port (M8, M9 and sleep transistor NS). Alternatively, if the data at QB is ‘0’, then RBL is maintained at VDD level due to the absence of discharge path. Read stability is improved in this structure as the data is isolated from the bit lines as well as the source of M9 are maintained at ~0V.

PA9T SRAM provides double ended write operation, wherein the bit lines BL and BLB are charged to VDD and 0V respectively in order to write ‘1’ at the storage node Q. Read word line RWL and sleep signal are maintained at 0V and VDD respectively. Data ‘1’ is forced from the bit line BL to the node Q through the write access transistor M5. To write ‘0’, BL is maintained at 0V while BLB is held at VDD. During this operation, the contention current of M3 is eliminated as the transistor M7 is in ‘off’ condition. Write stability is improved by the HVT cross coupled inverters without compromising layout area. In sleep or hold mode, word control line WL, read control line RWL, and sleep signal ‘S’ are maintained at 0V. Sleep transistor NS is cutoff to minimize read bit line leakage as the transistors M8, M9 and NS form stacked pair. Further the cell leakage current is reduced due to the usage of high threshold voltage transistors in cross coupled inverters. Since the source of M3 is at level ~0V, cross coupled inverters experience almost full supply voltage which enhances hold stability.

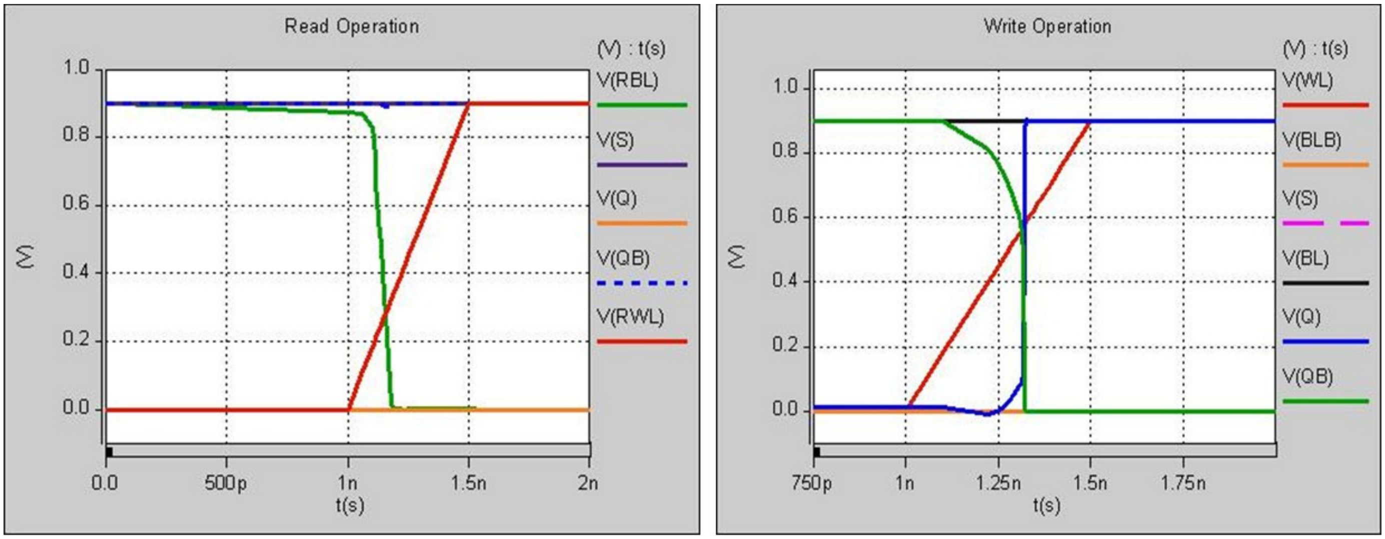
III. Results and Discussion

Stanford University CNTFET model and HSPICE is used to simulate the conventional and proposed SRAMs. Parameters used for SRAM cell design are:

- Channel length of both CNTFETs: 32nm
- Temperature: 27°C
- CNT channel mean free path (L_{geff}): 100nm
- Coupling capacitance (C_{sub}): 40pF/m

Influence of nanoarray pitch, gate dielectric, number of CNT tubes used in CNTFET, supply voltage on the performance of CA9T SRAM and PA9T SRAM is analyzed. Metrics considered for investigating the performance of CA9T SRAM and PA9T SRAM are hold, read and write power, read and write delay, power delay product (PDP) and SNM. Conventional and proposed asymmetrical SRAM structures are maintained in hold, read and write modes as discussed in Section II to evaluate these metrics.

The states of various nodes in proposed PA9T SRAM during read and write modes are provided in Fig. 4. Figs. 5 to 8 reveals the effect of CNFET parameters on power and delay of



(a) (b)

Fig. 4. PA9T SRAM various node states in (a) Read operation (b) Write operation

CA9T SRAM and PA9T SRAM. In hold mode, the SRAM is in data retention state, and no control signal is asserted. The power consumption in this mode is mainly due to leakage of the cross coupled inverters, write access transistors and read port transistors. As threshold voltage increases, power decreases hence leakage in cross coupled inverters is mitigated as it contains high threshold voltage transistors. Also, as the sleep transistor is in ‘off’ state and the source of the read port transistor M9 is maintained at VGND that is at a level higher than the ground level leading to leakage power minimization. Addition of the sleep transistor NS does not increase the layout area as it is shared by all the read ports in SRAM array, but it contributes a lot in leakage reduction in hold mode. Read power is reduced in PA9T SRAM because the ‘on’ sleep transistor raises the source level of the read transistor M8 and read operation is also faster than CA9T SRAM. Write access transistor is turned ‘off’ and the source of M3 is at potential greater than ground level, which aids in writing the data into the storage node Q quickly. The write power and write delay is low compared to conventional CA9T SRAM.

Dependence of leakage power and dynamic power on supply voltage of CMOS VLSI circuits is given by (10) and (11) respectively.

$$P_{leak} \propto I_{leak} V_{DD} \tag{10}$$

$$P_{dynamic} = C_L V_{DD}^2 f \tag{11}$$

where C_L - load capacitance, V_{DD} - supply voltage and f - switching frequency [10]. The gate delay is

$$T_{\mathrm{pd}} \alpha \frac{C_L V_{\mathrm{DD}}}{I_{\mathrm{DS}}} \tag{12}$$

where V_{th} - threshold voltage, and I_{DS} - drain to source current [10]. Simulation results in Fig. 5 confirm the above equations that as supply voltage increase leakage, dynamic power and delay increase.

When pitch is varied, the inner tube spacing between the consecutive tubes gets varied. According to (13), increasing the pitch raises the gate width, and gate width is inversely proportional to ‘on’ resistance. So, increasing the pitch increases power consumption and on contrary minimizes delay as shown in Fig. 6.

$$W_G = max(W_{min}, N * S) \tag{13}$$

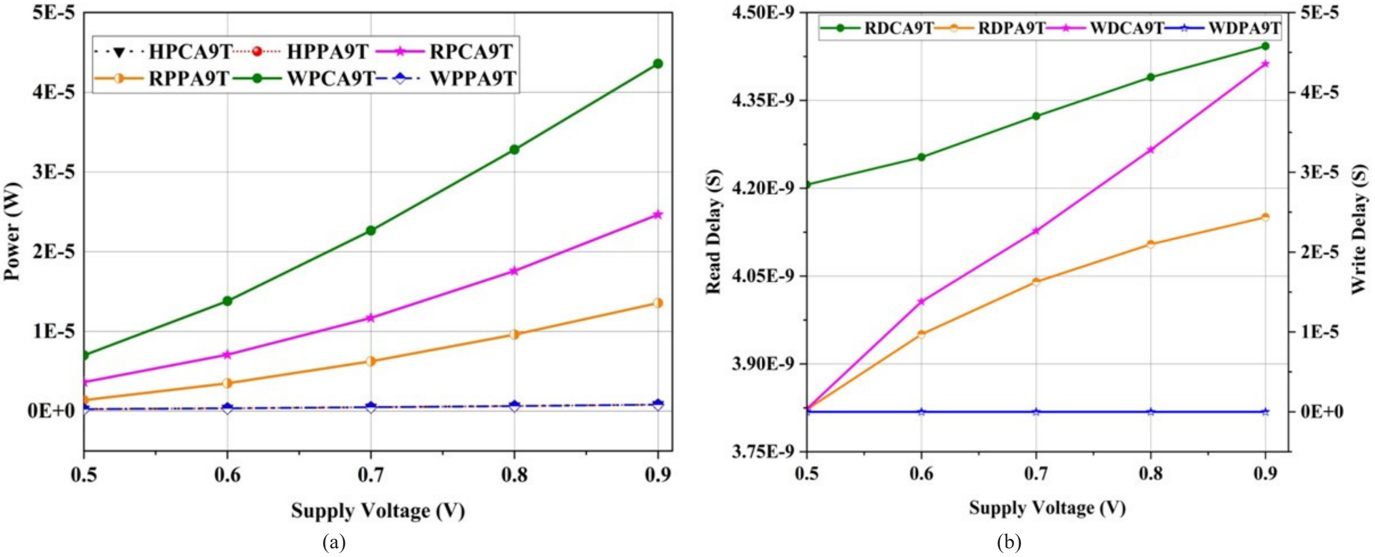
where W_G - total gate width, W_{min} - minimum gate width, N - number of CNT tubes and S - pitch [7]. Relation between the number of CNT tubes used in CNTFET and its ‘on’ current is

$$I_{CNFET} = N.g_{CNT}(V_{DD} - V'_{SS} - V_{th}) \tag{14}$$

where N - number of CNT tubes per device, g_{CNT} - transconductance, and V_{SS}’ - voltage drop between the external and inner source node [7]. Equation (14) states that increasing the CNT tubes number, improves the ‘on’ current and the power consumption rises as given in Fig. 7 and delay is lowered.

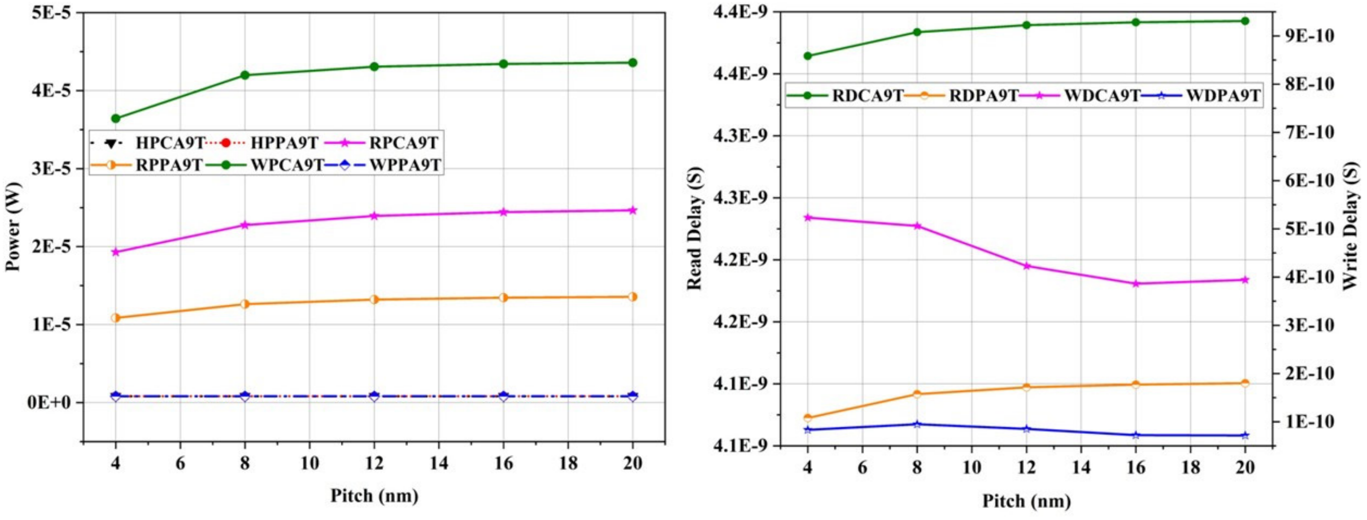
Dielectric material films are grown on top of gate of CNTFET. Power and delay for various dielectric materials like silicon dioxide (SiO2) with dielectric constant 4, Hafnium dioxide (HfO2) with dielectric constant 16, Zirconium dioxide (ZrO2) with dielectric constant 25, Tantalum pentoxide (Ta2O5) with dielectric constant 50 and Titanium dioxide (TiO2) with dielectric constant 80 is observed and the results are provided in Fig. 8. The current increases with dielectric constant resulting in higher power consumption and reduction in delay.

The results shown in Figs. 5 to 8 are averaged and tabulated in Table I and Table II. Percentage power reduction of PA9T with respect to CA9T is also given. Clearly, the performance of the proposed asymmetrical PA9T SRAM is better, under all parameter variations in view of circuit power and speed, especially in write power and write delay reduction, PA9T SRAM has high efficiency.



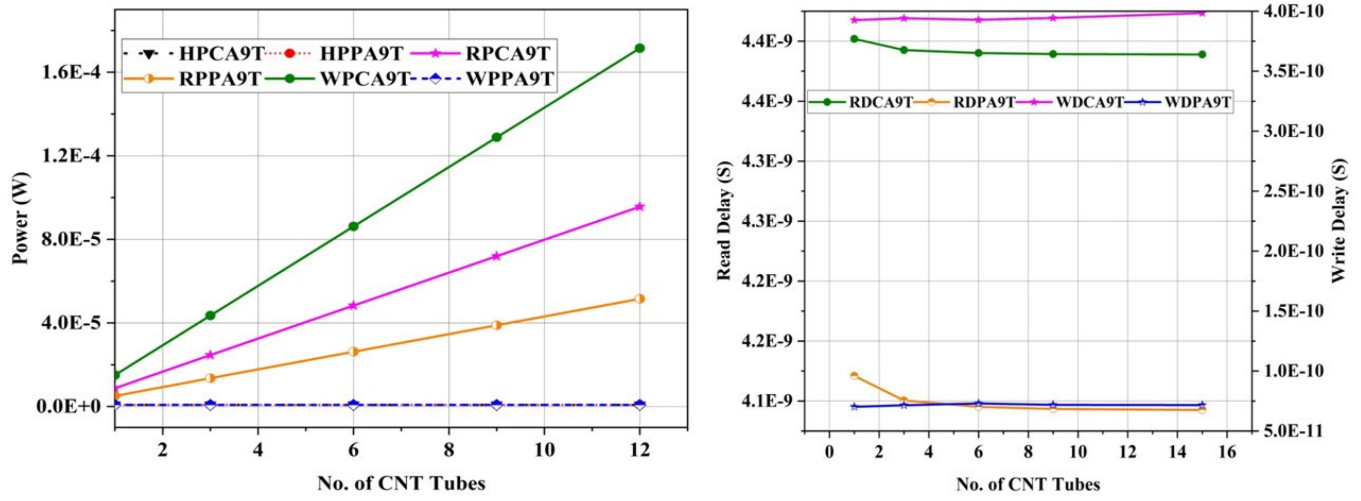
(a) (b)

Fig. 5. Voltage variations effect on (a) Power (b) Delay



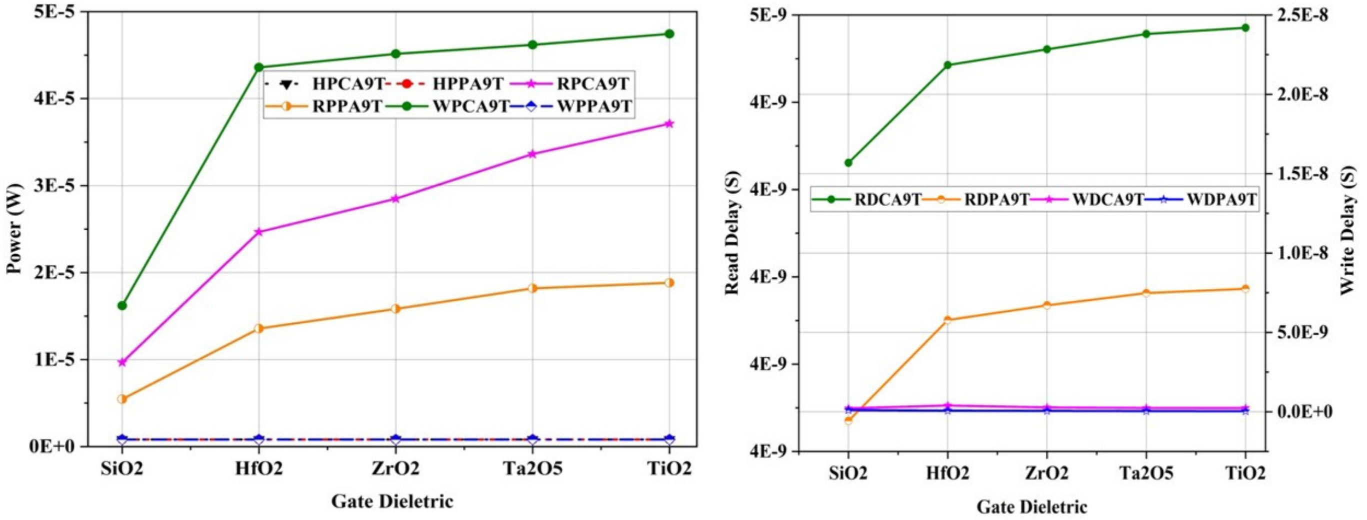
(a) (b)

Fig. 6. Pitch variations effect on (a) Power (b) Delay



(a) (b)

Fig. 7. CNT tubes variations effect on (a) Power (b) Delay



(a) (b)

Fig. 8. Gate dielectric variations effect on (a) Power (b) Delay

TABLE I
PARAMETER VARIATIONS EFFECT ON POWER

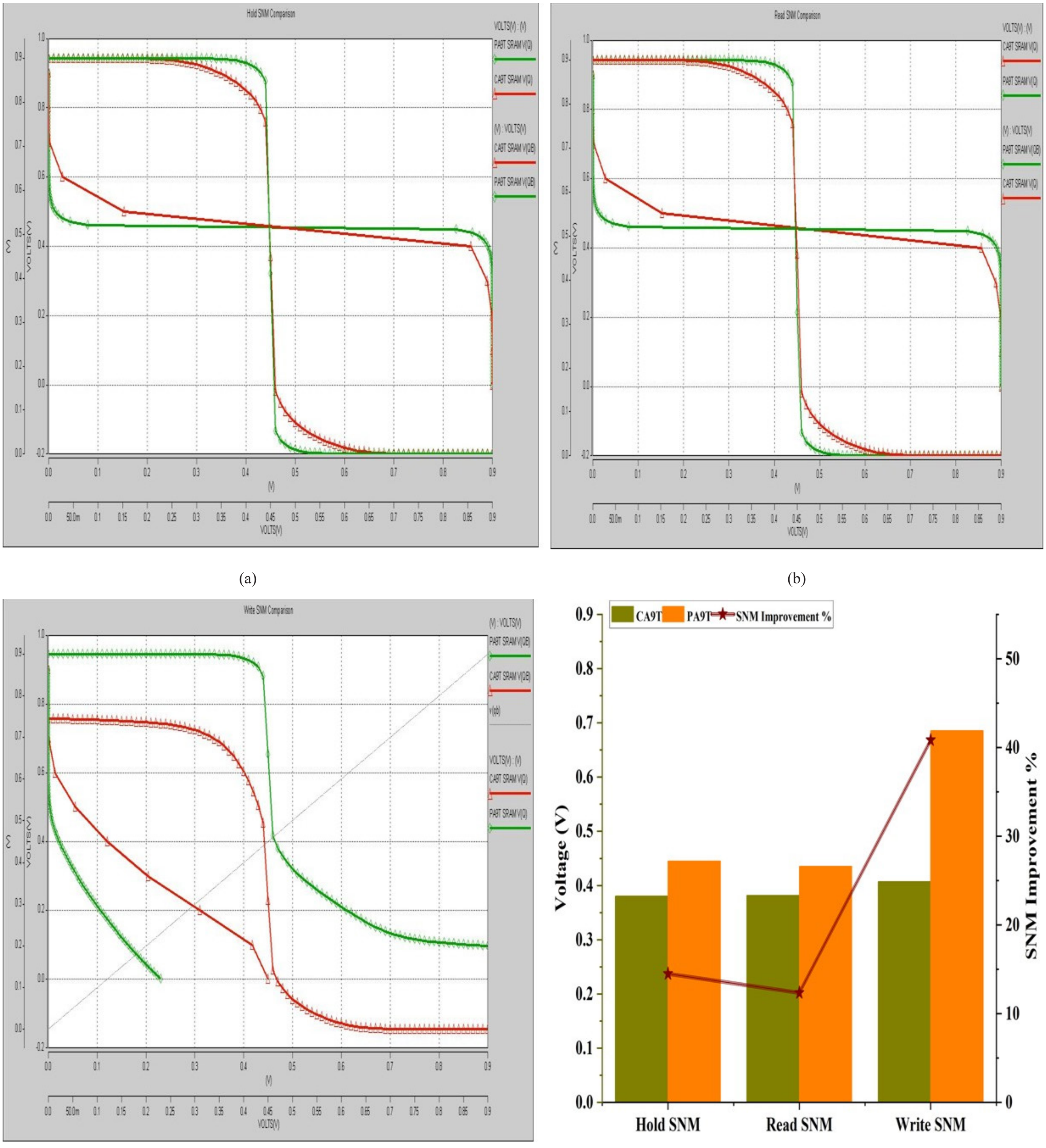
Parameters	CA9T SRAM	PA9T SRAM	%Power Reduction of PA9T
Effect of Voltage			
Hold Mode Power (W)	5.1047E-07	5.0999E-07	0.09
Read Mode Power (W)	1.2932E-05	6.8537E-06	47.00
Write Mode Power (W)	2.3969E-05	5.0999E-07	97.87
Effect of Pitch			
Hold Mode Power (W)	8.1171E-07	8.0998E-07	0.21
Read Mode Power (W)	2.3014E-05	1.2743E-05	44.63
Write Mode Power (W)	4.1698E-05	8.0999E-07	98.06
Effect of Number of CNT tubes			
Hold Mode Power (W)	8.1387E-07	8.0999E-07	0.48
Read Mode Power (W)	4.9874E-05	2.7070E-05	45.72
Write Mode Power (W)	8.9064E-05	8.0999E-07	99.09
Effect of Gate Dielectric			
Hold Mode Power (W)	8.1171E-07	8.0998E-07	0.21
Read Mode Power (W)	2.6712E-05	1.4372E-05	46.20
Write Mode Power (W)	3.9710E-05	8.0999E-07	97.96

TABLE II
PARAMETER VARIATIONS EFFECT ON DELAY

Parameters	CA9T SRAM	PA9T SRAM	%Speed Improvement of PA9T
Effect of Voltage			
Read Delay (S)	4.3228E-09	4.0136E-09	7.15
Write Delay (S)	2.2640E-05	9.1776E-11	100.00
Effect of Pitch			
Read Delay (S)	4.4343E-09	4.1423E-09	6.59
Write Delay (S)	5.2254E-10	8.1359E-11	84.43
Effect of Number of CNT tubes			
Read Delay (S)	4.4426E-09	4.1505E-09	6.57
Write Delay (S)	3.9447E-10	7.1605E-11	81.85
Effect of Gate Dielectric			
Read Delay (S)	4.4395E-09	4.1442E-09	6.65
Write Delay (S)	4.5591E-09	6.3931E-11	98.60

TABLE III
PARAMETER VARIATIONS EFFECT ON ENERGY

Parameters	CA9T SRAM	PA9T SRAM	%Energy Reduction of PA9T
Effect of Voltage			
Hold PDP (J)	1.4509E-11	4.4682E-17	100.00
Read PDP (J)	4.0313E-10	5.8230E-16	100.00
Write PDP (J)	7.3642E-10	4.4683E-17	100.00
Effect of Pitch			
Hold PDP (J)	4.2415E-16	6.5899E-17	84.46
Read PDP (J)	1.1644E-14	1.0332E-15	91.13
Write PDP (J)	2.1264E-14	6.5900E-17	99.69
Effect of Number of CNT tubes			
Hold PDP (J)	3.2105E-16	5.7999E-17	81.93
Read PDP (J)	1.9728E-14	1.9458E-15	90.14
Write PDP (J)	3.5231E-14	5.7999E-17	99.84
Effect of Gate Dielectric			
Hold PDP (J)	3.7007E-15	5.1783E-17	98.6
Read PDP (J)	4.8752E-14	8.0052E-16	98.4
Write PDP (J)	8.0502E-14	5.1783E-17	99.9



(a)

(b)

(c)

(d)

Fig. 9. SNM Comparison (a) Hold SNM (b) Read SNM (c) Write SNM (d) Percentage SNM Improvement

The most important concern apart from power and delay while designing a SRAM cell is its stability. Higher the stability or static noise margin (SNM) better is the memory cell's reliability and robustness. Stability of cell is estimated from the voltage transfer characteristics (VTC) of the cross coupled inverters in SRAM cell or from butterfly diagram. Butterfly diagram can be attained by merging the VTC of the cross coupled inverters by maintaining the SRAM in hold, read and write states. From the butterfly diagram, static noise margin (SNM) is measured by inscribing a square in the lobe and estimating its diagonal length. Fig. 9(a), Fig. 9(b) and Fig. 9(c) provides the butterfly diagram of CA9T SRAM and PA9T SRAM. On comparing the conventional and proposed SRAM it is found that the proposed PA9T SRAM hold and read stability is improved by 14% and 12% respectively. The write stability is highly enhanced in PA9T SRAM to about 41% than CA9T SRAM. The improvement in stability is due to the usage of HVT transistors and absence of P-type CNTFET in the proposed structure. Details about SNM values and percentage stability improvement are presented in Fig. 9(d).

Power delay product (PDP) is another important metric used to analyze the performance of VLSI circuits. PDP is the product of power and delay, and it implies the energy consumption of the circuit. Table III details the energy consumed by CA9T SRAM and PA9T SRAM due to parameter variations. PDP is evaluated for the same parameter values used to estimate power and delay, and all the corresponding values are averaged and given in Table III. It is obvious that the PA9T SRAM is better than CA9T SRAM with respect to energy also, and the energy reduction of PA9T SRAM varies from 82% to 100%.

IV. Conclusion

A CNTFET based SRAM which offers low power, low energy, high speed and enhanced stability is presented in this work. Supply voltage, pitch, number of CNT tubes and gate dielectric material are the parameters considered for the investigation. The performance of the proposed PA9T SRAM is better than the conventional CA9T SRAM proving it as a variations tolerant SRAM. Particularly PA9T SRAM read power, write power, write delay and PDP reduction is increased by 47%, 99%, 100% and 100% respectively. Write stability of the proposed SRAM is improved by 41% ensuring PA9T SRAM as a suitable structure for battery powered wearable devices in healthcare field where low power, high speed and reliability are of important concern.

References

[1] Y.H.Chen, T.Krishna, J.S.Emer and V.Sze, “Eyeriss: An Energy-efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks,” *IEEE J. Sol. St. Ckts*, vol.52, no.1, pp.127-138, Jan.2017. DOI: 10.1109/JSSC.2016.2616357

[2] H.N.Patel, F.B.Yahya, and B.H.Calhoun, “Optimizing SRAM bit cell reliability and energy for IoT applications,” in *Proc.17th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2016, pp. 12–17. DOI: 10.1109/ISQED.2016.7479149

[3] S.B.Baker, W.Xiang and I.Atkinson, “Internet of Things for Smart Healthcare: Technologies, Challenges and Opportunities,” IEEE *Access*, vol.5, pp.26521-26544, 2017. DOI: 0.1109/ACCESS.2017.2775180

[4] H.Bhatia, S.N.Panda, and D.Nagpal, “Internet of Things and its applications in healthcare – a survey,” in *Proc.2020 8th Int. Conf. Rel., Inf. Tech. Opt. (ICRITO)*, 2020, pp.305–310. DOI: 0.1109/ICRITO48877.2020.9197816

[5] M. Arbaoui, Brahmia and A. Rahmoun, “A review of IoT architectures in smart healthcare applications,” in *Proc.2022 Seventh Int. Conf. Fog. Mobile Edge Comp. (FMEC)*, Paris, France, 2022, pp.1-8. DOI: 10.1109/FMEC57183.2022.10062841

[6] Naagesh S. Bhat, “Design and modeling of different SRAM’s based on CNTFET 32nm technology,” *Int. J. VLSI Des. Commn. Sys.* vol.3, no.1, pp.1-9, 2012. DOI:10.5121/vlsic.2012.3106

[7] P.A.Gowri Sankar and K.Udhayakumar, “MOSFET-like CNTFET based logic gate library for low-power application: a comparative study,” *J. Semicond*., vol.35, no.7, pp.1-13, 2014. DOI: 10.1088/1674-4926/35/7/075001

[8] Shimaa Ibrahim Sayed, Mostafa Mamdouh Abutaleb, and Zaki Bassuoni Nossair,“Optimization of CNTFET parameters for high performance digital circuits,” *Ad. Mat. Sci. Engg.,* vol.2 no.1, pp.1-9, 2016. DOI: 10.1155/2016/6303725

[9] M.Kavitha and A.M. Kalpana, “Carbon Nanotubes: properties and applications - a brief review,” *I-manager’s J. Elect. Engg.,* vol.7,no.3, pp. 1-6,2017. DOI: 10.26634/JELE.7.3.13558

[10] M.Keating, D.Flynn, R.Aitken, A.Gibbons and K.Shi,“Low power methodology manual for system on chip design,” *Springer*, New York, USA, 2007. DOI: 10.1007/978-0-387-71819-4

[11] L.Chang *et al*., “Stable SRAM cell design for the 32 nm node and beyond,” in *Proc. Digest of Technical Papers. 2005 Symp. VLSI Tech.2005.* Kyoto, Japan, 2005, pp. 128-129. DOI: 10.1109/.2005.1469239

[12] Y.Sun, H.Jiao and V.Kursun, “A Novel Robust and Low-Leakage SRAM Cell With Nine Carbon Nanotube Transistors,” *IEEE Trans. VLSI Sys.*, vol. 23, no. 9, pp. 1729-1739, 2015. DOI: 10.1109/TVLSI.2014.2350674

[13] A Teman, L.Pergament, O.Cohen and A.Fish, “A 250 mV 8 kb 40 nm Ultra-Low Power 9T Supply Feedback SRAM (SF-SRAM),” *IEEE J. Sol. St. Ckts.*, vol. 46, no. 11, pp. 2713-2726, 2011. DOI: 10.1109/JSSC.2011.2164009

[14] M.Kavitha and T.Govindaraj, “Low-power multimodal switch for leakage reduction and stability improvement in SRAM Cell,” *Arabian J. Sci. Engg*., vol.41, pp.2945-2955,2016. DOI: 10.1007/s13369-016-2047-0

[15] M.Kavitha and A.M.Kalpana, “Low leakage charge recycling technique for power minimization in CNTFET circuits,” *Acta Polytechnica*, vol.59, no.1, pp. 24-34, 2019. DOI: 10.14311/AP.2019.59.0024

[16] Z.Liu and V.Kursun, “Characterization of a Novel Nine-Transistor SRAM Cell,” *IEEE Trans. VLSI Sys.*, vol. 16, no. 4, pp. 488-492, 2008. DOI: 10.1109/TVLSI.2007.915499

[17] M. Kumar and P. V. Sridevi, “A Proposed Low Standby Power Read Disturb-Free 9T SRAM Cell,” in *Proc. 2018 Int. Conf. Ckts. Sys. Dig. Ent. Tech.* (*ICCSDET)*, pp.1-4, 2018. DOI:10.1109/ICCSDET.2018.8821151

[18] Y.Sun, W.He, Z.Mao, H.Jiao and V.Kursun, “High-Yield and Robust 9T SRAM Cell Tolerant to Removal of Metallic Carbon Nanotubes,” *IEEE Trans. Dev. Mat. Rel.*, vol. 17, no. 1, pp. 20-31, 2017. DOI: 10.1109/TDMR.2017.2668761

[19] S.Narendra, S.Borkar,V.De, D.Antoniadis and A.Chandrakasan, “Scaling of stack effect and its application for leakage reduction,” in *Proc. Int. Symp. Low Power Elect. Des.,* pp.195-200, 2001. DOI: 10.1145/383082.38313

[20] J.Deng and H.S.P.Wong, “A Compact SPICE Model for Carbon Nanotube Field-Effect Transistors Including Non-idealities and Its Application—Part I: Model of the Intrinsic Channel Region,” *IEEE Trans. Elect. Dev.*, vol. 54, no. 12, pp. 3186-3194, 2007a. DOI: 10.1109/ted.2007.909030

[21] J.Deng and H.S.P.Wong, “A Compact SPICE Model for Carbon-Nanotube Field-Effect Transistors Including Nonidealities and Its Application—Part II: Full Device Model and Circuit Performance Benchmarking,” *IEEE Trans. Elect. Dev.*, vol. 54, no. 12, pp. 3195-3205, 2007b. DOI: 10.1109/ted.2007.909030

A GCN-Attention Model for Precision Irrigation Evaluation

Ying Huang and Meng Liu

***Abstract*—The challenges of traditional quantitative irrigation methods cannot adapt to the dynamic actual soil moisture content and meteorological changes, and the existing methods based on soil moisture thresholds cannot fully solve the problems of hysteresis and adaptability, lack comprehensive consideration of meteorological factors and growth dynamics, and fail to consider the subtle sensitivity to soil moisture changes and processing efficiency limitations. To address the above challenges, we propose UFO-GCN-SPANet, a novel and computationally efficient architecture specifically designed for resource-constrained precision agriculture. Its core innovation lies in the cascaded integration of: (1) a linear-complexity Unit Force Operated Vision Transformer (UFO-ViT) that replaces quadratic self-attention with matrix associativity and cross-normalization for efficient global spatio-temporal feature extraction; (2) Graph Convolutional Networks (GCNs) for modeling spatial dependencies; and (3) a Salient Positions-based Attention Network (SPANet) employing a novel Significant Position Selection (SPS) algorithm to dynamically focus computation on the most informative contextual features, drastically reducing complexity while enhancing discriminative power. This unique combination directly addresses the critical challenges of computational efficiency and effective context modeling in real-world irrigation systems. Experimental results show that the proposed method outperforms traditional GNN models such as SAGEConv with 12 standard time series forecasting methods in key metrics, including accuracy, precision, recall, and F1-Score.**

***Index Terms*—Precision Agriculture, Soil Moisture Prediction, Machine learning**

Original Research Paper
DOI 10.53314/ELS2529070H

I. Introduction

THE soil moisture content plays a significant impact on various aspects of agriculture and water resource management, affecting crop growth, irrigation efficiency, yields, and health. Insufficient soil moisture would impair crop growth, excessive soil water would lead to root diseases, affecting crop health. Precise irrigation decision is a complex task involving multiple factors, requiring consideration of soil water content, soil temperature, and various weather conditions, which involves technologies such as sensor technology, IoT (Internet of Things), artificial intelligence, and data analytics techniques.

A smart irrigation system is an intelligent management platform designed to optimize crop growth conditions by precisely controlling the amount of water applied, increasing water resource utilization efficiency, and mainly composed of sensing layer, communication layer, data layer, model layer, and decision layer, as shown in Fig. 1.

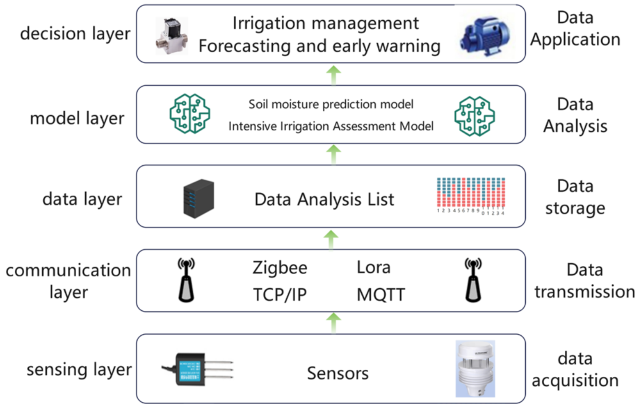


Fig. 1. Block Diagram of Precision Irrigation System Components

The sensing layer primarily consists of soil parameter sensors (e.g., moisture and temperature) and meteorological station sensors (e.g., temperature, humidity, and rain quantity), which collect data from the field and convert it into a signal for real-time status information to provide. The communication layer utilizes wireless communication technology to process data collected from sensors, sending to the central processing system. The data layer stores raw data and dealt with information, supporting model layer and decision layer's computation and analysis. The model layer includes soil moisture prediction model and precipitation evaluation model, providing scientific decision-making based on the model layer's results. The decision layer generates irrigation decisions based on the results

Manuscript received on May 17th, 2025. Received in revised form on August 26th, 2025. Accepted for publication on September 17th, 2025.

Ying Huang is with the Faculty of School of Automotive and Information Engineering, Guangxi Eco-engineering Vocational and Technical College, Liuzhou Guangxi, P.R. China. At the same time, he is pursuing a doctoral degree at Wuhan University. (e-mail: huangying800816@163.com).

Meng Liu is with the Faculty of School of Automotive and Information Engineering, Guangxi Eco-engineering Vocational and Technical College, Liuzhou Guangxi, P.R. China (corresponding author, e-mail: 517273947@qq.com).

This paper was supported by the Talent Introduction Scientific Research Project (NO.GXSTKYZX202406002) of Guangxi Ecological Engineering Vocational and Technical College, the first phase of the training project of famous teachers (craftsmen), and the project construction project of famous master studios.

from the model layer, determining when to water, how much to water, and how to water, using real-time data and predictive models to automatically adjust the irrigation plan and send an instruction for controlling the irrigation controller, valves, and pumps on the hardware devices, enabling the operation of the irrigation system.

In the system, information flow and control flow are clearly illustrated in Fig. 2. Lora nodes collect soil parameters and weather parameters via 3G/4G/GPRS technologies and transmit them to the cloud server. Users can interact with the terminal interface to retrieve and administer data. The user controller processes and models the gathered information, formulates the irrigation decision plan, and distributes results through a network to Lora nodes for control of the electric valves that regulate the irrigation machinery.

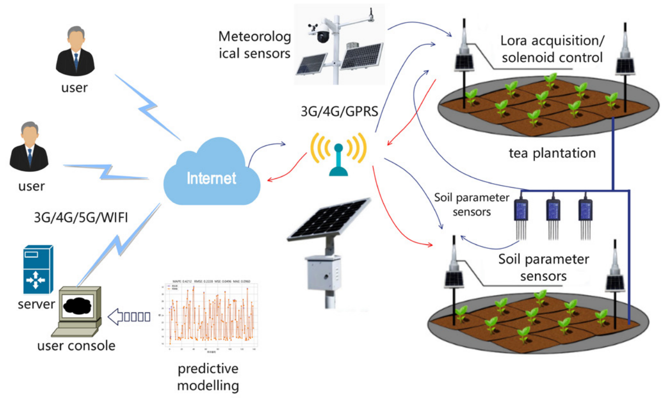


Fig. 2. Schematic diagram of data and information flow of precision irrigation system

The ecological economy and economic benefits of the tea industry are closely connected with irrigation efficiency. The growth of tea trees is highly sensitive to soil moisture content, and changes in soil moisture content directly affect the yield and quality of tea leaves. The water demand of tea trees exhibits significant seasonal variation. The water requirement of tea trees shows notable differences during various growth stages, with higher demand during the seedling and growth stages. Moderately controlling irrigation before the harvest period can increase the quality and yield of tea leaves, while irrigation should be reduced in winter to prevent root frost damage. This dynamic water demand places higher demands on the real-time monitoring accuracy of soil moisture content. Additionally, tea plantations are situated in hilly and mountainous areas, where the complex terrain leads to differentiated soil moisture content at different slopes and altitudes. These unique topographical conditions impose higher requirements for the design and implementation of irrigation systems, making it difficult for traditional empirical irrigation methods to cope with the moisture disparities caused by complex terrain, which can easily lead to localized water shortages or flooding. The unique microclimate conditions also directly impact the water supply and irrigation needs of the tea garden, adding to the difficulty of irrigation.

Accurately predicting soil moisture is essential for providing systematic irrigation guidance and optimizing water resources. Therefore, precise irrigation requires the use of accurate irrigation decision models that may be influenced by soil moisture, crop needs, and meteorological conditions. This demands a irrigation management system that can accurately detect and address these non-linear characteristics. The variability in soil moisture across different regions also necessitates that efficient irrigation handles localized monitoring and differential management of soil moisture. On the other hand, quantitative irrigation with fixed schedule is typically initiated by limiting soil moisture levels, such as when there are drought conditions. However, deciding whether a field is dry or need watering requires personal experience and effort during irrigation. Moreover, the difficulty in precise gauging irrigation duration and quantity, coupled with the variability across fields, leads to significant water waste issues.

The research and application of integrated water-saving techniques, such as drip irrigation and water-fertilizer integration, are still in the research stage and demonstration phase. Due to insufficient technical maturity, the fields in data collection and irrigation decisions will continue to face challenges, while also facing a gap with other technologies tree combinations. The precision of the methods for irrigation has room for significant research space regarding the application in tea trees. Deep learning is continuously advancing, making its modeling method prominent in the field of irrigation precise planning. The study on machine learning for tea tree irrigation decision-making prediction and assessment is deficient. In this manuscript, we propose the novel UFO-GCN-SPANet framework maintains compatibility with mainstream IoT irrigation platforms through standardized data interfaces and control protocols. As illustrated in Fig. 1 and Fig. 2, the model layer can directly process sensor data from existing monitoring systems and generate irrigation commands compatible with commercial controllers, which feature:

The first application of UFO-ViT's linearized self-attention mechanism (O(N) complexity) in precision agriculture, effectively overcoming the computational bottleneck of standard Transformers (O(N^2)) for modeling long-range dependencies in sensor network data.

The introduction of SPANet with the innovative SPS algorithm into the irrigation evaluation domain. SPS actively selects only the top-k salient positions for attention computation, achieving significant reductions in computational cost and memory footprint while enhancing model focus on critical contextual information and node-level interactions.

A purpose-built cascaded architecture (UFO-ViT -> GCN -> SPANet) designed to progressively learn rich global features, incorporate spatial topology, and perform focused context refinement, specifically optimized for the efficiency demands and dynamic nature of resource-limited agricultural environments.

Comprehensive experimental validation and ablation studies demonstrate the substantial individual and synergistic contributions of each proposed component (UFO-ViT, SPANet) to the overall model's superior accuracy and efficiency.

II. Related Works

Machine learning algorithms are increasingly used in water resources prediction, and scholars have made significant advancements. Puspaningrum et al. proposed a machine learning algorithm for irrigation prediction, which is capable of accurately predicting irrigation using classification algorithms such as support vector machine, k-nearest neighbors, Plain Bayes, Random Forest, and Decision Tree in the method [1].

Priya et al. integrated IoT technology and machine learning algorithms with precision agriculture by using sensors to collect data such as temperature, humidity, soil moisture, and water level, and used different machine learning models including support vector machine, decision tree, random forest, and plain Bayes to predict crop water requirement and control the operation of pumps based on the prediction results to optimise irrigation management [2].

Filgueiras proposed a methodology to predict water management parameters based on Vegetation Indices (VIs) and regression algorithms, where Random Forests, Cubic Regression, and Gradient Boosters were used for validation of the data to fulfil the need for fully remote irrigation management [3].

Abioye et al. focused on IoT-based monitoring and data-driven modelling of drip irrigation system for mustard leaf cultivation and proposed an improved monitoring and data-driven model for monitoring the dynamics of parameters affecting the irrigation of rapeseed leaf plants [4].

Narakala et al. constructed a Crop Water Stress Index (CWSI) model, which takes three parameters as inputs such as relative humidity, air temperature, and canopy temperature, and various machine learning methods such as multi-layer perceptron, random forests, and decision trees were used to evaluate the CWSI model, and all of them showed good performance [5].

The study by Bwambale et al. demonstrated the great potential of machine learning in improving the accuracy of irrigation management and also proposed a data-driven model predictive control approach for precision irrigation, highlighting the importance of accurate predictive modelling for optimising water use [6].

In order to be able to avoid water wastage, Bhoi et al. used machine learning and IoT techniques to build a smart irrigation recommendation system that combines various machine learning based support vector regression and KNN classifier models to achieve efficient water use and reduce human intervention [7].

Vianny et al. proposed a hybrid model of an irrigation system using IoT components to collect information on soil moisture, soil temperature, weather conditions and environmental conditions, incorporating algorithms such as KNN, gradient boosting based trees, long and short-term memory and Spearman rank correlation to optimise demand and thus reduce energy consumption [8].

The development of smart irrigation systems has been the focus of recent research efforts. Campos et al. proposed the Smart and Green framework to provide data monitoring, preprocessing, fusion, synchronisation, storage and irrigation management services for smart irrigation and to enrich these services by predicting soil moisture [9].

Liao et al. developed an intelligent irrigation system based on real-time soil moisture data, emphasising the importance of water-saving irrigation scheduling and the development of efficient automated irrigation systems [10].

Choi et al. emphasized the need for a more accurate transpiration model for tomato cultivation in order to achieve precise irrigation control in greenhouse soilless culture [11].

Similarly, Jo et al. aimed to develop a transpiration model for precise control of tomato irrigation under various environmental conditions in greenhouses. These studies highlight the importance of accurate transpiration modelling for efficient irrigation strategies [12].

In the field of precision irrigation, Chen et al. designed and implemented an intelligent irrigation robot to solve the problems of poor mobility, inaccuracy and high price in agricultural irrigation systems. The robot achieved precision irrigation through an improved irrigation path planning algorithm based on Bayesian theory [13].

In addition, Wu et al. explored the use of Ground Penetrating Radar (GPR) full-wave inversion to map soil moisture in a potato field in furrowed mound terrain, focusing on the effect of soil surface in furrowed mound terrain on GPR measurements for mapping soil moisture [14].

Abera et al. used extracted Soil-Adjusted Vegetation Index (SAVI) image greenness to identify irrigated cropland and used more precise irrigation techniques to minimize water losses by calculating the irrigation water requirement of the major crops grown under irrigation and determined using estimated irrigated area of the watershed [15]. These research methods and techniques provided the methodology and ideas to this project to study the precision irrigation techniques for tea trees.

Despite the success of research in precision irrigation, there are still multiple challenges to overcome. The variability of soil moisture content in different regions requires greater flexibility and adaptability of irrigation systems. The maturity and application scope of the technology needs to be improved, especially in the accuracy of data collection and irrigation decision making. Future research should focus on the integration and application of multiple technologies to promote the popularization and application of smart irrigation technologies and provide technical support for sustainable agricultural development.

III. Related Technologies

A. Graph Convolutional Network

Graph Convolutional Networks (GCNNs) are designed for modeling graph-structured data and can effectively capture complex relationships between nodes [16]-[17].

Given a graph $G = \{C, E\}$, let the node feature matrix be $X \in R^{N\times D}$, the adjacency matrix A and the degree matrix D. The output $Z \in R^{N\times F}$, with L layers. The propagation at each layer is defined as equation (1):

$$H^{(l+1)} = f(H^l, A) \tag{1}$$

where, $H^{(0)} = X, H^{(L)} = Z$, the number of layers of the network denoted L, the model selection and parameter approach will vary depending on the function $f(\cdot,\cdot)$, which $f(\cdot,\cdot)$ is the propagation rule.

So, a basic propagation rule is defined as equation (2):

$$f(H^{(l)}, A) = \sigma\left(AH^{(l)}W^{(l)}\right) \tag{2}$$

where $W^{(l)}$ denotes the weight matrix of the lth layer, which is used to map the lth layer features to the l + 1th layer, and $\sigma(\cdot)$ denotes the nonlinear activation function.

To stabilize the model, the adjacency matrix is typically normalized (adding self-loops followed by symmetric normalization), as shown in equation (3):

$$\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \tag{3}$$

The final standard propagation rule is then given by (4):

$$f(H^{(l)}, A) = \sigma\left(\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{4}$$

where $D^{-\frac{1}{2}}$ represents the square root of the inverse of the degree matrix, $\hat{A}$ =A+I is the unit matrix and $\hat{D}$ is the degree matrix of $\hat{A}$

The GCN-based soil moisture prediction method follows these key steps:

1. Sampling: Randomly select neighboring nodes (guided by a predefined strategy to limit computational load).

2. Aggregation: Combine neighbor features (via mean/max/ pooling) to represent the target node, capturing local graph structure.

3. Update: Transform aggregated features using a learnable weight matrix to enhance model expressiveness.

4. Propagation: Iteratively repeat sampling-aggregation-update until all nodes are processed, yielding updated node features.

B. Unit Force Operated Vision Transformer

The Transformer excels in vision and NLP tasks but faces three key challenges:

1) High Complexity: Self-attention scales quadratically with sequence length, increasing training time and resource demands.

2) Data Hunger: Requires large datasets to match CNN performance in low-data regimes.

3) Static Modeling: Struggles with spatio-temporal dynamics (e.g., video analysis).

To solve this problem, Unit Force Operated UFO-ViT (Unit Force Operated Vision Transformer) replaces softmax with cross-normalization (XNorm), reducing complexity while preserving performance [18]. Its architecture includes: normalisation layer, UFO model, convolutional layer and MLP model, the specific model is shown in Fig. 3.

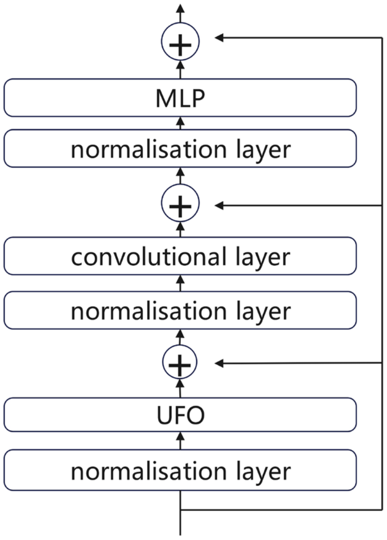


Fig. 3. Structure of the UFO-ViT model

For the attention mechanism A in the Transformer model is computed via equation (5), in which Q, K and V defined in equations (6), (7) and (8), are first obtained by linearly projecting the input sequence:

$$A = softmax\left(\frac{K^TQ}{\sqrt{d}}\right) \tag{5}$$

where:

$$Q = XW_Q \tag{6}$$

$$K = XW_K \tag{7}$$

$$V = XW_V \tag{8}$$

X denotes is the input feature vector, Q, K and V denote the query vector, key vector and value vector, W_Q , W_K and W_V weight matrices, d is the dimension of the input feature vector.

The nonlinear nature of softmax makes it impossible to decompose the result into $O\ (N \times h + h \times N)$. To solve this problem, the self-attention mechanism model is allowed to multiply first, eliminating the original softmax. but this method leads to performance degradation. Therefore, a cross-normalisation method (denoted as XNorm) is used, as defined by (9) and (10):

$$A(\mathbf{x}) = \mathrm{XN}_{\mathrm{dim=filter}}(Q)(\mathrm{XN}_{\mathrm{dim=s\ pace}}(K^TV)) \tag{9}$$

$$\mathrm{XN}(\mathbf{a}) := \frac{\gamma\mathbf{a}}{\sqrt{\sum_{i=0}^{h} ||\mathbf{a}||^2}} \tag{10}$$

where: γ denotes the learnable parameter, h denotes the embedding dimension, which is an L2 paradigm that applies not only to the spatial dimension of K^TV, but also to the channel dimension of Q.

Using the law of union, first let V and K be multiplied together and then multiplied with Q. The final result is linear with respect to n. The complexity of the two multiplications is O(hNd). Since the complexity of the two multiplications is O(hNd), the final result also has a linear relationship with n. The complexity of the two multiplications is O(hNd). This is shown in Fig. 4.

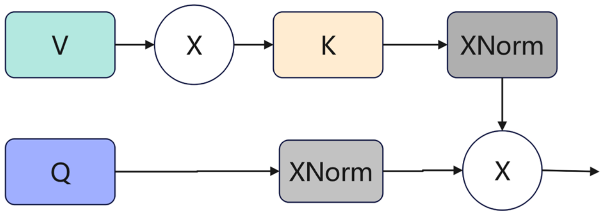


Fig. 4. Block diagram of the linearisation of the attention mechanism

C. Salient Positions-based Attention Schemer

Attention mechanisms, given their capacity to efficiently manage long-range dependencies in input sequences, are highly effective for tasks such as text and speech processing and have garnered substantial attention. As the global dependencies among input features increase, the demand for computational resources and memory correspondingly rises. While the attention mechanism can gather information globally, not all such information is beneficial for context modeling, potentially leading to performance degradation. To address this, SPANet introduces a method that computes attention solely on salient locations (a limited number of keypoints). This approach not only conserves a significant amount of computational resources and memory but also extracts useful information from the input feature maps, thereby enhancing prediction performance [19].

The structure of SPANet is shown in Fig. 5.

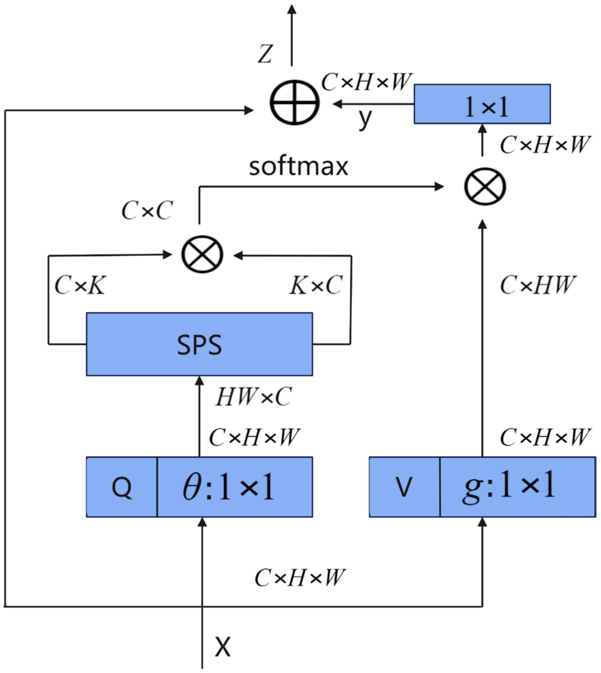


Fig. 5. SPANet structure

The input X is first transformed by a 1×1 convolution to yield Q and V, respectively. Q is then processed by the SPS algorithm, which extracts the top-k significant query-key matrices according to equations (11) - (13).

$$Q = \theta(x) \tag{11}$$

$$V = g(x) \tag{12}$$

$$K = S(Q) \tag{13}$$

The affinity matrix A is computed with a softmax over the top-k significant locations, as given in equation (14). And is multiplied by V to yield the output Y, as given by (15):

$$A = softmax\left(\frac{K^T K}{C}\right) \tag{14}$$

$$Y = VA \tag{15}$$

After a 1×1 convolution, Y is finally added back to the original input X.

In the SPS algorithm, positive context information is extracted when modelling global dependencies, i.e., attentional computation is performed at salient locations, which reduces the complexity of the computational procedure. The SPS algorithm can be described as:

```
1. input query key Q of dimension [c, h*w], hyperparameter k value
2. Calculate the square power of Q^T based on the channel dimension
3. Sum Q^T to get Qpow
4. sort by largest to smallest, take the first k values, and add index index
5. return K=Q (c, index)
```

IV. Methods for precision irrigation evaluation

A. General structure

Accurate prediction of soil water content is an important basis for providing scientific irrigation guidance and optimizing water use. Graph conventional neural networks may be unable to adequately capture all the complex relationships in the graph, while traditional attention mechanisms, although effective in dealing with long-distance dependencies in the input sequences, are deficient in capturing information about the spatio-temporal dynamics. As the global dependencies of the input features increase, the complexity and training cost of the model also rise, and the demand for computational resources and memory continues to increase. To address these issues, this manuscript proposes a graph neural network refined irrigation prediction method based on a fusion attention mechanism (UFO-GCN-SPANet) to overcome the inefficiency of graph neural networks in processing complex data, the insufficient global feature extraction, the non-enrichment of feature representations, and the multimodal data fusion problem, which firstly introduces a unit-forcing operation visual Transformer self-attention mechanism (UFO-ViT). The method firstly introduces the unit forced operation visual Transformer self-attention scheme (UFO-ViT), which significantly reduces the computational resources and generates high-quality feature representations to provide rich information for the graph neural network by using its linear complexity design, and then introduces the Salient Positions-based Attention Scheme (SPANet) after the graph neural network to provide rich information for the graph neural network by means of the Salient Positions Selection (SPS) to select only a limited number of salient points for attention graph computation, which not only reduces the computational complexity and memory requirements, but also improves the accuracy of image classification by extracting positive contextual information through selecting salient positions. The extraction of high-quality global features by UFO-ViT, graph structure information by GCN modelling, and positive context information by SPANet makes the whole network more efficient in processing large scale data and reduces computational resources and memory requirements. This method has significant practical application value in the field of agriculture, which can significantly improve the scientific and accuracy of irrigation strategies. The overall structure of the algorithm is illustrated in Fig. 6.

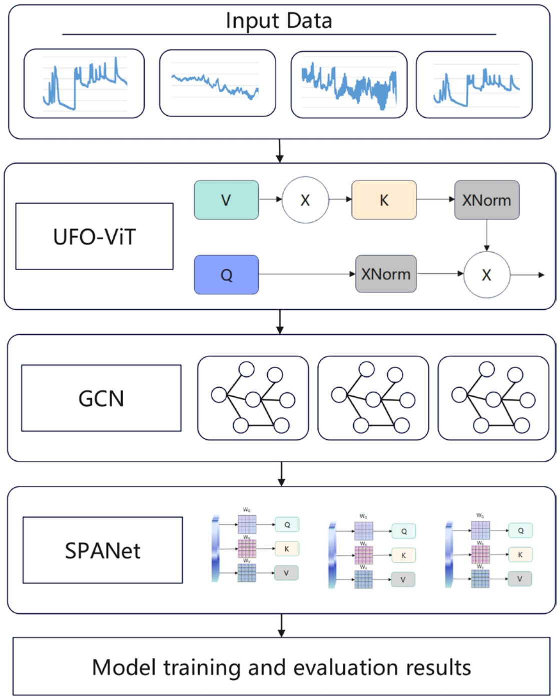


Fig. 6. Structure of graph neural network algorithm incorporating attention mechanism

The novelty and effectiveness of UFO-GCN-SPANet stem primarily from two key innovative components and their synergistic integration within the cascade.

UFO-ViT: Revolutionizing Efficiency in Global Modeling

Standard Transformer self-attention, with its O(N^2) computational complexity and memory requirements, becomes prohibitively expensive for processing the potentially long time-series and numerous nodes in agricultural sensor networks, hindering real-time deployment on edge devices. UFO-ViT fundamentally rethinks attention by leveraging matrix multiplication associativity and introducing Cross Normalization (XNorm). This allows it to completely eliminate the computationally intensive softmax(QK^T) operation central to standard Transformers. The resulting linear complexity (O(N)) enables UFO-ViT to efficiently capture complex long-range spatio-temporal dependencies across the entire sensor network without sacrificing representational power. This breakthrough in efficiency is crucial for practical applicability in resource-constrained precision irrigation systems.

SPANet: Selective Attention for Efficiency and Focus

While attention is powerful, computing interactions between all positions is often redundant and computationally wasteful. Noise and irrelevant variations in sensor data can dilute the effectiveness of standard attention. SPANet incorporates SPS algorithm dynamically identifies the top-k most significant positions (or nodes) based on the input features, slashing computational cost and memory usage, particularly beneficial for large graphs or long sequences. By focusing computation and modeling capacity on the most informative contextual interactions, SPANet effectively filters out noise and amplifies relevant signals, leading to more robust and discriminative node representations. This selective focus is a distinct advantage over standard, unselective attention mechanisms.

The sequential flow UFO-ViT (Efficient Global) -> GCN (Spatial Structure) -> SPANet (Focused Context Refinement) is carefully designed. UFO-ViT provides rich, efficient input features. GCN injects explicit spatial relationships. SPANet then performs localized, context-aware enhancement based on the graph-informed features, maximizing the utility of the selective attention. This holistic design directly targets the intertwined challenges of efficiency, global context, spatial structure, and relevant local interaction in precise irrigation evaluation.

B. Data processing and classification methods

In order to achieve dynamic and intensive irrigation, the soil water content variable servers as a basis for the study, and the data objects of the study are pre-processed. In order to make better decisions on precision irrigation, the soil water content can be classified as ‘low’, ‘medium’, ‘high’, or several types for specific purposes. In this section, the data are classified from 0 to 1, and the data are classified into three types of ‘low,’ ‘medium,’ and ‘high,’ or several types for specific uses, which can be classified into different irrigation levels to form a dynamic irrigation management model. In this section, the data are split into 10 equal parts from 0 to 1, in which interval they are classified, e.g., 0-0.1 is classified as the first class, 0.1-0.2 is classified as the second class, and so on.

1) Normalization. Each column of the input is first normalized to the range [0, 1] via equation (16):

$$x' = \frac{x-min(x)}{max(x)-min(x)} \tag{16}$$

where x is the input data, $min(x)$ represents the minimum value of the data and $max(x)$ represents the maximum value of the data.

2) create dictionary labels that map the index of each category to a key in the form of a string.

3) Produce the dataset. Obtain a list of paths and labels based on the dictionary of labels and the number of categories. The last column in the dataset is invoked as a random number, which is sorted according to the set number of categories to get the corresponding label.

C. Process for precision irrigation evaluation

1) Data classification. According to the data classification division method, the soil water content is divided into 10

classes, labelled 0-9 in order, and each class corresponds to a unique irrigation strategy.

2) feature mapping. After data transformation to get x, its data shape is (batch_size, dim), batch_size represents the number of samples extracted from the training data each time, dim represents the number of features. A linear transformation is performed by linear to map the input features from dim to a higher dimension (dimh).

3) Data reshaping. The obtained data ((batch_size, dimh) is reshaped into (batch_size, time_steps, nodes, 2) to fit the input format of graph convolutional network.

4) Graph Convolution Processing. The reshaped data is further reshaped into (batch_size * 10, 4, 2) and fed into the GCN model. Attention transformation is made on the input features using UFOAttention and attention weighted features are computed. Calculate support features support with shape (batch_size * time_steps, nodes, out_features). Graph convolution operation is performed using the adjacency matrix adj and the output shape is (batch_size * time_steps, nodes, out_features). out_features represent the number of output features. The output data is further processed by SPABlock by batch normalisation layer BatchNorm1d and the shape becomes (batch_size * 10, nhid), where nhid represents the hidden layer feature dimension.

5) Activation and reshaping. Data is reshaped to (batch_size, nhid,10) by ReLU activation function with transpose operation.

6) Feature Spreading. Reshape the processed obtained data into (batch_size, time_steps, -1) and transpose it into (batch_size, -1, time_steps), spreading it to fit the input of the linear classifier.

7) linear classification. Map the spread features to the number of categories by linear2, the output shape is (batch_size, num_classes), which represents the category prediction for each sample.

8) Output results. The final output is the category prediction for each sample in the shape of (batch_size, num_classes).

The labels of the data labels are an essential part of the training process used to calculate the loss, but they are not directly involved in the forward propagation of the model. In the training loop, the output of the model is to be sent into the loss function along with the labels, and the loss function calculates the difference between the predicted output and the true labels to generate a loss value. The calculated loss value is used for back propagation to update the parameters of the model to minimise the loss. The model is learnt by adjusting the parameters to make the forecast output closer to the labels.

V. Experiments and analyses

A. Experimental evaluation indicators and settings

1) Experimental Data

The experimental dataset consists of real-world field sensor data collected from Liucheng County state-owned Fuhu Overseas Chinese Farm Tea Factory, a commercial tea plantation located in Liuzhou Guangxi Province, China. A wireless sensor network (WSN) was deployed across the plantation, comprising nodes equipped with sensors for measuring Environmental parameters, soil temperature (ST), soil electrical conductivity (SEC), and soil moisture content (SMC). The environmental parameter collection uses the integrated environmental monitor of Guangzhou Hairui Intelligent Technology Co., Ltd., which can realize the collection of seven elements of air temperature, air humidity, illumination, air pressure, rainfall, wind speed and wind direction. And ST, SEC and SMC are completed by the MEC20 trinity sensor of Dalian Zheqin Technology Co., Ltd. Data was logged at intervals of 15 minutes.

The raw data collection spanned from August 10, 2020, to March 27, 2021, yielding a total of 31,464 raw data records. To analyze and utilize this data more effectively, missing values were addressed using linear interpolation for short gaps; records with consecutive missing values exceeding 1 hour were removed. 700 datasets were obtained by processing missing values during the preprocessing stage. Each dataset sample comprises the synchronized readings of the aforementioned environmental and soil parameters as input features, with the corresponding SMC serving as the target output variable for prediction and subsequent classification.

2) Evaluation Metrics

The model was trained and assessed using Accuracy (A_{cc}), Precision (P_{pv}) and Recall (R_{ec}) and F1-Score, whose explicit definitions are given in equations (17)–(20).

$$A_{cc} = \frac{TP+TN}{TP+TN+FP+FN} \tag{17}$$

$$P_{pv} = \frac{TP}{TP+FP} \tag{18}$$

$$R_{ec} = \frac{TP}{TP+FN} \tag{19}$$

$$\text{F1\_Score} = 2 \times \frac{P_{pv} \times R_{ec}}{P_{pv} + R_{ec}} \tag{20}$$

where TP denotes True Positive, i.e., the number of samples correctly predicted by the model as positive samples; TN denotes True Negative, i.e., the number of samples correctly predicted by the model as negative samples; FP denotes False Positive, i.e., the number of samples incorrectly predicted by the model as positive samples; and FN denotes False Negative (False Negative), i.e., the number of samples that the model incorrectly predicts as negative.

3) Experimental platforms

The MSI Punch Tank GL65 laptop with Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz, 2.59 GHz and 32GB of RAM is used for this experiment. The software environment for the experiment is under Windows 11 system platform using Anaconda development tools and pytorch_ geometric framework to perform the experiments.

4) Experimental settings

In the SPANet attention mechanism module, the main parameters are the number of channels in the input feature map in channels, the value of which depends on the input data; the number of ‘most important’ elements selected at each position of the feature map k = 4 (depending on the dimensionality of the input data); and the flag bit to set the value of k adaptively.

adaptive (set to False), if adaptive is True, then k will be calculated based on N (the spatial dimensionality of the feature map) and reductions, the default value is 16; whether to use k as a parameter for learning (set to False); the mode for calculating the spatial attention, the default is ‘pow’, i.e., the square calculation used to calculate the importance of the data.

The main parameters in the UFOAttention module are: the output dimension of the model d_model, the dimensions of the query vectors (queries) and key vectors (keys) d_k, the dimension of the value vectors (values) d_v the number of main headers h. According to the actual needs of the experiment, they are all set to 2.

The main data in the GCN module of the graphical convolutional neural network are: the number of output classification categories num_classes=10; the input data dimension in_features (depending on the input data), the output data dimension out_features (depending on the input data), the bias flag bit bias (set to True), and the number of hidden layers nhid=16.

B. Algorithms for comparison

On the collected soil data, the proposed method is compared with other models in an experiment. The main models compared are SAGEConv [20], GCNConv [21], GATConv [22], GraphConv [23], FeaStConv [24], ARMAConv [25], SGConv [26], GATv2Conv [27], TransformerConv [28], SSGConv [29], GENConv [30] and SuperGATConv [31].

C. Experimental results and analyses

1) Prediction performance experiment of the algorithm

The prediction performance experiments of the proposed method are conducted on the collected dataset of soil water content.

The confusion matrix of the UFO-GCN-SPANet model run is shown in Fig. 7.

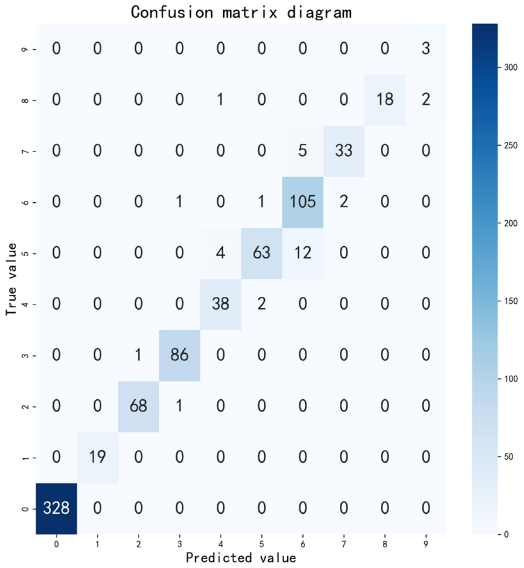


Fig. 7. UFO-GCN-SPANet model prediction confusion matrix

Since multiple classifications are involved, this method calculates the precision, recall and F1 values for each category separately, and then averages them as the final indicator value for the classification. In Table I, the precision, recall and F1 values for each category in the model prediction.

TABLE I
EVALUATION INDICATOR RESULTS BY CATEGORY

Class	P_{pv}	R_{ec}	F1_Score
Class 0	1.0000	1.0000	1.0000
Class 1	1.0000	1.0000	1.0000
Class 2	0.9855	0.9855	0.9855
Class 3	0.9773	0.9885	0.9829
Class 4	0.8837	0.9500	0.9157
Class 5	0.9545	0.7975	0.8690
Class 6	0.8607	0.9633	0.9091
Class 7	0.9429	0.8684	0.9041
Class 8	1.000	0.8571	0.9231
Class 9	0.6000	1.0000	0.7500

Depending on the confusion matrix and Table I, the accuracy of the UFO-GCN-SPANet model prediction is 0.9596, the average precision is 0.9205, the average recall is 0.9410, and the average F1 value is 0. 9239.The results of the experimental data show that the UFO-GCN-SPANet model has superior prediction performance and high accuracy on this soil dataset.

Plot the Receiver Operating Characteristic Curve (ROC Curve, ROC Curve Plot) predicted by the model, as indicated in Fig. 8.

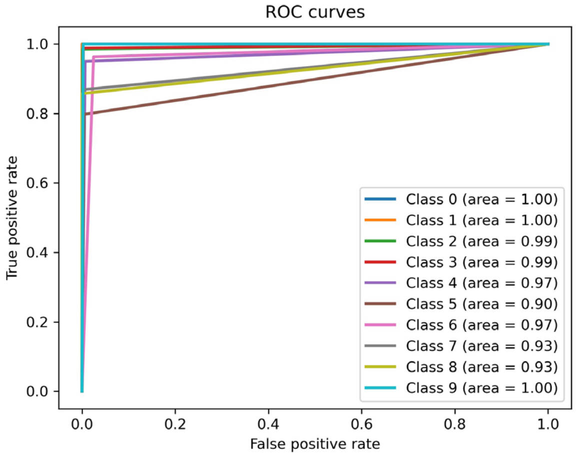


Fig. 8. ROC

As can be observed from Fig. 8, the area under the curve for each output category (Class 0 - Class 9) reaches 1.00 for Class 0, Class 1 and Class 9, with the lowest being Class 5, which is only 0.9, and above 0.9 for all other categories. It is shown that the prediction accuracy of each category is at a high level, indicating that the model predicts better.

2) Performance comparison experiments with other algorithms

The UFO-GCN-SPANet model was comparable to the more typical current models based on graph structure analysis, SAGEConv, GCNConv, GATConv, GraphConv, FeaStConv, ARMAConv, SGConv, GATv2Conv, TransformerConv, SSGConv, GENConv, and SuperGATConv were used for comparison experiments. The experimental results are given in Table II. As can be seen from Table II, in terms of the performance of prediction, the models based on graph structure processing all achieve better prediction results, with A_{cc} above 0.8, of which the GCNConv model has the lowest accuracy of 0.8562, and the GCN model optimised by introducing the unit forcing operation visual Transformer and the self-attention mechanism of the salient position (UFO-GCN-SPANet), the accuracy reaches 0.9596, an improvement of 12.07% compared to GCNConv. This significant improvement indicates that the prediction performance of the model can be effectively improved by the optimised UFO-GCN-SPANet model. In terms of P_{pv}, R_{ec} and F1_Score values, the performance of most of the models is concentrated around 0.75, among which the GENConv and UFO-GCN-SPANet models are relatively outstanding, with all the three metrics reaching above 0.8. In particular, the UFO-GCN-SPANet model achieves 0.9205, 0.9410, and 0.9239 in in P_{pv}, R_{ec} and F1_Score values, respectively, which is an improvement of 27.88% in P_{pv}, R_{ec} and F1_Score values compared to the lowest of the models, the GCNConv model, 33.26% and 30.29%. This result shows that the UFO-GCN-SPANet model is the optimal model among all the experimented models, which has a significant advantage in processing the data and is also reflected in the superiority in prediction accuracy.

TABLE II
PREDICTIVE PERFORMANCE INDICATORS OF THE MODELS

Class	A_{cc}	P_{pv}	R_{ec}	F1_Score
SAGEConv	0.8941	0.7597	0.7473	0.7504
GCNConv	0.8562	0.7198	0.7061	0.7091
GATConv	0.8764	0.7358	0.7329	0.7312
GraphConv	0.8752	0.7503	0.7080	0.7147
FeaStConv	0.8878	0.7537	0.7423	0.7466
ARMAConv	0.8777	0.7477	0.7336	0.7390
SGConv	0.8663	0.7492	0.7077	0.7196
GATv2Conv	0.8916	0.7745	0.7359	0.7494
TransformerConv	0.8916	0.7548	0.7626	0.7568
SSGConv	0.8840	0.7662	0.7398	0.7457
GENConv	0.9092	0.8872	0.7987	0.8210
SuperGATConv	0.8916	0.7570	0.7548	0.7553
UFO-GCN-SPANet	0.9596	0.9205	0.9410	0.9239

As evidenced in Table II, the proposed UFO-GCN-SPANet model achieves state-of-the-art performance across all evaluation metrics (Accuracy, Precision, Recall, F1-Score) compared to a wide range of established graph neural network baselines. This significant performance advantage, exemplified by the 12.07% improvement in Accuracy over GCNConv and substantial leads over other strong contenders like GATConv and GraphConv, unequivocally validates the effectiveness of the overall architecture. This superior performance is directly attributable to the core innovations embedded within UFO-GCN-SPANet. The cascaded integration of these specifically designed components (UFO-ViT -> GCN -> SPANet) creates a synergistic effect. UFO-ViT lays the groundwork with efficient global understanding, GCN incorporates the spatial sensor topology, and SPANet performs targeted, context-aware refinement. This holistic design, prioritizing both predictive accuracy and operational efficiency, is the cornerstone of UFO-GCN-SPANet's success and represents its key novelty in addressing the practical demands of precision irrigation systems.

D. Ablation Study

To rigorously evaluate the contribution of each key innovative component in the proposed UFO-GCN-SPANet architecture, we conduct a comprehensive ablation study. We systematically remove or substitute core modules and evaluate the performance on the same soil moisture dataset. The variants compared are:

UFO-GCN-SPANet (Full Model): Our proposed complete architecture.

UFO-GCN: Replace UFO-ViT with a standard Transformer layer (using softmax attention).

GCN-SPANet: Replace SPANet with a standard multi-head self-attention layer.

GCN Only: Use only the GCN module as the core model.

The results of the ablation experiment are shown in Table III.

TABLE III
ABLATION STUDY RESULTS (PERFORMANCE ON TEST SET)

Model Variant	A_{cc}	P_{pv}	R_{ec}	F1_Score
GCN	0.8675	0.8698	0.8676	0.8625
UFO-GCN	0.8978	0.9034	0.8978	0.8969
GCN-SPANet	0.9294	0.9308	0.9294	0.9281
UFO-GCN-SPABlock	0.9584	0.9587	0.9584	0.9581

As can be seen from the experimental results in Table III, UFO-GCN-SPABlock performed the best, with indicators of 0.9584, 0.9587, 0.9584 and 0.9581, respectively, which were significantly better than other models, indicating that the proposed method had advantages in comprehensive performance and was the most accurate in classifying the overall data. UFO-GCN, which removing SPANet entirely results in a further performance degradation compared to the full model. This illustration suggests that SPANet's SPS algorithm is not just an alternative but a superior mechanism that offers significant performance gains and critical computational savings. The GCN Only baseline performs substantially worse than the full model and even the ablated variants containing attention, underscoring the necessity of incorporating sophisticated global and contextual modeling. Adding UFO on top of GCN will increase all accuracy metrics compared to GCN, confirming that the linear complexity of UFO-ViT is critical to achieving high performance and practical efficiency in global feature extraction. The ablation study provides strong empirical evidence that the proposed UFO-ViT module and SPANet module (with its SPS algorithm), integrated in the specific cascaded order, are indispensable and synergistic innovations within the UFO-GCN-SPANet architecture. Each component contributes significantly to the model's final high accuracy and efficiency, justifying the novelty and design choices of the proposed framework.

E. Computational Efficiency and Performance Analysis

Experiments executed on consumer-grade hardware (MSI GL65 laptop, 32GB RAM). We compare the underlying algorithm GCN with the model we extracted from the total training time, Average CPU usage, Average memory usage, Average GPU usage,Average GPU memory usage and Best validation accuracy.The results of the ablation experiment are shown in the Table IV.

Table IV
Computational Efficiency and Performance Analysis of the Proposed Model

Evaluation indicators	GCN	Proposed Model	Delta
Total training time	218.62 s	558.57s	+339.95
Average CPU usage:	0.30%	1.25%	+0.95%
Average memory usage	52.50%	52.22%	-0.28%
Average GPU usage	11.99%	13.57%	+1.58%
Average GPU memory usage	14.06%	14.73%	+0.67%
Best validation accuracy	0.9168	0.9289	+1.21%

As can be seen from Table IV, the proposed UFO-GCN-SPBlock architecture achieves a notable accuracy improvement of 1.21% over the baseline GCN model, at the cost of a 2.6x increase in training time. Interestingly, we observed remarkably low hardware utilization rates (CPU ~1%, GPU ~13%) during training. This indicates that the current training pipeline is likely constrained by I/O bottlenecks (e.g., data loading and preprocessing) rather than the computational intensity of the model itself. This presents a significant opportunity for acceleration through pipeline optimization. Furthermore, the minimal change in GPU memory consumption (14.06% to 14.73%) suggests that our model's increased complexity is primarily computational, not parametric, making it a suitable candidate for memory-constrained edge deployment scenarios once the computational efficiency is improved.

VI. Conclusion and discussion

Soil moisture content plays a major role in agricultural production and water management, and is directly related to crop growth and irrigation efficiency. Accurate prediction of soil water content enables scientific irrigation and optimization of water resources. In tea tree precision irrigation, when making the decision of precision irrigation, farmers are not concerned about the specific soil water content data, but more concerned about whether the soil is in the degree of drought, moderate or over-wet state, and then dynamically adjust the amount of irrigation water according to the value of this degree. Therefore, changing the soil water content prediction problem into a classification problem and formulating an irrigation strategy based on the prediction results can achieve precision irrigation more effectively.

This manuscript proposes the innovative and efficient UFO-GCN-SPANet architecture for precise irrigation evaluation, with demonstrated compatibility for integration with IoT platforms using LoRaWAN, 4G/5G communication protocols and MODBUS sensor interfaces. The core novelty resides in three interconnected contributions:

The pioneering integration of UFO-ViT's linear-complexity self-attention (O(N)) for computationally efficient yet powerful global spatio-temporal feature extraction, directly overcoming the O(N^2) bottleneck in standard approaches.

The introduction of SPANet with its novel Salient Position Selection (SPS) algorithm, which dynamically focuses computation on the most relevant contextual features, achieving significant gains in both predictive performance and computational/memory efficiency compared to standard attention mechanisms.

The design and validation of an optimal cascaded structure (UFO-ViT -> GCN -> SPANet) that sequentially leverages global context, spatial relationships, and focused refinement, specifically tailored to the efficiency and accuracy requirements of resource-constrained agricultural IoT applications. Extensive experiments and ablation studies confirm the substantial individual and synergistic value of each component. This work demonstrates that intelligent architectural design, featuring purpose-built efficient modules, is key to enabling high-performance AI solutions for real-world precision agriculture.

While the proposed UFO-GCN-SPANet demonstrates strong performance, several limitations should be acknowledged:

Sensor Data Quality. The model's accuracy may degrade with missing or noisy sensor inputs, particularly for soil conductivity measurements which showed ±15% variability in our field tests.

Environmental Factors. Extreme weather conditions (e.g., heavy rainfall >50 mm/day) may temporarily disrupt LoRa communications as shown in Fig. 2, requiring manual data imputation.

Crop Specificity. Current validation is limited to tea plants, though the architecture is theoretically adaptable to other crops.

Temporal Resolution. The 15-minute sampling interval may miss rapid moisture changes during irrigation events, suggesting future work should explore higher-frequency sensing.

System Integration. The proposed framework maintains compatibility with commercial IoT irrigation platforms through MODBUS sensor interfaces and OpenADR control outputs, though field deployment may require gateway-level protocol adaptation, suggesting future work should develop plug-and-play middleware for heterogeneous systems.

Enhance practicality and scalability. This research can be undertaken in four primary dimensions. 1) Training Pipeline Optimization: We will eliminate the identified I/O bottleneck by implementing asynchronous data loading, prefetching, and on-GPU preprocessing to drastically reduce training overhead. 2) Large-Scale Validation: We will rigorously benchmark the model's scalability on massive sensor networks (>100,000 nodes) to assess real-world applicability. 3) Model Compression: Techniques like pruning and quantization will be explored to create lightweight variants for edge deployment. 4) Architectural Refinement: We will investigate more efficient attention mechanisms to reduce computational costs while preserving performance gains. These steps will directly address the trade-off between accuracy and efficiency, advancing the model toward practical agricultural implementation.

References

[1] A. Puspaningrum, A. Sumarudin and W.P. Putra. "Irrigation Prediction using Machine Learning in Precision Agriculture," in Proc. 5th International Conference of Computer and Informatics Engineering (IC2IE), Jakarta, Indonesia, 2022, pp. 204-208, doi: 10.1109/IC2IE56416.2022.9970092.

[2] N.M. Priya, G. Amudha, M. Dhurgadevi, N. Malathi, K. Balakrishnan and M. Preetha. "IoT and Machine Learning based Precision Agriculture through the Integration of Wireless Sensor Networks," Journal of Electrical Systems, vol.20, no.4s, pp. 2292-2299, 2024. DOI: 10.52783/jes.2399

[3] R. Filgueiras, T.S. Almeida, E.C. Mantovani, S.H.B. Dias, E. I Fernandes-Filho, F.F.da Cunha and L.P. Venancio."Soil water content and actual evapotranspiration predictions using regression algorithms and remote sensing data," Agricultural Water Management, vol.241, pp.106346, 2020. DOI: 10.1016/j.agwat.2020.106346

[4] E.A. Abioye, M.S.Z. Abidin, M.S.A. Mahmud, S. Buyamin, M.K.I. AbdRahman, A.O. Otuoze, M.S.A. Ramli and O.D. Ijike "IoT-based monitoring and data-driven modelling of drip irrigation system for mustard leaf cultivation experiment," Information Processing in Agriculture, vol.8, no.2, pp.270-283, 2021.DOI: 10.1016/j.inpa.2020.05.004

[5] L.M. Narakala, A. Yadav, H. Upreti and G. Das Singhal. "Prediction of Crop Water Stress Index (CWSI) Using Machine Learning Algorithms," in Proc. World Environmental and Water Resources Congress, Milwaukee, Wisconsin, 2024, pp. 969-980. DOI: 10.1061/9780784485477.086

[6] E. Bwambale, F.K. Abagale and G.K. Anornu. "Data-driven model predictive control for precision irrigation management," Smart Agricultural Technology, vol.3, pp.100074, 2023. DOI: 10.1016/j.atech.2022.100074

[7] A. Bhoi, R.P. Nayak, S.K. Bhoi and S. Sethi. "Automated precision irrigation system using machine learning and IoT," in Proc. Intelligent Systems: Proceedings of ICMIB 2020, Singapore, 2021, pp.275-282. DOI: 10.1061/9780784485477.086

[8] D.M.M. Vianny, A. John, S.K. Mohan, A.Sarlan, Adimoolam and A, Ahmadian. "Water optimization technique for precision irrigation system using IoT and machine learning," Sustainable Energy Technologies and Assessments, vol.52, pp.102307, 2022. DOI: 10.1016/j.seta.2022.102307

[9] G.S. Campos N, A.R. Rocha, R. Gondim, T.L. Coelho da Silva and D.G. Gomes. "Smart & green: An internet-of-things framework for smart irrigation," Sensors, vol.20, no.1, pp.190,2019. DOI: 10.3390/s20010190

[10] R. Liao, S. Zhang, X. Zhang, M, Wang, H, Wu and L. Zhangzhong. "Development of smart irrigation systems based on real-time soil moisture data in a greenhouse: Proof of concept," Agricultural Water Management, vol.245, pp.106632. 2021. DOI: https://doi.org/10.1016/j.agwat.2020.106632

[11] Y.B. Choi and J.H. Shin. "Development of a transpiration model for precise irrigation control in tomato cultivation," Scientia Horticulture, vol. 267, pp. 109358, 2020. DOI: 10.1016/j.scienta.2020.109358

[12] W.J. Jo and J.H. Shin. "Development of a transpiration model for precise tomato (Solanum Lycopersicon L.) irrigation control under various environmental conditions in greenhouse," Plant Physiology and Biochemistry, vol.162, pp. 388-394, 2021. DOI: 10.1016/j.plaphy.2021.03.005

[13] M. Chen, Y. Sun, X. Cai, B. Liu and T. Ren. "Design and implementation of a novel precision irrigation robot based on an intelligent path planning algorithm," arXiv preprint, arXiv:2003.00676, 2020. DOI: 10.48550/arXiv.2003.00676

[14] K. Wu, H. Desesquelles, R. Cockenpot, L. Guyard, V.Cuisiniez and S. Lambot. "Ground-penetrating radar full-wave inversion for soil moisture mapping in Trench-Hill potato fields for precise irrigation," Remote Sensing, vol.14, no.23, pp.6046, 2022. DOI: 10.3390/rs14236046

[15] A. Abera, N.E.C Verhoest, S. Tilahun and J. Nyssen "Assessment of irrigation expansion and implications for water resources by using RS and GIS techniques in the Lake Tana Basin of Ethiopia," Environmental monitoring and assessment, vol.193, pp.1-17, 2021. DOI: 10.1007/s10661-020-08778-1

[16] T.N. Kipf and M. Welling. "Semi-supervised classification with gra ph convolutional networks," arXiv preprint, arXiv:1609.02907, 2016.DOI: 10.48550/arXiv.1609.02907

[17] M. Defferrard, X. Bresson and P. Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering," arXiv preprint arXiv:1606.09375, 2016. Doi: 10.48550/arXiv.1606.09375

[18] J. Song. "Ufo-vit: High performance linear vision transformer without softmax," arXiv preprint, arXiv:2109.14382, 2021.DOI: 10.48550/arXiv.2109.14382

[19] S.Fang, K. Li and Z. Li. "Salient positions based attention network for image classification," arXiv preprint, arXiv:2106.04996, 2021.DOI: 10.48550/arXiv.2106.04996

[20] W.L. Hamilton, R. Ying and J. Leskovec. "Inductive representation learning on large graphs," arXiv preprint arXiv:1706.02216, 2017. Doi: 10.48550/arXiv.1706.02216

[21] T.N. Kipf and M. Welling. "Semi-supervised classification with graph convolutional networks," arXiv preprint, arXiv:1609.02907, 2016. DOI: 10.48550/arXiv.1609.02907

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio "Graph attention networks," arXiv preprint, arXiv:1710.10903, 2017. DOI: 10.48550/arXiv.1710.10903

[23] C. Morris, M. Ritzert, M. Fey, W.L. Haniton, J.E. Lenssen, G.Rattan and M. Grohe. "Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks," arXiv preprint, arXiv:1810.02244, 2018. DOI: 10.1609/aaai.v33i01.33014602

[24] N. Verma, E. Boyer and J.Verbeek. "FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis," arXiv preprint, arXiv:1706.05206, 2017. DOI: 10.48550/arXiv.1706.05206

[25] F.M. Bianchi, D. Grattarola, L. Livi and C. Alippi. "Graph neural networks with convolutional arma filters," IEEE transactions on pattern analysis and machine intelligence, vol.44, no.7, pp.3496-3507, 2021. DOI: 10.1109/TPAMI.2021.3054830

[26] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu and K. Weinberger. "Simplifying Graph Convolutional Networks," arXiv preprint, arXiv:1902.07153, 2019. DOI: 10.48550/arXiv.1902.07153

[27] S. Brody, U. Alon and E. Yahav. "How attentive are graph attention networks?,"arXiv preprint, arXiv:2105.14491, 2021. DOI: 10.48550/arXiv.2105.14491

[28] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang and Y. Sun. "Masked label prediction: Unified message passing model for semi-supervised classification," arXiv preprint, arXiv:2009.03509, 2020. DOI: 10.48550/arXiv.2009.03509

[29] H. Zhu and P. Koniusz. "Simple spectral graph convolution," in proc. International Conference on Learning Representations (ICLR 2021), Vienna, Austria, 2021.

[30] G. Li, C. Xiong, A. Thabet and B. Ghanem. "Deepergcn: All you need to train deeper gcns," arXiv preprint, arXiv:2006.07739, 2020. DOI: 10.48550/arXiv.2006.07739

[31] D. Kim and Oh A. "How to find your friendly neighborhood: Graph attention design with self-supervision," arXiv preprint, arXiv:2204.04879, 2022. DOI: 10.48550/arXiv.2204.04879

Preparation of Papers for Electronics (September 2011)

First A. Author, Second B. Author, and Third C. Author

***Abstract*—These instructions give you guidelines for preparing papers for ELECTRONICS journal. Use this document as a template if you are using Microsoft *Word* 6.0 or later. Otherwise, use this document as an instruction set. The electronic file of your paper will be formatted further. Define all symbols used in the abstract. Do not cite references in the abstract. Do not delete the blank line immediately above the abstract; it sets the footnote at the bottom of this column.**

***Index Terms*—About four key words or phrases in alphabetical order, separated by commas.**

Paper Classification
DOI: 10.7251/ELSxxxxxxxx

I. Introduction

THIS document is a template for Microsoft *Word* versions 6.0 or later.

When you open the file, select "Page Layout" from the "View" menu in the menu bar (View | Page Layout), which allows you to see the footnotes. Then, type over sections of file or cut and paste from another document and use markup styles. The pull-down style menu is at the left of the Formatting Toolbar at the top of your *Word* window (for example, the style at this point in the document is "Text"). Highlight a section that you want to designate with a certain style, then select the appropriate name on the style menu. The style will adjust your fonts and line spacing. **Do not change the font sizes or line spacing to squeeze more text into a limited number of pages.** Use italics for emphasis; do not underline. The length of the manuscript is limited to the maximum of 15 pages.

To insert images in *Word,* position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with "float over text" unchecked).

We will do the final formatting of your paper.

II. Procedure for Paper Submission

A. Review Stage

The manuscripts are to be submitted using the Electronics Journal online submission system – accessible from Journal's homepage. Prepare it in two-column format as shown in this template. Place all figures and tables at the end of the paper (after the references) on separate page(s). Figures and tables must have the same caption names as referenced in the text. Only PDF format of the manuscript is allowed at the review stage. Please, check if all fonts are embedded and subset and that the quality of diagrams, illustrations, and graphics is satisfactory. Failing to provide above listed requirements is a valid reason for rejection.

B. Final Stage

When you submit your final version (after your paper has been accepted), prepare it in two-column format, including figures and tables in accordance with this template. Pack all of your files (manuscript source file in *Word*, figures, and manuscript PDF form) within one archive file (you may use any of the available file compression tools: *WinZip*, *WinRAR*, *7-Zip*, etc.). Do not forget to provide the manuscript converted in PDF format that will be used as a reference for final formatting of your paper. Figures should be named as referenced in the manuscript (e.g. *fig1.eps*, *fig2.tif*, etc.)

C. Figures and Tables

Format and save your graphic images using a suitable graphics processing program and adjusts the resolution settings. We accept images in the following formats: PS, EPS, TIFF, GIF, and PNG. Additionally, it is allowed to use images generated by using one of the following software tools: Microsoft Word, Microsoft PowerPoint, or Microsoft Excel. The resolution of a RGB color file should be 400 dpi. Please note that JPG and other lossy-compressed image formats are not allowed. Use available software tools to convert these images to appropriate format.

Image quality is very important to how yours graphics will reproduce. Even though we can accept graphics in many formats, we cannot improve your graphics if they are poor quality when we receive them. If your graphic looks low in

Manuscript received 15 September 2011 (write the date when you have first sent the manuscript for review). Received in revised form 20 October 2011 (write the date when you have sent the manuscript in its revised form if revisions required for your paper after review).

(Place here any sponsor and financial support acknowledgments).

F. A. Author is with the Faculty of Electrical Engineering, University of Banja Luka, Banja Luka, Bosnia and Herzegovina (corresponding author to provide phone: +387-51-222-333; fax: +387-51-111-222; e-mail: author@etfbl.net).

S. B. Author was with Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia. He is now with the Institute "Mihailo Pupin", Belgrade, Serbia (e-mail: author@pupin.rs).

T. C. Author is with the School of Electrical Engineering, University of Belgrade, Belgrade, Serbia, on leave from the Faculty of Electronic Engineering, University of Niš, Niš, Serbia (e-mail: author@elfak.ni.ac.rs).

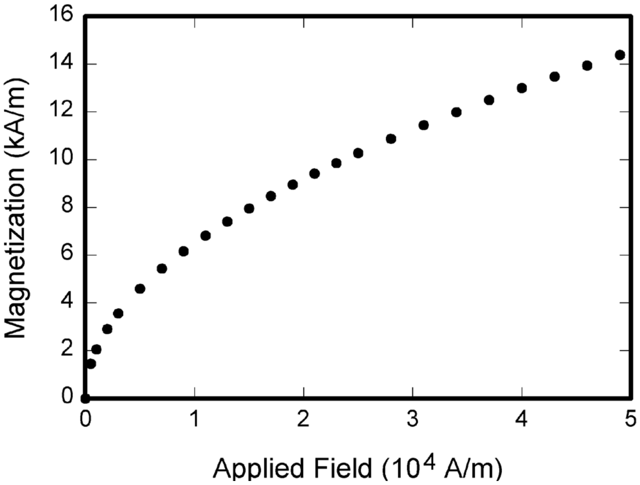


Fig. 1. Magnetization as a function of applied field. Note that "Fig." is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

TABLE I
Units for Magnetic Properties

Symbol	Quantity	Conversion from Gaussian and CGS EMU to SI [a]
Φ	magnetic flux	1 Mx → 10^{-8} Wb = 10^{-8} V·s
B	magnetic flux density, magnetic induction	1 G → 10^{-4} T = 10^{-4} Wb/m^2
H	magnetic field strength	1 Oe → $10^3/(4\pi)$ A/m
m	magnetic moment	1 erg/G = 1 emu → 10^{-3} A·m^2 = 10^{-3} J/T
M	magnetization	1 erg/(G·cm^3) = 1 emu/cm^3 → 10^3 A/m
$4\pi M$	magnetization	1 G → $10^3/(4\pi)$ A/m
σ	specific magnetization	1 erg/(G·g) = 1 emu/g → 1 A·m^2/kg
j	magnetic dipole moment	1 erg/G = 1 emu → $4\pi \times 10^{-10}$ Wb·m
J	magnetic polarization	1 erg/(G·cm^3) = 1 emu/cm^3 → $4\pi \times 10^{-4}$ T
χ, κ	susceptibility	1 → 4π
χ_ρ	mass susceptibility	1 cm^3/g → $4\pi \times 10^{-3}$ m^3/kg
μ	permeability	1 → $4\pi \times 10^{-7}$ H/m = $4\pi \times 10^{-7}$ Wb/(A·m)
μ_r	relative permeability	$\mu \rightarrow \mu_r$
w, W	energy density	1 erg/cm^3 → 10^{-1} J/m^3
N, D	demagnetizing factor	1 → $1/(4\pi)$

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

[a] Gaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

quality on your printer or monitor, please keep in mind that cannot improve the quality after submission.

If you are importing your graphics into this Word template, please use the following steps:

Under the option EDIT select PASTE SPECIAL. A dialog box will open, select paste picture, then click OK. Your figure should now be in the Word Document.

If you are preparing images in TIFF, EPS, or PS format, note the following. High-contrast line figures and tables should be prepared with 600 dpi resolution and saved with no compression, 1 bit per pixel (monochrome).

Photographs and grayscale figures should be prepared with 300 dpi resolution and saved with no compression, 8 bits per pixel (grayscale).

Most charts graphs and tables are one column wide (3 1/2 inches or 21 picas) or two-column width (7 1/16 inches, 43 picas wide). We recommend that you avoid sizing figures less than one column wide, as extreme enlargements may distort your images and result in poor reproduction. Therefore, it is better if the image is slightly larger, as a minor reduction in size should not have an adverse affect the quality of the image.

III. Math

If you are using *Word,* use either the Microsoft Equation Editor or the *MathType* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* MathType Equation). "Float over text" should *not* be selected.

IV. Units

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). **This applies to papers in data storage.** For example, write "15 Gb/cm^2 (100 Gb/in^2)." An exception is when English units are used as identifiers in trade, such as "3½-in disk drive." Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The SI unit for magnetic field strength H is A/m. However, if you wish to use units of T, either refer to magnetic flux density B or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., "A·m^2."

V. Helpful Hints

A. Figures and Tables

Because we will do the final formatting of your paper, you do not need to position figures and tables at the top and bottom of each column. In fact, all figures, figure captions, and tables can be at the end of the paper. Large figures and tables may span both columns. Place figure captions below the figures; place table titles above the tables. If your figure has two parts, include the labels "(a)" and "(b)" as part of the artwork. Please verify that the figures and tables you mention in the text actually exist. **Please do not include captions as part of the figures. Do not put captions in "text boxes" linked to the figures. Do not put borders around the outside of your figures.** Use the abbreviation "Fig." even at the beginning of a sentence. Do not abbreviate "Table." Tables are numbered with Roman numerals.

Color printing of figures is not available Do not use color unless it is necessary for the proper interpretation of your figures.

Figure axis labels are often a source of confusion. Use words rather than symbols. As an example, write the quantity "Magnetization," or "Magnetization *M*," not just "*M*." Put units in parentheses. Do not label axes only with units. As in Fig. 1, for example, write "Magnetization (A/m)" or "Magnetization ($A \cdot m^{-1}$)," not just "A/m." Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)," not "Temperature/K."

Multipliers can be especially confusing. Write "Magnetization (kA/m)" or "Magnetization (10^3 A/m)." Do not write "Magnetization (A/m) x 1000" because the reader would not know whether the top axis label in Fig. 1 meant 16000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type.

B. References

Number citations consecutively in square brackets [1]. The sentence punctuation follows the brackets [2]. Multiple references [2], [3] are each numbered with separate brackets [1]–[3]. When citing a section in a book, please give the relevant page numbers [2]. In sentences, refer simply to the reference number, as in [3]. Do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] shows ... ." Please do not use automatic endnotes in *Word*, rather, type the reference list at the end of the paper using the "References" style.

Number footnotes separately in superscripts (Insert | Footnote).[1] Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes (see Table I).

Please note that the references at the end of this document are in the preferred referencing style. Give all authors' names; do not use "*et al*." unless there are six authors or more. Use a space after authors' initials. Papers that have not been published should be cited as "unpublished" [4]. Papers that have been accepted for publication, but not yet specified for an issue should be cited as "to be published" [5]. Papers that have been submitted for publication should be cited as "submitted for publication" [6]. Please give affiliations and addresses for private communications [7].

Capitalize only the first word in a paper title, except for proper nouns and element symbols. For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [8]. All references **must be** written in Roman alphabet.

C. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have already been defined in the abstract. Abbreviations such as IEEE, SI, ac, and dc do not have to be defined. Abbreviations that incorporate periods should not have spaces: write "C.N.R.S.," not "C. N. R. S." Do not use abbreviations in the title unless they are unavoidable (for example, "IEEE" in the title of this article).

D. Equations

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). First use the equation editor to create the equation. Then select the "Equation" markup style. Press the tab key and write the equation number in parentheses. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence, as in

$$\begin{aligned}\int_0^{r_2} & F(r,\varphi)\, d \quad d\varphi = [\sigma r_2 / (2\mu_0] \\ & \cdot \int_0^{\infty} \exp(-\lambda \,|z_j - z_i|)\, \lambda^{-1} J_1(\lambda r_2)\, J_0(\lambda r_i)\, d\lambda\,.\end{aligned} \tag{1}$$

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (*T* might refer to temperature, but T is the unit tesla). Refer to "(1)," not "Eq. (1)" or "equation (1)," except at the beginning of a sentence: "Equation (1) is ... ."

E. Other Recommendations

Use one space after periods and colons. Hyphenate complex modifiers: "zero-field-cooled magnetization." Avoid dangling participles, such as, "Using (1), the potential was calculated." [It is not clear who or what used (1).] Write instead, "The potential was calculated by using (1)," or "Using (1), we calculated the potential."

Use a zero before decimal points: "0.25," not ".25." Use "cm^3," not "cc." Indicate sample dimensions as "0.1 cm x 0.2 cm," not "0.1 x 0.2 cm^2." The abbreviation for "seconds" is "s," not "sec." Do not mix complete spellings and abbreviations of units: use "Wb/m^2" or "webers per square meter," not "webers/m^2." When expressing a range of values, write "7 to 9" or "7-9," not "7~9."

A parenthetical statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.) In American English, periods and commas are within quotation marks, like "this period." Other punctuation is "outside"! Avoid contractions; for example, write "do not" instead of "don't." The serial comma is preferred: "A, B, and C" instead of "A, B and C."

If you wish, you may write in the first person singular or plural and use the active voice ("I observed that ..." or "We observed that ..." instead of "It was observed that ..."). Remember to check spelling. If your native language is not English, please get a native English-speaking colleague to carefully proofread your paper.

[1] It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date and authors' affiliations on the first page). Instead, try to integrate the footnote information into the text.

VI. Some Common Mistakes

The word “data” is plural, not singular. The subscript for the permeability of vacuum μ_0 is zero, not a lowercase letter “o.” The term for residual magnetization is “remanence”; the adjective is “remanent”; do not write “remnance” or “remnant.” Use the word “micrometer” instead of “micron.” A graph within a graph is an “inset,” not an “insert.” The word “alternatively” is preferred to the word “alternately” (unless you really mean something that alternates). Use the word “whereas” instead of “while” (unless you are referring to simultaneous events). Do not use the word “essentially” to mean “approximately” or “effectively.” Do not use the word “issue” as a euphemism for “problem.” When compositions are not specified, separate chemical symbols by en-dashes; for example, “NiMn” indicates the intermetallic compound $Ni_{0.5}Mn_{0.5}$ whereas “Ni–Mn” indicates an alloy of some composition Ni_xMn_{1-x}.

Be aware of the different meanings of the homophones “affect” (usually a verb) and “effect” (usually a noun), “complement” and “compliment,” “discreet” and “discrete,” “principal” (e.g., “principal investigator”) and “principle” (e.g., “principle of measurement”). Do not confuse “imply” and “infer.”

Prefixes such as “non,” “sub,” “micro,” “multi,” and “ultra” are not independent words; they should be joined to the words they modify, usually without a hyphen. There is no period after the “et” in the Latin abbreviation “*et al.*” (it is also italicized). The abbreviation “i.e.,” means “that is,” and the abbreviation “e.g.,” means “for example” (these abbreviations are not italicized).

An excellent style manual and source of information for science writers is [9].

VII. Editorial Policy

Each manuscript submitted is subjected to the following review procedure:

- It is reviewed by the editor for general suitability for this publication
- If it is judged suitable, two reviewers are selected and a single-blinded review process takes place
- Based on the recommendations of the reviewers, the editor then decides whether the particular paper should be accepted as is, revised or rejected.

Do not submit a paper you have submitted or published elsewhere. Do not publish “preliminary” data or results. The submitting author is responsible for obtaining agreement of all coauthors and any consent required from sponsors before submitting a paper. It is the obligation of the authors to cite relevant prior work.

Every paper submitted to “Electronics” journal are single-blind reviewed. For conference-related papers, the decision to accept or reject a paper is made by the conference editors and publications committee; the recommendations of the referees are advisory only. Undecipherable English is a valid reason for rejection.

VIII. Publication Principles

The contents of “Electronics” are peer-reviewed and archival. The “Electronics” publishes scholarly articles of archival value as well as tutorial expositions and critical reviews of classical subjects and topics of current interest.

Authors should consider the following points:

1) Technical papers submitted for publication must advance the state of knowledge and must cite relevant prior work.
2) The length of a submitted paper should be commensurate with the importance, or appropriate to the complexity, of the work. For example, an obvious extension of previously published work might not be appropriate for publication or might be adequately treated in just a few pages.
3) Authors must convince both peer reviewers and the editors of the scientific and technical merit of a paper; the standards of proof are higher when extraordinary or unexpected results are reported.
4) Because replication is required for scientific progress, papers submitted for publication must provide sufficient information to allow readers to perform similar experiments or calculations and use the reported results. Although not everything need be disclosed, a paper must contain new, useable, and fully described information. For example, a specimen’s chemical composition need not be reported if the main purpose of a paper is to introduce a new measurement technique. Authors should expect to be challenged by reviewers if the results are not supported by adequate data and critical details.
5) Papers that describe ongoing work or announce the latest technical achievement, which are suitable for presentation at a professional conference, may not be appropriate for publication in “Electronics”.

IX. Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Appendix

Appendixes, if needed, appear before the acknowledgment.

Acknowledgment

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” **Sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.**

References

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[5] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.

[6] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].

[9] M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, Jul. 1993.

[12] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[13] S. P. Bingulac, "On the compatibility of adaptive controllers (Published Conference Proceedings style)," in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.

[14] G. R. Faulhaber, "Design of service systems with priority reservation," in *Conf. Rec. 1995 IEEE Int. Conf. Communications,* pp. 3–8.

[15] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.

[16] G. W. Juette and L. E. Zeffanella, "Radio noise currents n short sections on bundle conductors (Presented Conference Paper style)," presented at the IEEE Summer power Meeting, Dallas, TX, Jun. 22–27, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, "An analysis of surface-detected EMG as an amplitude-modulated noise," presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.

[21] IEEE Criteria for Class IE Electric Systems (Standards style), IEEE Standard 308, 1969.

[22] Letter Symbols for Quantities, ANSI Standard Y10.5-1968.

[23] R. E. Haskell and C. T. Case, "Transient signal propagation in lossless isotropic plasmas (Report style)," USAF Cambridge Res. Lab., Cambridge, MA Rep. ARCRL-66-234 (II), 1994, vol. 2.

[24] E. E. Reber, R. L. Michell, and C. J. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (420-46)-3, Nov. 1988.

[25] (Handbook style) *Transmission Systems for Communications,* 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.

[26] *Motorola Semiconductor Data Manual,* Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

[27] (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume (issue). Available: http://www.(URL)

[28] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com

[29] (Journal Online Sources style) K. Author. (year, month). Title. *Journal* [Type of medium]. Volume(issue), paging if given. Available: http://www.(URL)

[30] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876–880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar

Information for Authors

Editorial objectives

In the journal ”Electronics”, the scientific papers from different fields of electronics and electrical engineering in the broadest sense are published. Main topics are electronics, automatics, telecommunications, computer techniques, power engineering, nuclear and medical electronics, analysis and synthesis of electronic circuits and systems, new technologies and materials in electronics etc. The main emphasis of papers should be on methods and new techniques, or the application of existing techniques in a novel way.

The reviewing process

Each manuscript submitted is subjected to the following review procedures:

- It is reviewed by the editor for general suitability for this publication;
- If it is judged suitable, two reviewers are selected and a double-blind review process takes place;
- Based on the recommendations of the reviewers, the editor then decides whether the particular paper should be accepted as it is, revised or rejected.

Submissions Process

The manuscripts are to be submitted using the Electronics Journal online submission system – accessible from Journal’s homepage.

Manuscripts have to be prepared in accordance with the instructions given in the template for paper preparation that can be found on the journal’s web page (www.els-journal.etf.unibl.org).

Authors should note that proofs are not supplied prior to publication and ensure that the paper submitted is complete and in its final form.

Copyright

Articles submitted to the journal should be original contributions and should not be under consideration for any other publication at the same time. Authors submitting articles for publication warrant that the work is not an infringement of any existing copyright and will indemnify the publisher against any breach of such warranty. For ease of dissemination and to ensure proper policing of use, papers and contributions become the legal copyright of the publisher unless otherwise agreed.