DOI: 10.17725/rensit.2022.14.027

Classical and intelligent approaches for VLSI static timing analysis: an overview

Nikolay O. Vasilyev, Mariya A. Zapletina, Galina A. Ivanova

Institute for Design Problems in Microelectronic of Russian Academy of Sciences, http://www.ippm.ru/

Zelenograd, Moscow 124365, Russian Federation

E-mail: vasilyev_n@ippm.ru, zapletina_m@ippm.ru, ivanova_g@ippm.ru

Received September 25, 2021, peer-reviewed October 11, 2021, accepted October 25, 2021

Abstract: **The paper systematizes the modern scientific and technical experience of VLSI timing analysis. It provides a comparative overview of some existing classical and intelligent methods of static timing analysis, as well as identifies actual tasks and directions for future investigation. The special attention is paid to the intelligent methods for reducing the pessimism of the integrated circuits' static timing analysis results. In particular, an elimination of the false paths and the technological and operational parameters variations are considered. According to recent publications (issued in 2013-2021), various types of neural networks and regression analysis tools are often used to build intelligent methods for estimating the timing characteristics of integrated circuits. In a number of works aimed at reducing the pessimism of classical static timing analysis, the advantage of ensemble methods is argued. It is noted that methods of this type demonstrate more accurate results compared to single methods in conditions of insufficient training data.**

Keywords: **static timing analysis, statistical static timing analysis, intelligent methods, artificial intelligence, machine learning, false paths, Signal Integrity**

Acknowledgements. This research is supported by RSCF (grant No. 17-19-01645).

For citation: Nikolay O. Vasilyev, Mariya A. Zapletina, Galina A. Ivanova. Classical and intelligent approaches for vlsi static timing analysis: an overview. *RENSIT: Radioelectronics. Nanosystems. Information technologies,* 2022, 14(1):27-38. DOI: 10.17725/rensit.2022.14.027.

Contents

1. INTRODUCTION

The analysis of timing parameters of integrated circuits (ICs) is a standard task of ICs design flow. The methods of static timing analysis (STA) are usually used to solve it. They allow analyzing the performance of an integrated circuit under given timing constraints. STA provides information about critical paths in a circuit, i.e. those on which the greatest delay is observed during signal propagation, and at what highest clock signal frequency the functionality of the circuit is not violated.

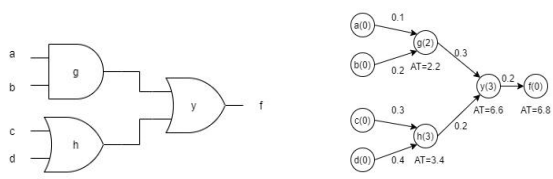
STA methods have several advantages over earlier IC timing analysis methods such as the use of input signal vectors. However, with the reduction in the size of integrated circuit components and the accompanying increase in the complexity of design tasks, the features of classical STA methods become the reason for their efficiency decrease. For example, to obtain a spread of timing characteristics of a circuit by use of traditional STA methods it is necessary to calculate the timing characteristics several times for different parameters variations of the circuit elements. To cover all possible options it is necessary to perform timing analysis 2^n times, where n is the number of parameters. In practice, the assumption is made that if the circuit works correctly under the worst conditions, then under typical conditions its functioning will not be disturbed, so the analysis is usually carried out only for the best and worst cases. Although this assumption is correct, it must be considered that the worst case implies the worst values of all parameters at the same time, which is an unlikely event. Because of this, often the result obtained by means of classical STA methods turns out to be excessively pessimistic.

There are several directions in the development of methods to eliminate the shortcomings of static timing analysis. First, to reduce the pessimism and minimize the complexity of the STA, false paths analysis methods [1] are used. Secondly, there is a direction that uses statistical distributions to describe the spread of the timing parameters of an integrated circuit - statistical static timing analysis (SSTA) [2]. Thirdly, in recent years, STA VLSI methods based on artificial intelligence (AI) approaches have been developed. The purpose of this overview is to systematize the modern scientific experience concerning the above approaches to the evaluation of the timing parameters and characteristics of digital integrated circuits. The paper presents a comparative analysis of intellectual and classical STA methods.

2. STATIC TIMING ANALYSIS. TERMS AND DEFINITIONS

Let us consider the mathematical apparatus, the main terms and definitions used to formulate and solve the problem of VLSI timing analysis. To represent an integrated circuit in STA methods, a timing graph model is commonly used. The timing graph is a weighted directed acyclic graph G = {V, E}, where the set of elements $v_i \in V$ represents the vertices of the graph, and the set of elements $e_{ij} = (v_i, v_j)$, $e \in E$ represents its edges. Each edge $e_{ij} = (v_i, v_j)$ starts from the vertex v_i and ends in the vertex v_j. Each vertex v_i of the timing graph G corresponds to an electrical node of the circuit. Each edge e_{ij} represents a delay between vertices v_i and v_j, and the weight of this edge d_{ij} corresponds to the value of this delay. The number of edges entering the vertex v_i is called an in-degree and is denoted by $d^+(v_i)$, and the set of such edges is called a fanin of v_i (*fanin*(v_i)). The number of edges outgoing from the vertex v_i is called the outdegree and is denoted by $d^-(v_i)$, and the set of such edges is called fanout of the vertex v_i (*fanout*(v_i)). A path p in a graph G is a sequence of edges in the graph where the end of a previous edge is the beginning of a new edge. The delay d_{ij} of the path p_{ij} can be obtained by sum of all the weights of the edges and the delays of the vertices along this path.

The Arrival Time (AT) of v_i is the maximum time required for signal to reach v_i. It is calculated as the sum of the maximum delays on the path from the primary inputs of the circuit to the vertex v_i. *The required time* (RT)



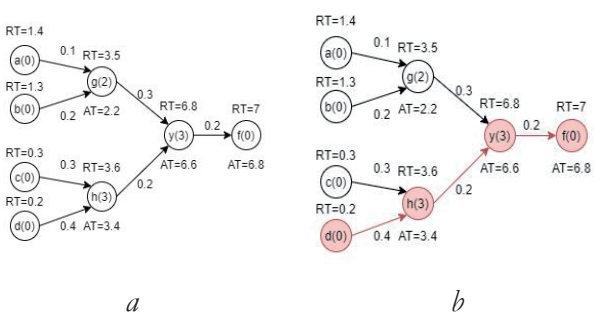
a *b*

Fig. 1. *A logic circuit (a) and the arrival time on the timing graph (b).*

is the time for which the signal must reach the vertex v_i to satisfy the timing constraints specified by the user. *Transition Time* is the time required for an element to switch its logical state when the signal at the previous node of the circuit changes. The *critical path* is the path between the input and output of the circuit, on which the greatest signal delay is observed. *Slack* is the difference between *RT* and *AT*. A positive slack value means that the circuit satisfies the user's timing constraints, while a negative slack value means that the delay on the critical path is too big and the path needs to be optimized.

Consider the example circuit in **Fig. 1***a*. The timing graph for this circuit is shown in Fig. 1*b* (the vertex delay values are shown in parentheses, and the weights of the edges are written near them). The arrival time of the signal is calculated from the inputs of the circuit to the outputs.

The required time *RT* propagates in the opposite direction from the circuit outputs to the inputs (**Fig. 2***a*). The slack is calculated last, due to *AT* and *RT* should be known to



a *b*

Fig. 2. *Required time (a); the critical path and a slack for it (b).*

calculate it. As one can see from Fig. 2*b*, the greatest delay is observed on the path $d \rightarrow h \rightarrow y \rightarrow f$, and the slack at node *f* is $Slack(f) = RT(f) - AT(f) = 7 - 6.8 = 0.2$.

One of the problems of traditional STA methods is that the timing graph doesn't consider the logical structure of the circuit. This shortcoming can potentially lead to false paths appearing among the critical paths. A false path is a path in the timing graph the signal along which will not pass for any input combination of signal values. Due to false critical paths, the result of STA can be unnecessarily pessimistic.

3. STATISTICAL STATIC TIMING ANALYSIS

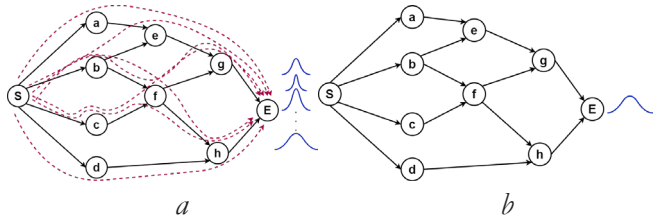
As noted earlier, STA is performed for two extreme cases of variations in the parameters of internal subcircuits (for example, standard cells) of the analyzed IC. The input data for such an assessment is formed by characterizing the IC subcircuits in PVT (Process, Voltage, Temperature) corners corresponding to the best and worst conditions for the functioning and physical implementation of these subcircuits. Although variations in macro parameters, such as temperature or supply voltage, can be approximated by characterizing subcircuits in several PVT corners, variations in local parameters, which include degradation phenomena or fluctuations in technological parameters of elements during production, cannot be fully considered due to their diversity. According to [3], the number of simulation runs required to cover all possible sources of variation in the form of process parameters is in the range of 2^7–2^{10}. For earlier semiconductor fabrication processes, incomplete consideration of technological parameters did not have a large impact on the results of deterministic

STA, however, with a decrease in the size of circuit devices, it becomes the reason for excessively pessimistic results.

These problems can be partly solved by using statistical static timing analysis (SSTA). SSTA uses random values of delays and variations of environment parameters (voltage and temperature) and process parameters (fluctuations of structures and interconnections), which are characterized by probability density function. The result of this analysis is the distribution of the timing characteristics of the circuit, which makes it possible to estimate the actual clock frequency as well as the process yield due to given restrictions on timing characteristics.

SSTA methods can be divided into two groups. Methods of the first group are focused on the analysis of signal propagation paths (Path-based SSTA, PBA) [4]. The disadvantage of these methods is that they must be applied repeatedly to all elements of the set of analyzed paths, which leads to their high computational complexity. The methods of the second group are aimed at analyzing blocks of subcircuits (Block-based SSTA, BBA, also Graph-Based Analysis, GBA). The computational complexity of these methods is usually linear in the number of blocks in the circuit, but the accuracy is lower.

The difference in approaches is shown in **Fig. 3***a,b*. The number of paths increases significantly as the number of vertices in the timing graph increases. Moreover, the greater the connectivity of vertices in the graph, the more paths need to be analyzed in the PBA approach (Fig. 3a). At the same time, in the block approach, the complexity of analysis depends only on the number of vertices of the graph representing the scheme. In addition, PBA methods analyze each path individually, while in the BBA approach, the analysis is



a *b*

Рис. 3. *Path based analysis (a) and block based analysis (b).*

performed in one continuous pass in the direction from the inputs of the circuit to its outputs (Fig. 3*b*).

According to [5], the following idea is often implemented in commercial STA software. The block-oriented approach can be used to preliminarily analyze the circuit and obtain critical paths, which are then further analyzed using the PBA approach.

Two operators [5] are used in SSTA: addition and maximum of distributions. Addition for n independent normal distributions P_1, P_2, ..., P_n with means $\mu_1, \mu_2, ..., \mu_n$ and standard deviations $\sigma_1, \sigma_2, ..., \sigma_n$ is calculated as follows:

$$\sum_{i=1}^{n} P(\mu_i, \sigma_i) = N(\mu, \sigma),$$

$$\mu = \sum_{i=1}^{n} \mu_i, \quad \sigma = \sqrt{\sum_{i=1}^{n} \sigma_i^2},$$

where $N(\mu, \sigma)$ is the normal distribution with mean μ and standard deviation σ.

For correlated distributions, the total standard deviation takes the following form:

$$\sigma = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} \sigma_i \sigma_j},$$

where p_{ij} is the correlation coefficient between delays of elements i and j.

As noted in [5], the application of the maximum operation does not always have an obvious solution. If two normal distributions P_1 and P_2 and with means μ_1 and μ_2 and standard deviations σ_1 and σ_2, respectively, are independent and do not intersect, or two normal distributions have a strong dependence, and their standard deviations are comparable, i.e. $\sigma_1 \approx \sigma_2$, then for $\mu_1 > \mu_2$, $\max(P_1, P_2) = P_1$. However, in the case when two normal distributions

intersect, they are highly correlated, and their standard deviations are difficult to compare, or two normal distributions are independent but intersect, the result of their maximum operation may be a non-Gaussian distribution, that makes statistical timing analysis more difficult. This means that special attention should be paid to the study of the operation of the maximum of random variables in statistical static timing analysis. At the same time, according to the authors of [5], the result of the maximum distributions in SSTA in a form different from the normal distribution is a rare situation and, as a rule, has little effect on the final result of SSTA.

4. INTELLIGENT METHODS OF STATIC TIMING ANALYSIS

One of the important problems of static timing analysis, as noted earlier, is excessive pessimism due to the influence of false critical paths. Several works of foreign and Russian researchers are devoted to the critical paths identification in the timing graph. In particular, the work [6] uses the resolution method, which can be called the simplest version of the intellectual approach to solve this problem.

The resolution method is used to derive new rules from a set of existing ones. It is used in artificial intelligence systems and is the basis of Prolog programming language. Having given timing constraints and an initial set of trivial logical implications of circuit elements, the authors of [6] derive new implications using the resolution method and find logical-timing contradictions in signal propagation paths. Logical implications represent relations between the circuit nodes in the following form. If the node a equals logical zero, then the node b equals one. When a logical-timing contradiction is detected, the algorithm proposed in [6] leaves this path and does not take it into account. The results of the simulation using this method demonstrate that it is possible to reduce the delay estimate by more than 50% for a number of circuits.

The work [7] is also devoted to the false paths elimination problem in STA. It proposes algorithms for false paths identification based on the analysis of the past and current states of the circuit nodes using a program for automatic test sets generation. In [7], three algorithms for false paths are implemented and compared. The result of the work is a decrease in the number of critical paths by ~33%.

The algorithms used in work [7] were also considered in [8]. Paper [8] presents a comparison of the results of timing analysis using STA without false path identification, STA with false path identification, SSTA without false path identification, SSTA with false path identification, and STA and SSTA using neural networks [9]. The neural network architecture used by the authors has 1 hidden layer with 20 neurons, 1 input layer and 1 output layer. The operation of the neural network considered such parameters as the number of inputs and outputs of the analyzed circuit, the number of logic elements, the number of interconnections, power consumption, and the number of gates on a critical path. The results of the work show that the signal arrival time calculated using SSTA is, on average, 7% less than that obtained using STA.

For most of the test circuits, neural network analysis was faster than traditional SSTA, and the accuracy of the results turned out to be higher. However, it should be noted that the report on the results in [8] is sketchy and requires extension and discussion.

The use of neural networks for statistical static timing analysis is also described in the article [10], where the distributions of parameters that affect the result of STA are considered as discrete values. Under these conditions, the authors draw attention to the problem of the operation of a maximum of two discrete distributions. They also pointed out that the addition operation for distributions can lead to an explosive growth in the number of sampling intervals of the distribution function as we move deeper into the timing graph, which results in an exponential increase in computational complexity.

To solve these problems, the authors of [10] propose to replace the standard operations of addition and calculation of the maximum with an approximation of these operations created using deep learning. According to their assumptions, this will allow one to approximate the result with high accuracy and linear computational complexity. The experimental calculations carried out showed an average error of 0.7% for the mathematical expectation and 2.56% for the standard deviation for the proposed method compared to the Monte Carlo method. At the same time, the acceleration was 20.7 times on average compared to standard solutions.

A combined approach based on PBA and machine learning methods was proposed in [11]. Although graph-based timing analysis is significantly faster than path-based analysis, the slack estimation in GBA methods is unnecessarily pessimistic. In this regard, the paper [11] proposes the use of machine learning methods to predict the PBA result based on the GBA result. The basic unit of analysis in the work is the so-called bigram, which combines two consecutive circuit elements (cells) on a timing path.

During the formation of the predictive model [11], a set of electrical and physical features of a bigram unit that can affect PBA-GBA divergence was evaluated. These parameters include:

1) transition time of the first cell in a bigram unit;

2) transition time of the second cell in a bigram unit;

3) arrival time of the first cell in a bigram unit;

4) transition time ratio (TR) of the first cell in a bigram unit;

5) arrival time of the second cell in a bigram unit;

6) drive strength of the first cell in a bigram unit;

7) drive strength of the second cell in a bigram unit;

8) functionality of the first cell in a bigram unit;

9) functionality of the second cell in a bigram unit;

10) fanout of the first cell in a bigram unit;

11) load capacitance of the first cell in a bigram unit;

12) accumulated transition time ratio of the first cell in a bigram unit;

13) propagation delay of the second cell in a bigram unit.

According to the results of [11], the exclusion of at least one of these items can lead to a decrease in the prediction accuracy by 2% to 27%.

The method [11] is based on nonlinear models of classification trees [12] and regression trees. Regression trees were used to calculate signal arrival times for each bigram based on the GBA results, and classification trees helped to determine the divergence between the PBA and GBA results based on the training dataset. In comparison with GBA methods the results obtained during testing

demonstrate the reduced pessimism while maintaining low computational complexity. On average, the decrease in the difference between PBA and GBA in the estimation of signal arrival time was 26.6%.

The problem of correlation of results obtained by the STA software from different vendors is studied in [13]. It is a question of interest due to the fact that the software tools of leading companies (providing the so-called "golden" industry standard STA results) are expensive, therefore to purchase them and to update regularly is not always possible. Thus, there is a need to distinguish both between "gold standard" STA tools and their cheaper (or freely available) counterparts, and between different versions of them.

The authors of [13] proposed a method based on machine learning to predict the results of one STA software tool based on the results obtained by another one. The method was set up based on the following parameters of the analyzed ICs:

1) effective load capacitance;
2) total coupling capacitances;
3) wire ground capacitance;
4) wire resistance;
5) cell input slew;
6) cell output slew;
7) cell delay;
8) wire delay;
9) total stage delay;
10) flip-flop setup time;
11) path slack.

In [13], a set of linear and nonlinear machine learning methods (least squares regression (LSQR), artificial neural networks (ANN), support vector machine regression with a radial basis function kernel) and random forests [14] were used. For each of the models, the authors selected the method that best reduced both the root-mean-square spread and the difference between the maximum and minimum deviations of the results of the two compared software for STA. It is noted that LSQR regression and ANN show less efficiency compared to random forests and support vector regression for the problem of minimizing the range of deviations. The ANN-based model is the most effective at modeling setup times and cell delays. The support vector machine regression performs well when modeling interconnect and cascade delays. The random forest model provided good results for slack calculation. To implement these models, the built-in Matlab vR2013a toolkit for ANN, the LIBSVM library for the support vector machine in Matlab [15], and the implementation of a random forest in Matlab with an open source code [16] were used.

The authors of [13] achieved a decrease in the divergence of up to 6.6 times for the calculated values of the signal setup time and delays of the circuit elements, considering the signal integrity (SI) analysis and the spread of on-chip variation and up to 4 times without SI.

With a decrease in the size of elements in microelectronics, the problem of signal integrity in the STA becomes more and more relevant. Most often, STA is performed for digital signals, however, any digital signal is analog in nature, and it is a subject to similar effects (crosstalk, power supply noise, etc.) that become stronger as the size of the layout elements decreases. Performing STA with regard to SI is a laborious task, and software tools capable of performing this type of analysis are usually available at a higher price. Due to this, in [17], the electrical and logical parameters that lead to the time difference

between classical STA and STA with SI were analyzed. The obtained parameters were used to train a neural network with the following structure: 1 input layer, 2 hidden layers, each of which contains the number of neurons up to twice the number of input parameters, and 1 output layer. In addition, the support vector machine with a radial basis function was used for modeling. To combine the results of the analysis using a neural network and the support vector machine, hybrid surrogate modeling was used, the basics of which are given in [18].

The practical part of work [17] consists of three experiments. The first one was performed to validate the delay and transition time prediction model. The average error for transition time and delay in absolute terms was 0.6% and 1.4%, respectively, compared to the STA method with signal integrity. The worst case error was 8.8% for transition time and 6.9% for delay. In the second experiment, the reliability test of the model was carried out by calculating the timing characteristics for variants of test circuits that were not used in a training dataset. As a result of the experiment, the average prediction error was 2.6%, and in the worst case it was 12.3%. In the third experiment, the accuracy was compared with the previously mentioned model [13]. The model proposed by the authors of [17] shows a much smaller error.

Another important application of STA methods is the evaluation of the timing characteristics of circuits at various stages of the ICs design flow. Thus, in [19], the issue of timing analysis of implementations of user circuits based on the data of the placement procedure is considered. Interconnection's routing makes a significant contribution to the timing characteristics, that's why it is necessary to use a pessimistic estimate at all stages before routing to ensure the correct functioning of the IC. The authors of [19] suggest using machine learning methods to reduce the pessimism of timing analysis characteristics at the placement stage. When forming models and training machine learning algorithms, the following parameters were considered:

1) the output capacitance of the driver (the output capacitance of the previous logic element) and the sink capacitance (the input capacitance of the load).

2) The distance between the driver and the target. According to [19], the vertical and horizontal distance from the driver is proportional to the interconnection delay.

3) Max driver input slew.

4) Location of sink other than the target.

Due to the fact that machine learning requires a fixed number of input parameters, the work uses the average value of the location of the remaining sinks along x and y coordinates, as well as their spread.

The paper [19] compares three machine learning methods for solving the problem: 1) the Lasso linear regression algorithm; 2) artificial neural network; 3) random forest. The training was performed on the STA data applied to already routed circuits. The problem of false paths in this approach is considered indirectly using the STA algorithm used to prepare the training data. Testing was performed on the same test circuits as training, but resynthesized with changed parameters to obtain a different implementation of the circuits. As a result, the random forest method showed the smallest standard error in both the delay estimate and the transition time estimate. In the work, a comparison was made with a commercial STA software tool, according to which the proposed method based on

machine learning showed the best result. It was confirmed not only by comparing the calculated delay, which turned out to be closer to the real values, but also by a smaller spread of the timing parameters of the circuit compared to the results of commercial software for STA.

A study of the dependence of the timing characteristics of the circuit on the supply voltage is presented in [20]. Its main thesis is that the difference in supply voltage has the greatest impact on the difference in delays when working under different PVT conditions. As we noted earlier, the characterization of elements is usually performed for several PVT corners, and it does not cover all possible variations of both technological and macro parameters. The authors of [20] suggest using machine learning to perform STA with different power supply voltages.

The method [20] is based on a convolutional neural network [21] with the assumption that the task of determining the path delay is similar to the task of computer vision. Thus, images in computer vision tasks are represented as a three-dimensional tensor (W, H, C), where W is the image width, H is the image height, and C are the color channels of the image. By analogy with this, the characterization corners can be represented as a three-dimensional tensor (P, V, T), where P are the variations in the technological parameters of electronic components, V are the variations in the supply voltage, and T are the variations in temperature. For modeling, the authors of the work used the linear regression method, the Lightgbm method [22], and an ensemble of these two methods. Two experiments were performed in [20]. The first one was dedicated to the path delay prediction for one value of the supply voltage based on STA for another value of the supply voltage at the same value of parameter P. The ensemble of methods showed the best result among the others with a maximum error of less than 4.9%. In the second experiment, the path delay was predicted for a given value of the supply voltage with given variations of the process component P based on STA with other parameters of the supply voltage and variations in the technological parameters of electronic components. The maximum error in this case was 7.9%.

5. DISCUSSION OF STA INTELLIGENT METHODS

Among the presented works on the use of artificial intelligence for solving STA problems, various types of neural networks are mostly often proposed. This conclusion is also supported by works that are not considered in detail within this overview, given in **Table. 1**. At the same time, ANNs are not a universal way to solve any STA problems. There are several works [11,14,16,17] devoted to the study and comparison of the results of several machine learning methods for particular STA problems.

Table 1
Application of intelligent approaches to the problems of static timing analysis of integrated circuits

Object	STA task	Solution method	Ref.
Noise analysis	STA considering power supply noises	Neural network, support vector machine, LSBoost	[23]
Correlation of results of the STA	STA considering variations in path delays when modeling at different corners of characterization	Neural network	[24]
	Correlation of STA results under different modeling conditions	Multiple Linear Regression	[25]
Transistor Degradationв	Timing Prediction with Transistor Degradation	Feedforward Neural Network	[26]
STA runtime reduction	STA with parallel data processing	Neural Network	[27]

As it turned out, in most of these works, the best result is provided by an ensemble of methods (for example, a combination of the linear regression method and the Lightgbm method in [20], a neural network and the support vector machine in [17]). Also, good results are demonstrated by methods based on the "Random Forest", which is an ensemble of decision trees. Presumably, ensemble methods show more accurate results compared to single methods due to their better behavior in conditions of insufficient training data. In addition, due to the variety of mathematical dependencies contained in their composition, ensemble methods are able to take into account the nonlinear nature of the problems of timing analysis.

The most cautious thesis, deduced on the basis of modern scientific and technical experience, can be formulated as follows. Each task of static timing analysis requires careful selection of an appropriate intelligent method or a combination of them, as well as careful study of the way the training and test data sets are formed. Currently, a unified methodology for solving STA problems based on AI methods has not been formed.

6. CONCLUSION

The article considers the application of classical and intelligent approaches to the problem of timing analysis of integrated circuits. One of the frequent problems solved in modern works is to reduce the pessimism of the results, which is inherent in the methods of static timing analysis. A small part of the work is devoted to the acceleration of STA without significant loss of accuracy.

Great interest in the application of intelligent methods to the timing analysis of IC is observed on the part of foreign scientists, while there are very few Russian-language studies on this topic. Presumably, the reason for this lies in the fact that for a long time domestic enterprises and microelectronics development centers had used exclusively foreign proprietary software and there was no urgent need for own developments.

An analysis of the existing scientific and technical literature helped to identify promising areas for future research. These include the elimination of the influence of false paths, the reduction of the pessimism, a detailed study of the operations of addition and the search for the maximum of the distributions of variations in the technological and macro parameters of the IC, the consideration of the noise factor and the signals integrity observation, establishing correlations between the results of timing analysis at different stages of IC design, as well as between different, including proprietary, software for timing analysis, or in intermediate PVT corners.

REFERENCES

1. Wang SJ, Tzeng TH, Li KSM. Fast and accurate statistical static timing analysis. *Proceedings of IEEE International Symposium on Circuits and Systems* (ISCAS), 2014: 2555-2558.
2. Blaauw D, Chopra K, Sristava A, Scheffler L. Statistical timing analysis: From basic principles to state of the art. *IEEE transactions on computer-aided design of integrated circuits and systems,* 2008, 27(4):589-607.
3. Forzan C, Pandini D. Statistical static timing analysis: A survey. *Integration*, 2009, 42(3):409-435.
4. Shibuya T, Nitta I, Homma K. Statistical static timing analysis technology. *Fujitsu Sci. Tech. J*, 2007, 43(4):516-523.
5. Amin CS, Menezes N, Killpac K, Dartu F, Choudhury U, Hakim N, Ismail YI. Statistical static timing analysis: How simple can we get? *Proceedings of the 42nd annual Design Automation Conference*, 2005:652-657.

6. Solovyev RA, Glebov AL, Gavrilov SV. Staticheskij vremennoj analiz s obnaruzheniem lozhnyh provodjashhih putej na osnove logicheskih implikacij [Static timing analysis with false path detection based on logical implications]. Problemy razrabotki perspektivnyh mikrojelektronnyh sistem (MES) [Problems of Perspective Micro- and Nanoelectronic Systems Development (MES)], 2006, 1:22-28. (In Russ.).
7. Parnerkar SV. Timing False Path Identification using ATPG Techniques. *Master's thesis*. 2010: 20 p.
8. Ramesh SR., Jayaparvathy R. Artificial neural network model for arrival time computation in gate level circuits. *Automatika*, 2019, 60(3):360-367.
9. Abiodun OI, Jantan A, Dada KV, Omolara AE, Mohamed NA, Arshad H. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 2018, 4(11):41 p.
10. Savari MA, Jahanirad H. NN-SSTA: A deep neural network approach for statistical static timing analysis. **Expert Systems with Applications**, 2020, 149:20 p.
11. Kahng AB, Mallappa U, Saul L. Using machine learning to predict path-based slack from graph-based timing analysis. *Proceedings of IEEE 36th International Conference on Computer Design* (*ICCD*), 2018:603-612.
12. Charbuty B, Abdulazeez A. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2021, 2(01):20-28.
13. Han SS. Kahng, AB, Nath S, Vydyanathan AS. A deep learning methodology to proliferate golden signoff timing. *Proceedings of Design, Automation & Test in Europe Conference & Exhibition* (*DATE*), 2014:1-6.
14. Probst P, Wright MN, Boulesteix AL. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2019, 9(3):19 p.
15. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology* (*TIST*), 2011, 2(3):1-27.
16. Random Forest. Available at: https://code.google.com/archive/p/randomforest-matlab/(accessed 12.11.2021).
17. Kahng AB, Luo M, Nath S. SI for free: machine learning of interconnect coupling delay and transition effects. *Proceedings of ACM/IEEE International Workshop on System Level Interconnect Prediction* (*SLIP*), 2015:1-8.
18. Kahng AB, Lin B, Nath S. Enhanced metamodeling techniques for high-dimensional IC design estimation problems. *Proceedings of Design, Automation & Test in Europe Conference & Exhibition* (*DATE*), 2013:1861-1866.
19. Barboza EC, Shukla N, Chen Y, Hu J. Machine learning-based pre-routing timing prediction with reduced pessimism. *Proceedings of the 56th ACM/IEEE Design Automation Conference* (*DAC*), 2019:1-6.
20. Bao W, Cao P, Cai H, Bu A. A learning-based timing prediction framework for wide supply voltage design. *Proceedings of the 2020 on Great Lakes Symposium on VLSI*, 2020:309-314.
21. Dhillon A, Verma GK. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence,* 2020, 9(2):85-112.
22. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Liu TY. Lightgbm: A highly efficient gradient boosting decision tree. *Advances*

in neural information processing systems, 2017, 30:3146-3154.

23. Liu YC, Han CY, Lin SY, Li JCM. PSN-aware circuit test timing prediction using machine learning. *IET Computers & Digital Techniques,* 2017, 11(2):60-67.
24. Guo J, Cao P, Sun Z, Xu B, Liu Z, Yang J. Novel prediction framework for path delay variation based on learning method. *Electronics,* 2020, 9(1):11 p.
25. Kahng AB, Mallappa U, Saul L, Tong S. "Unobserved Corner" Prediction: Reducing Timing Analysis Effort for Faster Design Convergence in Advanced-Node Design. *Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition* (*DATE*), 2019:168-173.
26. Ebrahimipour SM, Ghavami B, Mousavi H, Raji M, Fang Z, Shannon L. Aadam: A fast, accurate, and versatile aging-aware cell library delay model using feed-forward neural network. *Proceedings of the IEEE/ACM International Conference On Computer Aided Design* (*ICCAD*), 2020:1-9.
27. Abrishami MS, Ge H, Calderon JF, Pedram M, Nazarian S. NN-PARS: A Parallelized Neural Network Based Circuit Simulation Framework. *Proceedings of the International Symposium on Quality Electronic Design* (*ISQED*), 2020:452-456.