Valeriy ROGOZA
WEST POMERANIAN UNIVERSITY OF TECHNOLOGY,
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Using intelligent programming paradigm in CAD systems

Prof. dr hab. Eng. Valeriy ROGOZA

Recieved the M.Sc. degree in the Radioelectronics Department of the National Technical University of Ukraine "Kiev Polytechnic Institute" in 1971. He received PhD (1975) and DrSc degrees (1996) in the mathematical methods in scientific research, and professor's certificate in 1999. His current research interests are focused on the numerical methods of the applied mathematics, methods of building intelligent adaptive information systems and simulation of complicated objects and processes.

e-mail: wrogoza@wi.ps..pl

Abstract

The intelligent programming paradigm is considered as a concept that combines two basic properties of a sophisticated software, namely: adaptive tuning and evolutionary self-organization. Such properties can be realized at the algorithmic level using object-oriented programming languages.

Keywords: adaptive simulation, self-organization, CAD system, VLSI design.

Wykorzystanie paradygmatu programowania inteligentnego w systemach projektowania wspomaganego komputerowo

Streszczenie

Paradygmat programowania inteligentnego jest rozpatrywany jako koncepcja, która łączy w sobie dwie zasadnicze własności skomplikowanego oprogramowania, mianowicie: adaptacyjne dostrajanie modeli i ich samoorganizacja ewolucyjna. W artykule pokazano, że omówione właściwości mogą być realizowane z wykorzystaniem specjalnych algorytmów syntezy modeli składników obiektów ulegających symulacji oraz paradygmatu programowania obiektowego.

Słowa kluczowe: symulacja adaptacyjna, samoorganizacja, systemy projektowania wspomaganego komputerowo, projektowanie układów scalonych o dużym stopniu scalenia.

1. Introduction

Simulation and computer-aided design (CAD) of objects with sophisticated behavior, such as very large scale integration circuits (VLSI), robotic systems, economy systems and so on require the use of programming paradigm, which reflects the complexity of such systems and allows us to simulate adequately sophisticated relations between the parts of the mentioned systems. Any complex object (system) differs fundamentally from "ordinary" objects (systems) by a number of specific features, and the most remarkable among them are [1]: uniqueness of behavior (each complex system is distinct from the others possessing unique properties), unpredictability of future states, information no-entropy (that is, the capability of the system for reducing the unpredictability of behavior during system operation), and structural heterogeneity. As to Turing's statement [2], any model of complex object that reflects adequately the behavior of such an object, is as sophisticated as the original object. Thus, dealing with the problem of complex objects design and analysis, we come up against the problem of choosing a proper programming paradigm possessing expressive tools for simulation.

A possible approach to the development of the intelligent programming paradigm is discussed in the paper, and the example of the VLSI CAD system, in which the above paradigm is realized, is presented.

2. The complexity of VLSI

The challenge of high-accurate simulation of complex objects is widely discussed in literature. We would like to illustrate the sources of complexity phenomenon by the example of a VLSI. The requirement of high-precision simulation of VLSI, especially if we deal with precision integrated operational amplifiers, analogue filters, embedded electronic systems and so on, is customary. This brings up the problem of synthesis of high-accurate models of VLSI components and the problem of effective use of such models in computer-aided design. The way to do this is the synthesis of hybrid models combining physical and circuit models in the simulation process. The necessity of considering physical effects in combination with the circuit design stage stems from the fact that a number of effects cannot be taken into account if circuit models are solely used. Much evidence points to this conclusion. For better understanding of the essence of the mentioned challenge, let us consider a simplified structure of the integrated *n*-channel MOS transistor presented in Fig. 1.

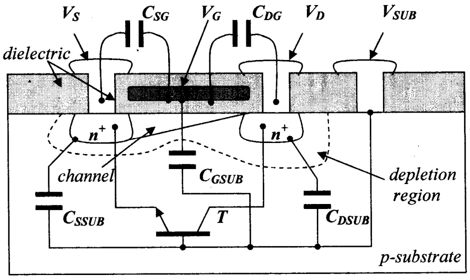

Fig. 1. A simplified structure of a MOS-transistor with spurious components
Rys. 1. Uproszczona struktura tranzystora MOS z elementami pasożytniczymi

As can be seen, the structure of the MOS-transistor consists of a number of passive components (Fig. 1 shows five capacitances) and active components (for simplicity, a single n^+pn parasitic transistor is presented in the Figure, but actually there are a number of parasitic transistors accompanying such a MOS structure). The fragment given in Fig.1 exhibits five inherent capacitances: C_{SG} - between the source lead and the gate, C_{DG} – between the drain lead and the gate, C_{SSUB} – between the n^+-source and the *p*-substrate, C_{DSUB} – between the n^+-drain and the *p*-substrate, and C_{GSUB} – between the gate and the *p*-substrate, and the latter in turn is comprised of two capacitances - the capacitance between the gate and the channel C_{GC} and the capacitance between the channel and the substrate C_{CS} (are not shown in the Figure).

Capacitances C_{SG} and C_{DG} are constant, whereas C_{GSUB}, C_{DSUB} and C_{GSUB} vary in magnitude with variation of terminal voltages (that is, physical states of the structure). In addition, there is a parasitic bipolar n^+pn transistor *T* with the n^+-source region as the emitter, *p*-substrate as the base, and n^+-drain as the collector. It comes into particular prominence for a short-channel MOS-transistor, because a short channel plays a role of a thin base of the n^+pn parasitic transistor providing a reasonable value of parameter β that may impact significantly on the behavior of the general MOS transistor.

Some of the circuit components shown in Fig.1 are dependent not only on lead voltages, but also on inner physical parameters and variables of the structure, such as the intensity of transverse and lengthwise electric fields across the channel, carrier mobility and density, two- and three-dimensional effects including parasitic effects, doping concentration and so on. For example, they are capacitance C_{GSUB} and the $n^{+}pn$-transistor. There are a lot of publications on this topic, where efforts have been mounted to overcome the problem of simulation parasitic and second-order effects and to hit on plausible solutions remaining in the context of circuit models. But unfortunately all the proposed methods are oriented to the realization of more and more sophisticated models of components rather than on the ways in which such models can be adapted to the varying physical states of semiconductor structures.

As to the VLSI CAD systems, the main idea, which provides the foundation of adaptive systems, is to extend the set of variables that describe physical states of the integrated circuit, by the way of combination of several stages of design within an integrated through design simulation.

The experimental VLSI CAD system LINE dedicated for the design of analogue integrated circuits, in which the physical and circuit simulation phases were combined in the form of the through design CAD system, has been created some years ago at the National Technical University of Ukraine, and later research has been continued at the West Pomeranian University of Technology in Szczecin, Poland. It is appropriate to recall here that, on the above reasons, physical as well as circuit models synthesized in this CAD system are not fundamentally conservative, they are steadily tuned and varied according to the varying physical states of transistors. In other words, the synthesized models have no fixed structures and parameters, and they are modified as new information on physical states of IC transistors becomes available. In this connection, two issues are of a fundamental nature: what methods have been used as the basis for model synthesis in the above through design system and in what manner controls might be exerted in such a system to provide correlated actions of all the modules. Next we would like to concentrate on the idea of the adaptive simulation as a possible approach to the development of the simulation system with the desired properties.

3. Adaptive models synthesis

Adaptation, in general, is the feature of a system to tune its architecture and component functions with the aim of finding optimal mode of operation. In particular, adaptive simulation in CAD systems means the way of model formation using step-by-step approximation of models that allows us to maximize the accuracy of simulation. The topic of adaptive model building has been studied as a general scientific problem applicable in various areas of human activity, such as economy, automated control, artificial intelligent systems and so on. In spite of different applications, the proposed approaches have many common features. We would like to restrict our consideration to the two methods only, they are: the method of block building complicated models using adaptors [3] and polynomial model synthesis using the method of grouping arguments [4].

The first approach is based on the idea of the CAD system to accommodate itself to the specific needs of the problem to be solved by way of automatic model tuning realized by the use of specialized software components called the adaptors. The latter are capable of making decision on the basis of prior information, which is brought into the system by the user (or/and expert), as well as posterior information, which is formed in the simulating CAD system with its operation. And the second approach has been realized in several forms, such as evolutionary programming, stochastic identification, and evolutionary self-organization. In the LINE subsystem, the third approach has been implemented.

For better understanding of key concepts laying at the basis of structural adaptive synthesis of models in CAD systems and the role of adaptors, let us consider the mentioned LINE system, which consists of two fundamental subsystems that perform computation at the two stages of VLSI design: physical design and circuit design (as shown in Fig. 2).

At the stage of physical design, the well-known basic set of physical equations are to be solved:
1. The equation for electron (n) and hole (p) current densities

$$J_{n,p} = qn\mu_{n,p}E \pm qD_{n,p}\nabla_r n,p \; ; \qquad (1a)$$

2. The equation for electron and hole continuities

$$divJ_{n,p} \mp q\partial n,p/\partial t = \pm qR \; ; \qquad (1b)$$

3. Poisson's equation for potentials

$$\nabla^2\varphi = -(1/\varepsilon_S)(p-n+N) \; , \qquad (1c)$$

where q = 1.6 $\cdot 10^{-19}$C; $n(p)$ is the density of electrons (holes); μ_n (μ_p) is the effective mobility of electrons (holes); D_n (D_p) is the diffusion coefficient for electrons (holes); E is the electric field intensity; R is the carrier recombination ratio; N is the total doping density; ε_S is the dielectric constant of the semiconductor; φ is the electric potential; and ∇ is the Laplace differential operator.

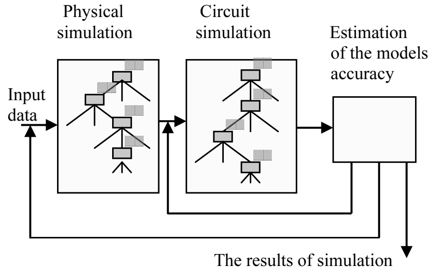

Fig. 2. The two-subsystems platform LINE used in the VLSI CAD design
Rys. 2. Platforma LINE o dwu podsystemach zastosowana w projekcie VLSI CAD

To solve the given set of equations, one should substitute predetermined parameter values such as $n(p)$, μ_n (μ_p), D_n (D_p), R , and N into the equations. But these values are not constant and vary with variations of physical states of the semiconductor structure. This reasoning alludes to the conclusion that iterative simulation processes including simultaneous physical and circuit analysis is a way of overcoming the problem. Moreover, each parameter can be calculated using hierarchical multistage procedures, each having a number of alternatives.

For example, one of the possible hierarchical schemes for obtaining electron (n) and hole (p) densities in the semiconductor substrate is given in Fig.3. The following notation is adopted: $(x_1, x_2, x_3, \ldots) \rightarrow y$ is the mapping of variables (and/or parameters) x_1, x_2 , x_3, … into variable (or parameter) y. Such a mapping reflects the fact that there is a proper relationship in the semiconductor physics, which allows us to obtain the y value using values x_1, x_2 , x_3, …; if several alternative formulae exist for desired parameters, which vary in accuracy, then additional criteria are required to be used to choose one of them. Alternative mappings are placed in the same boxes in the diagram (Fig. 3). Say, Stage 1 includes four alternative versions of models for certain parameters, and these parameters serve as input parameters for the next stage, namely, Stage 2, which, in turn, includes four alternative mappings for the parameters being input parameters for the third stage, and so on.

Omitting physical sense of the parameters given in Fig. 3, we would like to emphasize that similar hierarchical schemes are typical for any model of the physical level of simulation as well as of the circuit level. Indeed, for example, a classical Ebers-Moll model of bipolar transistor has at least three levels of accuracy, which are divided into the three levels – EM1, EM2, and EM3 models, each can be used successfully at the multistage circuit simulation scheme. Multilevel model hierarchy is also observed for the numerous versions of the family of Sah's models of MOS transistors as well as for other devices.

Once circuit analysis has been completed, the LINE system compares voltages and currents obtained at the circuit analysis phase and those which has been used at the previous physical phase of simulation as input variables and parameters. If they are agreed (at the given value of discrepancy), the simulation process is terminated, otherwise the physical simulation phase is repeated once more with the refined voltages and currents (causing modification to the magnitudes of potentials, electric fields, and other variables).

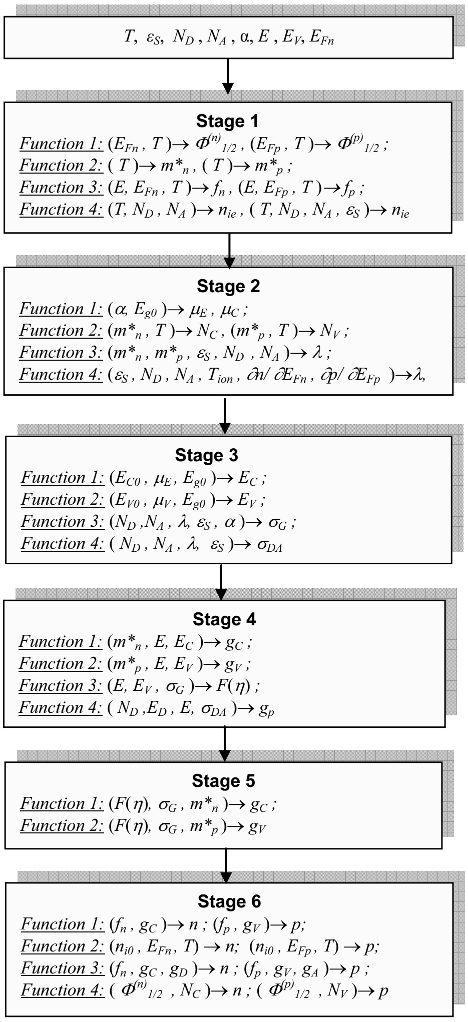

Fig. 3. The scheme of computation of *n* and *p*
Rys. 3. Schemat obliczeń parametrów *n* i *p*

The generalized structure of the problem adaptive system is shown in Fig. 4.

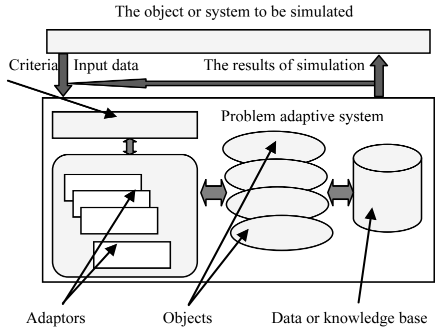

Fig. 4. Generalized structure of the problem-adaptive system
Rys. 4. Uogólniona struktura systemu problemowo-adaptacyjnego

The specific feature of such a system is that there are a number of special-purpose program units called adaptors, which manage the processes of model synthesis on the basis of a priori formulated criteria or criteria generated directly during system operation. In the latter case, adaptors are possible to set up their functionality by way of self-learning with the use of knowledge accumulated in the adaptor memory (programming tools for the realization of adaptors have been discussed in [3]).

Thus the mentioned properties of the above approach with the use of adaptors give us the basis to affirm that we deal with the paradigm based on knowledge, that is, the paradigm of intelligent programming.

The objects presented in Fig.4 denote program components, and adaptors associated with the objects may serve one or several objects simultaneously.

As was mentioned above, evolutionary self-organizing polynomial model method is another approach to the synthesis of adaptive models. The method presented in the paper deals with the Gabor-Kolmogorov polynomials and it is called the method of grouping arguments. We briefly outline the mathematical foundations of the method, additional information the reader can find in [4].

Assume that we have an object in the form of a "black box" with the set of input time-dependent signals (variables) $X = \{x(t)_1, x(t)_2, \ldots, x(t)_N\}$ and the set of output time-dependent signals (variables) $Y = \{y_1(t), y_2(t), \ldots, y_M(t)\}$ (Fig. 5). Let there be K observations of the sets of input signals $X(i)$ and output signals $Y(i)$, $i = 1, 2, \ldots, K$. The problem is to determine functional $Y = \mathrm{F}(X)$ using the mentioned observations.

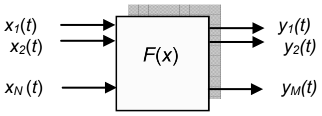

Fig. 5. The object under consideration is given in the form of a "black box"
Rys. 5. Analizowany obiekt w formie czarnej skrzynki

The general math model $F(x)$ can be obtained using the Gabor-Kolmogorov polynomial (GKP) [4]. For example, if we have the set of input variables $X = \{x_1, \ldots, x_N\}$, then the third order GKP has the form

$$Y = a x_0 + \sum_{i=1}^{N} a_i x_i + \sum_{i=1}^{N} \sum_{j \le i}^{N} a_{ij} x_i x_j + \sum_{i=1}^{N} \sum_{j \le i}^{N} \sum_{k \le j}^{N} a_{ijk} x_i x_j x_k \qquad (2)$$

where a_i are unknown values.

When generating model (that is, when we determine coefficients a_i), the criterion of minimal error between the accurate output y_i and the output obtained from (2), is used:

$$\bar{\varepsilon}^2 = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2, \quad \bar{\varepsilon}^2 \to 0 \qquad (3)$$

Three principles are in the basis of the solution, namely:

1. For the given set of input variables there are a lot of GKPs providing $\bar{\varepsilon}^2 \to 0$. Moreover, the plots of functions $\bar{\varepsilon}^2 = \varphi(C)$, where C is the complexity of the GKP, that is, the power of the Gabor-Kolmogorov polynomial, have well-defined minima. It means that there exists a value of C, for which the error is minimal.
2. In an arbitrary formal logical system, there is a set of statements, which cannot be proved or refute within the framework of axioms of the given system. It is a well-known Gölder theorem [5]. In our case, it means that any set of input signals will not be complete. This leads us to the necessity of the use of the external supplement as a specimen providing

model training. (The similar idea is successfully used in the development of artificial neuron networks.) Thus the sets of input data may be divided into two groups, one of them is the set used for "training" (that is, for building GKP) and the other for testing (that is, for the estimation of error): $N_{in} = N_{tr} \cup N_{test}$.

3. The selection of optimal solutions implies the use of the freedom concept, that is, with selection of any decision we should save some degree of freedom in order to correct our decision in future (if necessary). This concept is known as the Gabor freedom principle [4].

The scheme of operation of the method of grouping arguments is given in Fig. 6.

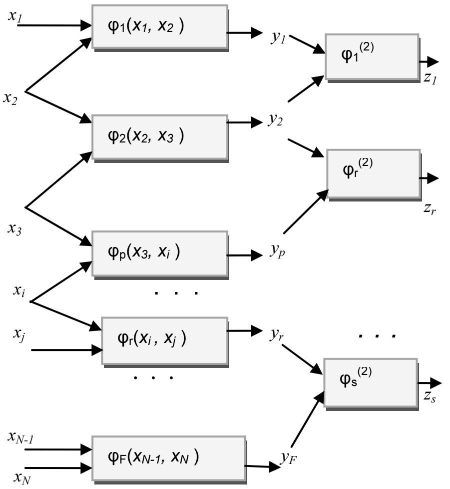

Fig. 6. The scheme of model synthesis by the method of grouping arguments
Rys. 6. Schemat syntezowania modelu metodą grupowania argumentów

The method operates in the following manner. Selecting the first set of input data, we combine all the pairs of input data and form GKPs (2), in such a way we form functions of the kind $\varphi_r(x_i, x_j)$, $\forall i, j$. Then using testing sets we compute the errors by (3) for each function $\varphi_r(x_i, x_j)$. If there exists a function with the admissible value of error, the process is terminated, otherwise we select a number of the best functions $\varphi_r(x_i, x_j)$, $\forall i, j$, which provide minimum error, and form more complicated combinations of input data combining the selected functions $\varphi_r(x_i, x_j)$ by pairs, and next we analyze errors comparing the formed functions of the kind $\varphi_s^{(2)}$ with samples, and so on.

It should be noted that the presented method is based on the selective use of the best samples of GK polynomials and in a certain sense it is close to the evolution self-organization concept. There are a wide variety of modifications of the presented method. For example, we can complicate the formed functions adding new variables and forming GKPs of higher powers, as well as we can combine this method with the fuzzy set techniques [4]. In the LINE system, we have realized only the simplest version of the method of grouping arguments discussed above. It was used for the synthesis "formal" models of integrated components (that is, models having no physical rather mathematical sense). More appropriate research of the method of grouping arguments is to be done in future.

4. Conclusive discussion

The above approaches to the synthesis of models of VLSI components has been realized in the mentioned LINE system dedicated to the design of precision analogue integrated circuits. Two design modules have been included into the system: physical and circuit design.

In distinction to other similar CAD systems, the LINE system is based on the idea of dynamic tuning models of integrated components realized by using the two methods of model synthesis described above. Our discussion shows, in particular, that the results of model synthesis realized in the LINE system are not predictable, but the system is able to reduce information entropy with operation. Thus the system possesses properties of complex systems as was determined in Introduction. A peculiarity of the system is that it includes both central control subsystem which plays a role of a job manager, and a number of local control subsystems called the adaptors. The functionality of adaptors can be modified during system operation, as well as models synthesized can be modified according to the varying physical states of semiconductor structure. This allows us to consider such an approach as a realization of paradigm of intelligent programming.

Programming tools of building adaptors are based on the object-oriented programming paradigm [3]. In particular, encapsulation makes it possible to create classes of special-purpose adaptors combining the sets of data and methods in the common classes, inheritance allows us to build a structural hierarchy of models, and polymorphism provides capability for using a unique interface for different implementations of adaptors, what is especially important in building processes of selection of alternative solutions. More details are analyzed in the cited article.

The LINE system was successfully applied for the design of precision analogue integrated circuit with improved parameters, such as KT140UD25, KT140UD26, KT140UD27, KT140UD17 and some others (Fig. 7) used in civil and military engineering systems.

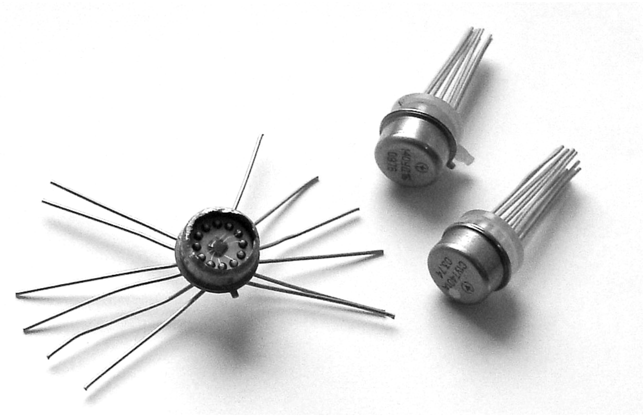

Fig. 7. Some samples of VLSI designed with the use of the LINE system
Rys. 7. Przykłady układów VLSI zaprojektowanych przy użyciu systemu LINE

5. References

[1] Casti J.: Connectivity, Complexity, and Catastrophes in Large-Scale Systems. N.Y., John Wiley & Sons, 1979.
[2] Turing A.: Computing machinery and intelligence. Mind, 59, 1950, p. 433- 460.
[3] Rogoza V.S.: Adaptive simulation of VLSI. Proc. of the 11th Intern. Conf. "Mixed Design of Integrated Circuits and Systems", MIXDES – 2004, Szczecin, Poland, 24 – 26 June 2004, p. 326 – 331.
[4] Zajchenko Ju. P.: Fuzzy models and methods in intelligent systems. Kiev: SLOVO, 2008, 344 p. (in Russian).
[5] Gölder K. Über formal unentscheidbare Sätze der Principia mathematica und verwandter Systeme I.", Monatshefte für Matematik und Physik, 38, 1931, p. 173 – 198.

Artykuł recenzowany