micro- and nanoelectronic systems – 2010. Proceedings of the IV All-Russian Scientific and Technical Conference / under the total. Ed. Academician of the Russian Academy of Sciences A. L. Stempkovsky. – Moscow: IPPM RAS, 2010. – P. 283-288.

[5] Korshunov, F. P. Radiation effects in the technology of semiconductor materials and devices / F. P. Korshunov, Yu. V. Bogatyrev, S. B. Lastovskii, I. G. Marchenko, N. E. Zhdanovich. // Actual problems of solid-state science. – 2003. – 245 p.

PARALLEL COMPUTING ENVIRONMENT FOR DIGITAL DEVICES SIMULATION AND VLSI TOPOLOGY VERIFICATION

A. Belous[1], V. Solodukha[1], S. Shvedov[1], A. Borovik[2], A. Kostrov[2], V. Stempitsky[2]
[1]Joint-Stock Company "INTEGRAL", Minsk, Belarus
[2]Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

INTRODUCTION

During several decades semiconductor (digital) electronics is developed exponentially according to Moore's Law. By now the number of transistors on chip for wide application ICs has exceeded 2.5 billion, and has reached 8 billion in the segment of graphics processors. On the one hand this allows to place a projected device including interface components on one chip entirely, which significantly increases the manufacturability and reliability of the device. On the other hand the complexity of the designed devices increases greatly. In this connection several problems appear, among which the following ones can be noted increasing time of design, verification and reliability control of digital ICs and the equipment based on them. Design of the digital electronic devices is a cyclical process. Design cycle includes a series of consecutive steps with different computational complexity, such as description compilation and synthesis, circuit simulation, tracing into the crystal, simulation at the crystal level. These steps are usually performed automatically under the control of a project developer with the use of computational tools and specialized programs included in CAD. Implementation of a series of steps using CAD of widely known companies, such as Actel, Xilinx and Mentor Graphics, even for small devices with hundreds of thousands of equivalent gates can take up to several hours of computer time even on a hi-end class multi-core personal computer (PC).

With an increasing project size the number of development-testing cycles increases, which leads to more than linear growth of development time as a whole. As a result, design and debugging of digital devices can take from several months to one year despite the high performance of desktop computers, while in today's market conditions success of a product can completely depend on how a manufacturer (supplier) outstrips the competitor.

The stage of project testing (verification) is the most critical and time consuming stage in the whole process and it can take up to 90% of the total development time. The number of verification tests for a large project can reach thousands and all of them are performed consecutively if typical CAD software that runs on a standard PC is used for development.

II. PARALLELIZATION OF TESTS FOR LIBERO CAD AND XILINX CAD

Both in Actel Libero CAD and in Xilinx ISE CAD Mentor Graphics ModelSim program can be used as a software tool for devices simulation. The main operating mode of ModelSim is graphical. Two methods are used for ModelSim configuration: modelsim.ini and do file. Both of these files are generated in both CAD dynamically for each simulation run. Use of the files as a data exchange mechanism and for parameters transmission allows to carry out the simulation not only in the graphical, but also in the batch mode, which allows to perform the device simulation parallel using several computational nodes by placing a different part of a task on each node.

Fig. 1 shows the scheme of user interaction with ModelSim simulator in the case of classical, as provided for by a typical design technical process, and parallel implementation.

In the case of classical implementation the user carries out the instruction to start the simulation with CAD. The CAD prepares data based on project content for simulation. Data for simulation are the file that contains a device model and the file that contains simulation parameters. Also in addition to these two files, additional files with components that can be used from the model file, as well as files with a description of time delay of a device can be transmitted to the simulator. The prepared data in do file form are transmitted

to ModelSim program. The result of the simulation is a timing diagram. The timing diagram is shown in the main ModelSim window.

In the case of parallel computing for device model simulation the process is different. Instead of ModelSim CAD transmits the data to a grid or a parallel application that performs the following functions:

Selection of the files required to run the simulation based on the analysis of the do file and the command line.

III. GENERATION OF DO FILES SET FOR THE COMPUTATIONAL NODES

Preparation and transfer of all the files necessary to run the simulation on the cluster nodes.
Starting of remote simulation. To start the remote simulation it is necessary that Mentor Graphics ModelSim software has been installed on each node of the cluster.
Transmission of the simulation results to the local machine and display of them in the ModelSim software. Thus ModelSim, which is locally installed on the user's machine, will be used only to display the results.

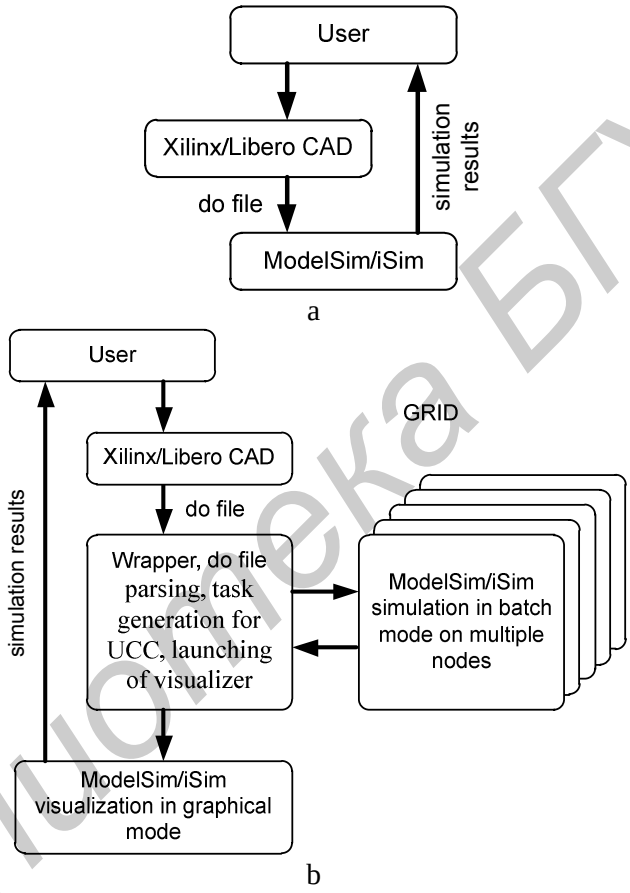

Figure 1 – Scheme of user interaction with the system in the case of classical (a) and parallel (b) implementation

Figure 1 shows that the user doesn't interact directly with the grid application or the cluster. Work in the parallel computing environment is carried out completely transparent for the user. All details of his interaction with the infrastructure are hidden under natural CAD interface. The wrapper of ModelSim intercepts the data from the CAD generated on the basis of the user actions, performs their parsing, generates a set of tasks for the nodes, ensures their performance in the parallel computing environment, acquisition of results and their return to the user.

IV. PARALLELIZATION OF LAYOUT VERIFICATION BY DRC-RULES

Mentor Graphics Calibre software which uses SVRF (Standard Verification Rule Format) file for verification rule storage can be used for verification of VLSI layout on the basis of design rule checking (DRC) system.

The rules control the following actions: initial definition of layers, generation of derivative layers, design rule checking, extraction of connections, electrical rule checking, recognition of a device from an electric circuit, comparison of circuits and others. For convenience the entities in the file can be divided into

two classes: specification statements and operations. The main difference between these two classes is that the operations perform over the data of layers and the specification statements define the environment in which these operations are performed.

The main place where parallelism is concentrated is the set of operators for rules checking. Each operator is an independent unit of action, which doesn't depend on other operators. The result of work of rules checking operators is collected to the output database. All operators of rules verification work in the context of other operators therefore the source file can be conditionally divided into two parts – context and the set of rules verification operators. This makes it possible to generate SVRF file that will contain the original context and one rule. Such files can be created as much as there are rules in the source file. Each such SVRF file can be sent to its node of the cluster and runs independently, after which the results from each node are integrated into the general database, which is given to the user.

The main point, reducing the effectiveness of the approach described above, is different processing time for each rule and additional time associated with compilation of the source file. In other words, compilation of the whole context and its load will be performed even for the simplest rule. Therefore the approach where all rules are performed separately may lead to not entirely rational use of computing resources.

Two solutions can be offered to resolve this problem – on the user's side and on the side of the system. From the viewpoint of the user the rules can be combined into groups. For this purpose the GROUP operator which is available in SVRF format can be used. Then not a separate rule, but a group of rules will be sent to each node. The user will have the opportunity to manage this group.

The second variant is that the system of tasks run on the cluster must collect statistics about separate rules runtime. At the first start the system runs each rule separately, at a repeated start short rules are grouped on the basis of collected statistics, but long rules are run separately.

You can also use the combination of the first and the second approaches. In this case the user gets the opportunity to decide independently when to start the profiling and when to use the existing grouping.

The use of the cluster will be effective only if there is no rule that significantly exceeds the processing time of other rules or sets of rules. As a separate rule cannot be parallelized, the time of the whole file processing will be determined by the longest rule.

Fig. 2 shows the scheme of software operation for DRC rules verification.

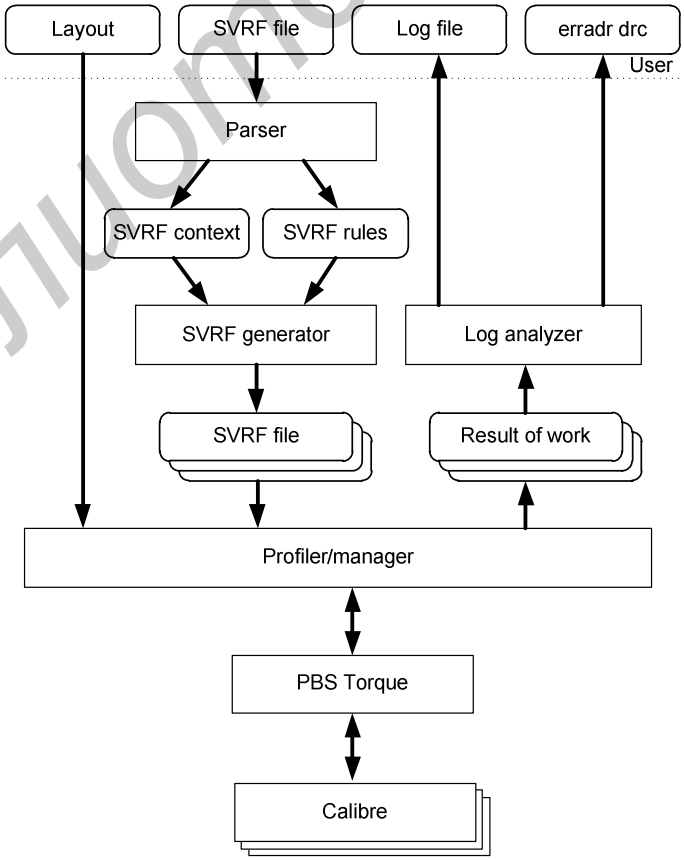

Figure 2 – Scheme of software operation for DRC rules verification

Java programming language has been used for program development because performing of a significant amount of calculations hasn't been required. The parser is located at the top level. Its main task is parsing of SVRF input file. Conventionally the file is divided into two parts – SVRF context and rules. Context is conditionally constant. The output files are generated by SVFR generator based on the context and the rules. Context and few rules are certainly included into each output file.

Management of the output files generation process can be carried out by the user or by module of manager. In initial SVRF the user can create groups of rules. Then in the process of parsing data about groups will also be extracted and transferred to the generator.

The parser performs processing in two stages. During the first stage all the rules and rule groups are extracted. The names of rules and groups are stored in the symbol table. At the second stage check showing that such names are not found anywhere else in the generated context is carried out. Otherwise the program operation is stopped because clean context couldn't be formed. A clear context is a context without names of rules and their derivatives (rules group names), so it can be used with any rule. SVRF generator creates output files based on groups of rules that are provided to it by the parser or the manager. The generator operation algorithm is simple – the context is placed at the beginning of each output file and the group of rules is placed behind it.

Generated SVRF files are sent to the manager. Task of the manager is copying of all the files required for run to the cluster, start of verification, measurement of each file verification time, collection of results from the cluster and their transfer to the log analyzer. SSH protocol and OpenSSH client are used to access the cluster. PBS Torque is used to run tasks on the cluster.

V. CONCLUSIONS

Real digital and digital-to-analog devices with high and ultrahigh degree of integration are substantially "parallel" objects, whereas the software in CAD structure intended to simulate and verify them are successive to a greater extent. In this connection the development of parallel simulation and verification technology for digital electronic devices by using supercomputing infrastructure seems to be quite perspective.

The developed technology and the software allow to reduce costs and the development time of VLSI topology as a result of multiple reduction of the time required on the process of VLSI simulation and topology verification.

REFERENCES

[1] ISE Design Suite, http://www-mtl.mit.edu/researchgroups/Well/.
[2] http://www.thefreelibrary.com/Model+Technology+and+Platform+Computing+Corp.Dramatically+Increase.

APPROACHES TO IMPLEMENTATION OF THE ION-SENSITIVE FIELD-EFFECT TRANSISTOR COMPACT MODELS

Dao Dinh Ha
Le Quy Don Technical University, Hanoi, Vietnam, ha.dao.dinh@bsuir.by

I. INTRODUCTION

The detection of toxic chemicals/biomolecules is of paramount importance for medical applications, environmental monitoring, food and pharmaceutical industries. Chemical/biological sensors based on ion-sensitive field-effect transistor (ISFET) offer several advantages, such as, higher sensitivity, lower cost, and smaller size. In this paper, different FET structures and detection principles are discussed. Conventional structure with the gate metal removed and an optional recognition element membrane deposited on top of the gate. We develop a physics-based (Verilog-A) compact model of ISFET sensors. The Verilog-A implementation would allow ISFET integration with complex signal processing circuits, and prediction of the integrated performance.

Unfortunately, very poor reliability/stability of these sensors in the fluidic environment has been a key roadblock to the commercialization of technology. Figure 1-A shows the structure of a traditional ISFET with the gate metal removed and an optional recognition sensing layer deposited on top of the gate [1]. The sensing layer of the transistor is directly exposed to the ionic solution. Ions from the solution can penetrate