2021 International Seminar on Electron Devices Design and Production (SED)

The Basics of Digital System Designs Simulation for Hardware/Software Joint Debugging

Alexander Ivannikov
The Institute for Design Problems in Microelectronics of
Russian Academy of Sciences
Moscow, Russian Federation
adi@ippm.ru

Alexander Stempkovskiy
The Institute for Design Problems in Microelectronics of
Russian Academy of Sciences
Moscow, Russian Federation
stal09@ippm.ru

***Abstract*—Based on the analysis of the features of modern digital systems, the structure of a system for joint hardware/software simulation on functional-logic level are described. The purpose of the simulation is to debug a design of digital system before production. A hardware mathematical model is proposed, taking into account the peculiarities of modern digital systems, namely: the presence of bidirectional buses, high-impedance state of the outputs, internal memory of blocks. It is shown that the joint hardware/software simulation is reduced to solving systems of logical equations at each cycle of simulation.**

Keywords*—digital systems design, functional logic simulation, design debugging, hardware mathematical model, logical equation systems, solving by iteration

I. Introduction

In the last decade, foreign computer-aided design (CAD) systems are mainly used for the design of domestic digital microelectronic systems. These systems are well developed and provide efficient and high-quality design. They are accredited by foreign factories that manufacture integrated circuits.

However, in modern conditions the task of import substitution is very acute. Even in the case of the manufacture of integrated circuits abroad, it is desirable that the design was carried out using domestic means that guarantee the correct functioning of the product in all modes. It is especially important for modern domestic developments to take into account, when designing, various destabilizing effects on the designed products and ensure their high stability in this case. In this regard, the task of developing methods, algorithms and automation tools for the design of digital super large integrated circuits (VLSI) for special purposes, highly resistant to various destabilizing influences, as well as the creation of domestic CAD systems on this basis, is an urgent task.

In this article, we will consider only one stage in the design of digital systems, namely the joint debugging of hardware and software and firmware using the simulation method [1-4]. At the same time, in order to create the specified subsystem of the domestic CAD system, it is necessary to formulate the requirements and basic principles for constructing this CAD subsystem, taking into account the accumulated design experience, which is the task of this work.

The experience gained at present in digital circuits simulation can be used to build a system for digital systems simulation intended for joint debugging of hardware and software and firmware [5-8].

II. Features of Modern Digital Systems that Determine the Requirments for a Simulation System

The task of joint development and debugging of hardware and software or firmware of digital systems at the design stage can be solved by their joint simulation. The debugging process in this case consists of the following stages [9-11]:

- obtaining a software model (emulator) of a digital system hardware or its blocks;
- compilation of a program or microprogram into a sequence of logical signals fed to the inputs of a digital system or its blocks, or a set of states of memory cells;
- cycle-by-cycle simulation of the operation of a digital system or its blocks;
- analysis of output and internal logic signals of a digital system or its blocks, current states of memory cells and registers.

Let us formulate the features of modern digital systems that determine the basic principles of building a simulation system of the considered level (Fig. 1).

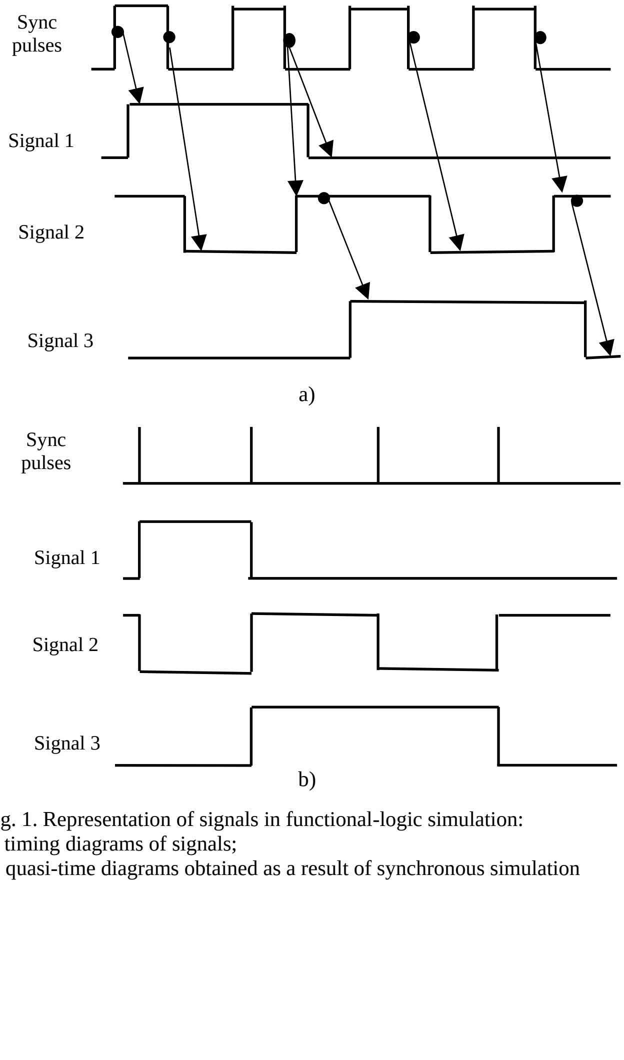

Fig. 1. Representation of signals in functional-logic simulation:
a) timing diagrams of signals;
b) quasi-time diagrams obtained as a result of synchronous simulation

978-1-6654-3941-1 @2021 IEEE

1. When debugging a digital system, the developer must be able to control the logic signals at the LSI block outputs, as well as monitor changes in the states of the system's memory elements - memory cells and registers. In this regard, the simulation of the entire digital system should be carried out at the level of logical signals at the LSI block outputs, and the block simulation at the functional level, that is, to provide an adequate description of the logical signals at the LSI block outputs and register states [12-14]. In this case, the adequacy of the structure of the block model to its internal logical structure is not obligatory.

2. The vast majority of digital systems are built in such a way that all changes in logic signals caused by the arrival of a sync pulse or some other input signal are completed before the arrival of the next sync pulse or input signal. In this regard, it is advisable that the stage of checking timing diagrams to determine the required duration of the synchronization cycle, identifying "races" and risks of failures should precede the stage of joint debugging of the structure of hardware and software and firmware, that is, the stage of structural and logical debugging [15, 16]. This separation makes sense because of the different debugging strategies used in the two phases. When checking timing diagrams, it is possible not to distinguish between commands of the same type, but to simulate only the intervals between the appearance of signals [17-20]. At the same time, for structural and logical debugging, in order to save machine time, it is preferable to use synchronous functional logical modeling. In the process of such simulation, the developer can obtain quasi-time diagrams of logical signals at the outputs of all LSI blocks (Fig. 2).

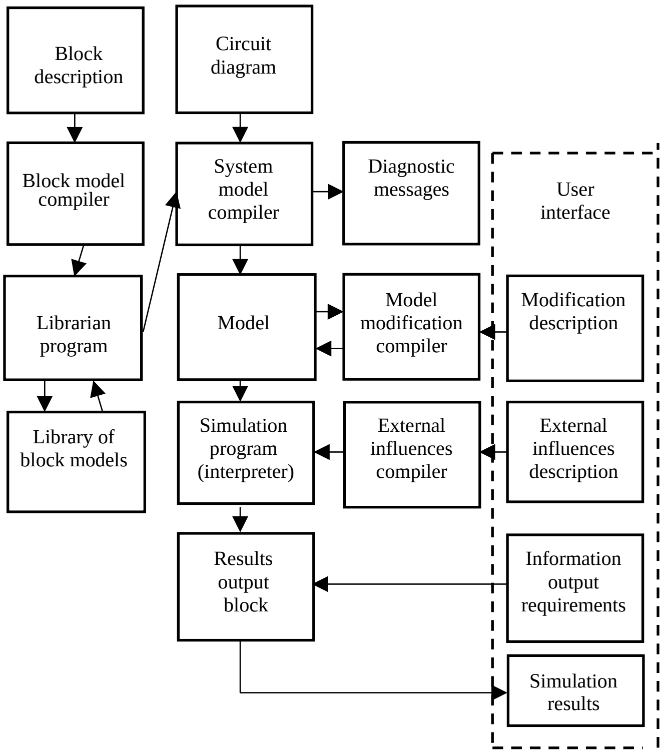

Fig. 2. The structure of the system of functional-logic simulation of digital systems

3. A digital system contains, as a rule, from several LSI blocks to several hundred LSI blocks, which are complete subsystems. In this regard, the functional diagram of the technical means is actually a schematic diagram and can be quite simply specified by a list of electrical circuits.

4. Since a limited set of LSI blocks and components is used in the design of a digital system, a description of their functioning can be performed once. The resulting software models must be entered into the library of general-use block models.

5. The requirement for the convenience of dealing with the hardware model, the need for easy changes in this model determines the interpretive nature of the simulation. In this case, the hardware model is a certain data structure containing information about the blocks connections, the blocks internal states and the names of the signals.

6. The overwhelming majority of digital systems have a bus structure. All lines, for example, an address bus or a data bus, are connected in the same way. In addition, it is more convenient to analyze information on the bus lines and in the registers, considering this information as one variable with 2^n states, where *n* is the number of lines in the bus. In this regard, when specifying a connection diagram and in modeling, the signals of lines combined into buses should be considered as one multi-valued signal, and the contents of multi-digit registers - as one variable [4, 21]. One of the states of the bus signal, in particular, may be a state with a high output impedance. This approach allows you to reduce the amount of memory required to store the values of signals and variables, since for storing the values of signals and variables, multiples of a byte of memory are used.

7. Models of digital LSI blocks should take into account the bidirectionality of a number of buses and lines, when, depending on the internal state of the blocks, the same pin can be both input and output.

8. When connecting the conclusions of several blocks, various logical functions can be implemented, namely: "wire AND"; wired "OR"; combining, typical for systems with a common bus, when only one output is active, and the rest must be in a state with high output impedance; prohibition of combining block outputs. A simulation system based on the type of block outputs being combined should distinguish between these cases and, in case of violation of the established rules (for example, when conflicting signals appear at the block combined outputs), give an error indication.

9. Due to the high laboriousness of developing LSI models, specialized languages for describing the functioning of LSI blocks and translators from these languages are needed.

10. When designing digital devices, you can also use already debugged blocks containing several blocks of a lower level. In this case, it is advisable to use functional macromodels of blocks that do not allow changes and do not provide access to signals at the internal nodes of the block, but reduce the computer time compared to simulating the block as a connection of separate subblocks [14, 22, 23].

11. An integral part of the simulation system is the programming and microprogramming automation subsystem. Strict requirements for the timing characteristics of the software and firmware of specialized digital systems necessitate its development in assembly language. However, in some cases it is advisable to use high-level languages. The programming automation subsystem includes an assembler, a micro-assembler, and translators from high-level languages,

which provide the states of the memory cells in which the program is stored, or the signal sequences coming from the microprogram memory. When developing firmware, it is advisable to have a micro-assembler tuned to a series of microprogrammed blocks and a specific digital system. The system is configured by setting the size of the control fields, assigning semantic mnemonic symbols to different field values, and defining the default field values.

12. The system for joint simulation of hardware and software and firmware should have convenient means for setting the conditions for issuing information about logical signals and states of registers and memory cells, modifying these states, setting tracing modes, for storing and restoring the state of a digital system, automated generation of test examples [5, 12, 13, 15].

III. The Proposed Mathematical Model of Digital Systems Hardware

Consider a mathematical model of a digital system hardware, compiled taking into account the formulated requirements.

The LSI model or block m includes the following components.

1. The set of variables $\mathbf{P}$, which we will call terminal variables. Each terminal variable p can take values from a finite set $\mathbf{Z}_p$. Terminal variables correspond to signals at the outputs of the LSI blocks.
2. The set of internal variables $\mathbf{R}$, each of which can take values from the finite set G_r. Internal variables correspond to block registers. The Cartesian product $\prod_{r\in\mathbf{R}}\mathbf{G}_r$ forms the set of internal states of $\mathbf{A}$.
3. Mapping the sets $\mathbf{A}$ into the set of subsets $\mathbf{P}$, that is, $\Gamma\colon \mathbf{A}\to\{\mathbf{P}'\}, \mathbf{P}'\in\mathbf{P}$, where $\mathbf{P}'$ is the set of input variables, $\mathbf{P}''=\mathbf{P}/\mathbf{P}'$ is the set of output variables, $\mathbf{X}_a=\prod_{p'\in\mathbf{P}'}\mathbf{Z}_{p'}$ - set of input states, $\mathbf{Y}_\mathrm{a}=\prod_{p''\in\mathbf{P}/\mathbf{P}'}\mathbf{Z}_{p''}$ - set of output states.
4. For each $a\epsilon\mathbf{A}$, the following mappings are defined: $H{:}(\mathbf{X}_a,\mathbf{A})\to\mathbf{Y}_a$; $W{:}(\mathbf{X}_a,\mathbf{A})\to\mathbf{A}$.

Thus, the model is $m=(\mathbf{P},\{\mathbf{Z}_p\},\mathbf{R},\{\mathbf{G}_r\},\Gamma,H,W)$.

When the models are combined, a network S is formed, which includes the following components.

1. The set of models $\mathbf{M}=\{m_i\}$. We denote by $\mathbf{V}=\bigcup_i\mathbf{P}_i$ the set of terminal variables of all models.
2. The set of nodes $\mathbf{U}$.
3. The set $\mathbf{P}_\mathrm{S}$ of terminal variables of the network, each of which can take values from a finite set $\mathbf{P}_\mathrm{S}$ Z.
4. The mapping $\mathrm{Q}{:}\mathbf{V}_\mathrm{S}\to\mathbf{U}$, where $\mathbf{V}_\mathrm{S}=\mathbf{V}\cup\mathbf{P}_\mathrm{S}$, is such that all terminal variables $v_\mathrm{S}\in\mathbf{V}_\mathrm{S}$ mapped to the same element u have the same range of values $\mathbf{Z}_u$, that is, $(\forall u)(u=\mathrm{Q}(v'_\mathrm{S})\&u=\mathrm{Q}(v''_\mathrm{S})\to\mathbf{Z}_{v'_\mathrm{S}}=\mathbf{Z}_{v''_\mathrm{S}}=\mathbf{Z}_u$.
5. Partially defined union functions $\mathbf{F}_u(v_{\mathrm{S}1},\dots,v_{\mathrm{S}j},\dots)$, given for each node $u\in\mathbf{U}$. Here $v_{\mathrm{S}1},\dots,v_{\mathrm{S}j},\dots$ are all output variables of the models and input variables of the network mapped to the node u. The range of $\mathbf{F}_u$ is $\mathbf{Z}_u$.

Thus, $\mathbf{S}=(\mathbf{M}, \mathbf{U}, \mathbf{P}_\mathrm{S}, \{\mathbf{Z}_p\}, \mathrm{Q}, \{\mathbf{F}u\})$.

The block diagram of the algorithm for digital systems simulation using the introduced models is shown in Fig. 3.

The network S is a model $\widetilde{\mathbf{M}}$ of some block under the following conditions [24].

1. The set of internal variables $\widetilde{\mathbf{R}}\supseteq\bigcup\mathbf{R}_i$. Each of the internal variables can take values from $\mathbf{G}_{\tilde r}$, $\tilde r\in\widetilde{\mathbf{R}}$. Then $\widetilde{\mathbf{A}}\supseteq\prod_i\mathbf{A}_i$.
2. The set of terminal variables $\widetilde{\mathbf{P}}=\mathbf{P}_\mathrm{S}$, and the equality holds:

$$\mathbf{P}=\mathbf{P}_\mathrm{S}=\left\{p_S\left|\begin{array}{c}(\forall u)((\exists p_S)\;u=\mathrm{Q}(p_S)\\ \cup(\forall\tilde a)\times(\exists i,v)v\in\mathbf{P}_i'',u=\mathrm{Q}(v))\end{array}\right.\right\}.$$

This condition requires that each node u, for any state $\tilde a$, correspond to at least one output variable of some model or at least one terminal variable of the network.

3. The mapping $\tilde\Gamma{:}\widetilde{\mathbf{A}}\to\{\widetilde{\mathbf{P}}'\}$, $\widetilde{\mathbf{P}}'\subseteq\widetilde{\mathbf{P}}$, assigns to each $\tilde a\in\widetilde{\mathbf{A}}$ variable set

$$\widetilde{\mathbf{P}}'=\left\{\tilde p'\left|\begin{array}{c}(\forall u)\;(\exists p_S')\;u=\mathrm{Q}(p_S')\\ \cup\;(\forall\tilde a)\times(\exists i,v)\;v\in\mathbf{P}_i'',u=\mathrm{Q}(\mathrm{v}))\end{array}\right.\right\}$$

Moreover, for any $\tilde a$, each node u corresponds to at least one input variable of the model $\widetilde{\mathbf{M}}$ or an output variable of some model m_i. The set $\widetilde{\mathbf{X}}_{\tilde a}=\prod_{\tilde p'\in\widetilde{\mathbf{P}}'}\tilde{\mathbf{Z}}_{\tilde p'}$ is the set of input states, the set $\widetilde{\mathbf{Y}}_{\tilde a}=\prod_{\tilde p''\in\widetilde{\mathbf{P}}/\widetilde{\mathbf{P}}'}\tilde{\mathbf{Z}}_{\tilde p'}$ is the set of output states.

4. The values of the input variables of the models m_i, namely $p'\in\mathbf{P}_i'$, are equal to the values of the functions $F_u(v_{\mathrm{S}1},\dots,v_{\mathrm{S}j},\dots)$, where $u=Q(p')$, and the values of the output variables of the network $\tilde p''\in\widetilde{\mathbf{P}}''$ - the values of the functions $F_\mathrm{u}(v_{\mathrm{S}1},\dots,v_{\mathrm{S}j},\dots)$, where $u=Q(\tilde p'')$, and $v_{\mathrm{S}1},\dots,v_{\mathrm{S}j},\dots$ take only those values for which F_u are defined.
5. Maps $\widetilde{\mathrm{H}}{:}(\widetilde{\mathbf{X}}_{\tilde a},\widetilde{\mathbf{A}})\to\widetilde{\mathbf{Y}}_{\tilde a}$; $\widetilde{\mathrm{W}}{:}(\widetilde{\mathbf{X}}_{\tilde a},\widetilde{\mathbf{A}})\to\widetilde{\mathbf{A}}$ exist.

Let us consider condition 5 in more detail. For the existence of the mappings $\widetilde{\mathrm{H}}$, $\widetilde{\mathrm{W}}$ it is sufficient that for known $\tilde a$ and $\tilde p'$ all terminal variables of the models m_i are defined. In this case, the existence of $\widetilde{\mathrm{H}}$ follows from the fulfillment of condition 4, and the existence of $\widetilde{\mathrm{W}}$ from the existence of the mappings W_i.

The mappings $\mathrm{H}_i{:}(\mathbf{X}_a,\mathbf{A})\to\mathbf{Y}_a$ for a fixed $a\epsilon\mathbf{A}$ are a set of multivalued logical functions $\mathbf{Y}_i=\mathrm{F}_i(\mathbf{X}_i)$, where $\mathbf{X}_i$, $\mathbf{Y}_i$ are vectors of input and output variables of the i-th model. Then for determined $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{P}}'$ all variables $\mathbf{V}$ are defined if the following system of multivalued logical equations has a solution [25]:

$$\begin{array}{l}v_{ik}=f_{ik}(v_{i1},\dots,v_{i\mathbf{L}_i}),\ i=1,\dots,\mathrm{N};\ k=1,\dots,\mathrm{K}_i;\quad(1)\\ v_j=\mathrm{F}_{u_j}(v_{j1},\dots,v_{j\mathrm{T}_j}),\ j=1,\dots,\mathrm{J},\end{array}$$

where N is the number of models m_i;
K_i is the number of output variables of the model m_i;
J is the number of nodes in the set $\mathbf{U}$;
T_j is the number of output variables of the models mapped to the u_j node;
f_{ik} is a multivalued logical function that defines the dependence of the output variable v_{ik} on the input variables of the model m_i;
F_{u_j} is the union function at the u_j node.

The algorithm for digital systems simulation using the introduced models consists of the following steps.

1. Assigning initial values to the $\mathbf{R}_i$ variables.
2. Determination of the values of the input.
3. Checking conditions 3 and 4. If conditions are not satisfied it means that models m_i union is not correct. Stop simulation. If there is no errors go to the next step.

4. Solving equations (1). If there is no solution that means models m_i union is not correct. Stop simulation. If there is a solution go to the next step.
5. Values of output variables $\mathbf{P}''$ calculation.
6. Determination of new values of $\mathbf{R}_i$.
7. If simulation is not finished go to step 2, otherwise stop simulation.

IV. Conclusion

The proposed structure and mathematical model are the basics for building can a domestic system for joint simulation of hardware and software at the design stage for debugging and comparative analysis of options. We hope that in the next paper we will be able to discuss experience of developed system applications.

References

[1] J. Shi, W. Liu, M. Jiang, “Software Hardware Co-Simulation and Co-Verification in Safety Critical System Design”, 2013 IEEE International Conference on Intelligent Rail Transportation (ICIRT), pp. 71–74. DOI: 10.1109/ICIRT.2013.6696270

[2] Y. Gao, L. Liu, H. Du and Q. Gong, "Software and Hardware Co-Verification Technology Based on Virtual Prototyping of RF Soc", 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET), Beijing, 2018, pp. 244-247. DOI: 10.1109/CCET.2018.8542186

[3] M.D. Nguyen, «Hardware/software formal co-verification using hardware verification techniques”, Fourth Int. Conf. on Communications and Electronics (ICCE), 2012, pp. 465-470. **DOI:** 10.1109/CCE.2012.6315951

[4] A.D. Ivannikov, A.L. Stempkovskiy, “Formal mathematical representation for the task of digital system projects debugging”, Informacionnie Technologii, 2014, no. 9, pp. 3—10 (in Russian).

[5] A. Matsuda, T. Ishihara, “Developing an integrated verification and debug methodology”, Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011, pp. 1–21. DOI: 10.1109/DATE.2011.5763087

[6] Y. Lin, A. Su, “Functional Verifications for SoC Software/Hardware Co-Design: From Virtual Platform to Physical Platform,” 2011 IEEE International SoC Conference (SOCC), pp. 201–206. DOI: 10.1109/SOCC.2011.6085104

[7] Y. Choi and J. Cong, "HLScope: High-Level Performance Debugging for FPGA Designs", 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM*)*, Napa, CA, 2017, pp. 125-128. DOI: 10.1109/FCCM.2017.44

[8] A.L.Stempkovsky, A.D.Ivannikov, “Formal Description of Digital Control System Operation and Its Use in Designing, Russian Microelectronics, 2019, vol. 48, no.5, pp. 318-325. DOI: 10.1134/S1063739719050093

[9] R. Willenberg and P. Chow, "SimXMD: Simulation-based HW/SW co-debugging", 2013 23rd International Conference on Field programmable Logic and Applications, Porto, 2013, pp. 1-1. DOI: 10.1109/FPL.2013.6645632

[10] T. Chang, S. Hou and I. Huang, "A unified GDB-based source-transaction level SW/HW co-debugging", 2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS*)*, Jeju, 2016, pp. 506-509. DOI: 10.1109/APCCAS.2016.7804015

[11] A. Jasnetski, S.A. Oyeniran, A. Tsertoy, “High-Level Modeling and Testing of Multiple Control Faults in Digital Systems”, IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2016, paper # 7482445. DOI: 10.1109/DDECS.2016.7482445

[12] S. Jain, P. Govani, K.B. Poddar, A.K. Lal, R.M. Parmar, “Functional Verification of DSP Based On-Boad VLSI Design”, International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA), 2016, pp. 1–4. DOI: 10.1109/VLSI-SATA.2016.7593030

[13] A.M. Cruz, R.B. Fernandez, H.M. Lozano, M.A. Ramirez Salinas, L.A. Vila Vargas, “Automated Functional Test Generation for Digital Systems Through a Compact Binary Differential Evolution Algorithm”, Journal of Electronic Testing-Theory and Applications, 2015, vol. 31, no. 4, pp. 361–380. DOI: 10.1007/s10836-015-5540-6

[14] S.V. Gavrilov, A.D. Ivannikov, A.L. Stempkovsky “Method of mathematical description for digital system blocks logical models”, Problems of Perspective Micro- and Nanoelectronic Systems Development, 2019, no. 2. pp. 8-11. DOI: 10.31114/2078-7707-2019-2-8-11

[15] A. Ivannikov, B. Pozdneev, I. Romanova and S. Tumkovskiy, "Debugging test set generation for digital control system functions", 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT), Moscow, 2018, pp. 1-5. DOI: 10.1109/MWENT.2018.8337238

[16] Y. Tai, W. Hu, Lantian Guo, B. Mao and D. Mu, "Gate Level Information Flow analysis for multi-valued logic system", 2017 2nd International Conference on Image, Vision and Computing (ICIVC*)*, Chengdu, 2017, pp. 1102-1106. DOI: 10.1109/ICIVC.2017.7984724

[17] S.V. Gavrilov, O.N. Gudkova, A.L. Stempkovskiy, “The Analysis of the Performance of Nanometer Intellectual Property Blocks Based on Interval Simulation”, Russian Microelectronics, vol. 42, no. 7, 2013, pp. 396–402. DOI: 10.1134/S1063739713070068

[18] P.M.Maurer, “Time-parallel multi-dalay logic simulation”, 2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS*)*, Pasadena, CA, 2016, pp.1-7. DOI: 10.23919/TMS.2016.7918823

[19] G. Ivanova, A. Korshunov, T. Zhukova, I. Tiunov “Development of mathematical models for standart cell timing analysis at the 28 nm technology and below”, Proc. 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conf., ElConRus 2017, 2017, pp.1401-1405. DOI: 10.1109/EIConRus.2017.7910832

[20] D.A. Zheleznikov,M.A. Zapletina, V.M. Khvatov. “The rip-up and reroute technique research for physical synthesis in the basis of reconfigurable socs”, Problems of Perspective Micro- and Nanoelectronic Systems Development, 2018, no. 1, pp. 188-192 (in Russian). DOI: 10.31114/2078-7707-2018-1-188-192

[21] I. Považan, M. Krnjetin and N. Četić. “Communication interface libraries as an extension to the debugging framework for DSP applications”, *2015 23rd Telecommunications Forum Telfor (TELFOR)*, Belgrade, 2015, pp. 1005-1008. DOI: 10.1109/TELFOR.2015.7377635

[22] J. Chatterjee, A. Saxena, A. Mehra, G. Vyas and V. Mukesh, "Verification and Debugging of LC-3 Test Bench Environment using System Verilog", 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA*)*, Coimbatore, 2018, pp. 1253-1258. DOI: 10.1109/ICECA.2018.8474724

[23] S. Kunapareddy, S. D. Turaga and S. S. T. M. Sajjan, "Comparision between LPSAT and SMT for RTL verification", 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], Nagercoil, 2015, pp. 1-5. DOI: 10.1109/ICCPCT.2015.7159418

[24] A.D. Ivannikov, A.L. Stempkovskiy, “Basic principles of digital system design simulation for hardware/software debugging”, Information Systems and Technologies, 2018, no. 6 (110), pp.13-19 (in Russian

[25] A. Ivannikov, "Iterative Methods for Multi-Valued Logical Equation System Solving while Digital System Simulating," 2020 IEEE East-West Design & Test Symposium (EWDTS), Varna, Bulgaria, 2020, pp. 1-6. DOI: 10.1109/EWDTS50664.2020.9224976