About the features of the verification of VLSI class “System on a chip” for complex information systems

A B Petrov, I E Tarasov

MIREA - Russian Technological University, 78, Vernadsky ave., Moscow, 119454, Russia

E-mail: tarasov_i@mirea.ru

Abstract. The article discusses the hierarchical levels of the verification process for VLSI of the System-on-Chip class, as well as information systems built on their basis. The increasing complexity of modern information systems creates additional requirements for the composition and complexity of the verification of such systems, including the early stages of their development. An approach to verification is proposed using a combination of software tools for modeling digital circuits at the lower levels of abstraction and specialized verification tools developed for a specific information system and implementing verification primarily at the system level. The proposed approach will enhance the performance of VLSI modeling of the System-on-Chip class, which will improve the reliability of verification of complex systems by improving test coverage. It is noted that when connecting mutually asynchronous fragments of VLSI it is necessary to carry out modeling at a high level, able to identify problems of data transmission between individual synchronous subsystems.

1. Introduction
The increasing complexity of information systems (IS) and VLSI, which form the basis of their hardware, increases the importance of their verification at high levels of abstraction. The system-on-chip class VLSIs themselves are complex objects for verification. And their integration into an information system adds additional levels of verification, determined by the interaction between the components of a complex system and, possibly, issues of interaction with the “real world” (for example, network activity, physical effects the system as a whole and other effects that cannot be considered at the VLSI level). The issues of building complex ICs and related additional questions are currently the subject of research from the point of view of both digital circuitry [1-8] and system engineering [9, 10].

Currently in the field of information systems there are several levels of testing and verification, including:

- unit tests (unit test), performed at the VLSI level or its components;
- integration tests performed to verify the correctness of the connections between the VLSI or IS components and their interaction;
- system tests performed to control the correctness of IP actions when interacting with control objects in a real environment.
The practical effect of identifying levels of verification is the possibility of applying different approaches and tools at each level, and with increasing levels of abstraction, the performance of verification or modeling increases due to the rejection of checks of low-level parts that were verified before. However, an important task is to identify the level of detail of the model, which would be sufficient at an appropriate level.

2. Causes of malfunction in VLSI forming information systems
Due to the high complexity of ICs, which include not only VLSI, but also peripheral devices that interact with some environments, the sources of errors in the work are of a different nature. You can consider these reasons, grouping them by the level of system design, which is possible to analyze them and eliminate the causes.

At the level of VLSI or their modules, the occurrence of structural defects caused by both architectural errors and technological errors in the manufacture is possible. In addition to an improperly designed circuit, the problem may be caused by a combination of temperature, supply voltage and variation of technological parameters in the manufacture of a specific VLSI sample, which will cause its behavior to deviate from the expected one. This type of errors in modern CAD systems is subject to monitoring and elimination with the help of tools that are combined with the concept of ‘Static Timing Analysis’ (STA).

![Diagram](image.png)

**Figure 1.** An example of analyzing the propagation time of a signal with respect to design constraints for VLSI with an FPGA architecture.
The metastable state is understood as such a trigger operation mode in which its output has an intermediate voltage level, perceived by different components as a logical zero or a logical unit (Fig. 3). This is an undesirable mode in which the triggers are forced, if the voltage at the data input changes immediately before the arrival of the clock signal. According to the specification, each trigger has a setup time (setup time) and a hold time (hold time) - the time intervals before and after the clock edge, during which the signal level at its input D should not change. Metastability is probabilistic in nature and is explained by the peculiarities of switching transistors, of which the trigger consists, and not by any special design techniques for digital electronics. The signal can be metastable, input, for example, from a mechanical switch, or any external chip, which is clocked not from an internal clock generator.

Any design that uses two clock signals is potentially subject to metastability, which should be eliminated, or at least reduce the likelihood of its occurrence. Even in the case when the nominal values of the clock frequencies are the same or multiples, normal work is possible only in the ideal case unattainable in practice - when the fronts of both clock signals appear at strictly certain points in time.
3. VLSI Verification Levels in Information Systems

For most synchronous components in VLSI, Static Timing Analysis execution is acceptable. Since the design constraint set for VLSI clock circuits (create_clock) is analyzed by modern CAD systems with a sufficiently high level of adequacy, obtaining positive STA test results can be the basis for conclusions about the reliable operation of the synchronous node as a whole.

However, it should be borne in mind that even at the level of individual VLSIs, there is currently a tendency to separate clock domains in order to reduce the complexity of tracing clock circuits over a large area crystal.

A common mistake is to attempt to simulate a project to prove that there are no problems with metastability. The modeling performed by the CAD system of the VLSI is not able to reveal the problem of the appearance of metastability, since this process is of a probabilistic nature and, moreover, is not purely digital. Even the identification of the fact that for some kind of trigger, conditions are violated by the time of installation or holding a signal does not help to answer the question ‘what state will this trigger take into account temperature, supply voltage and technological variation of parameters’.

A practical way to solve the problem of metastability is the correct resynchronization of signals transmitted from one clock domain to another. Dual port memory should be used with ports connected to the appropriate clock signals, where possible. A reliable way to transmit a one-bit signal is to use a chain of 2 or 3 flip-flops, and the optimization of this circuit by a synthesizer should be avoided. It is recommended to describe resynchronization nodes as separate modules and check the details of their implementation after synthesis.

Based on the above information, verification of the correct operation of the resynchronization scheme cannot be performed at the CAD VLSI level. Therefore, verification of such a project should be performed at the integration or system level, abstracting from the behavior of specific schemes and considering the resynchronization circuit as a ‘black box’ or ‘gray box’. Such a chain can be attributed to the latency introduced by it in the receiver clock cycles, assuming that the metastable state will occur regularly. For a dual-port memory, it is necessary to simulate the appearance of the “data received” flag, taking into account the additional latency that excludes data reception, if the metastable state for the ready flag did not arise, but such a state arose for the received data.

The next level of abstraction may include such processes as receiving packets in communication systems or the appearance of information components of signals in measuring systems. Since these processes are related to the information environment of the IC, and cannot be adequately modeled at the CAD level, to simulate them, it is necessary to develop special software that takes into account the characteristics of the domain to correctly reproduce the behavior patterns external to the IC.

4. Levels of abstraction in the verification of VLSI

The considered problems of verification of projects on the basis of FPGA indicate an increase in the complexity of the verification process and the associated allocation of levels of abstraction. Levels imply the use of the following types of verification (table).

| Type         | Object of verification          | Verification tool          |
|--------------|---------------------------------|----------------------------|
| Module-level | Synchronous designs             | VLSI CAD or FPGA CAD       |
| Integration  | Systems with Cross-Domain       | 1. VLSI CAD                |
|              | Clocking or systems with many   | 2. Specially developed      |
|              | VLSIs                           | software                   |
| System level | Information systems and         | Specially developed software|
|              | environment                     |                            |

Thus, with an increase in the level of abstraction, the role of special software increases, ensuring adequate reproduction of input actions and response of control objects characteristic of the subject area.
For correct verification, it is necessary to ensure an appropriate level of test coverage. If for static time analysis the test coverage is mainly determined by CAD (including analysis of the effect of temperature, supply voltage and technological variation), then reproducing situations arising from the interaction of synchronous components of VLSI and VLSI and its environment is specifically a task developed software.

It is possibly to specify the obvious reasons for developing software designed for a specific IP or class of such an IC.

1. Network systems for which quality parameters are high-level characteristics, such as average exchange rate, percentage of lost packets, etc., which does not include analysis of individual VLSI signals. In addition, the formal correctness of the SBIS connection does not answer the questions of interaction of network infrastructure elements at high levels of the OSI model (i.e. above the MAC level, which is still subject to verification at the level of static time analysis).

2. Measuring systems, including those based on digital signal processing systems, for which metrological characteristics are relevant. Despite the fact that these characteristics can in principle be obtained on the basis of the results of static time analysis, such VLSI modeling in building AFC is too time consuming, therefore, it should be replaced with modeling at a higher level of abstraction.

For the examples given, the performance check at the system level is decisive in nature and allows you to make a decision about the feasibility of continuing development without implementing low-level project details.

5. Conclusion
When developing information systems based on digital VLSI, the allocation of abstraction levels in modelling and verification contributes to increasing development productivity and identifying possible system malfunction in the early design stages, including at the system modelling level, without involving specialists in the field of digital circuitry.

References
[1] Tanenbaum A S 2013 Structured Computer Organization, 6th Edition. (Vrije University, Amsterdam, The Netherlands. Todd Austin)
[2] Harris D, Harris S 2013 Digital Design and Computer Architecture. 2nd Edition (Morgan Kaufman)
[3] Hennessy L, Patterson D A. 2017 Computer Architecture. 6th Edition. A Quantitative Approach. (The Morgan Kaufmann Series in Computer Architecture and Design)
[4] Rowen C 2004 Engineering the Complex SOC. Fast, Flexible Design with Configurable Processors. (Prentice Hall)
[5] Nurmi J 2007 Processor Design. System-on-Chip Computing for ASIC and FPGAs. (Springer)
[6] Villarraga C 2014 Software in a hardware view: New models for HW-dependent software in SoC verification and test In Proc. of International Test Conference pp. 1-9.
[7] Ganesh R. 2014 Design Issues in Hardware/Software co-Design
[8] Lian F-L, Yook JK, Tilbury DM, Moyne J 2006 Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems IEEE Transactions on Automation Science and Engineering 3 287
[9] Denisenko V M 2014 Computerized control for technological processes, trials and facilities (in Rus.) (Moscow: Goryachay Liniya Telekom)
[10] Emelyanova N Z, Partyka T L, Popov I I 2018 Information system design (Moscow: Infra-M)