Vol. 05, No. 3 (2023) 399-416, doi: 10.24874/PES05.03.005

Proceedings on Engineering Sciences

www.pesjournal.net

SOFTWARE IMPLEMENTATION OF IC TOPOLOGY TRANSFORMATIONS FOR PROTECTION AGAINST HARDWARE TROJANS

Evgeniy Pevtsov
Tatiana Demenkova[1]
Alexander Sigov
Alexander Shnyakin
Semyon Moskolenko

Received 24.11.2022.
Accepted 24.04.2023.
UDC – 004.491.42

Keywords:

hardware Trojans, Topology Transformation, Built-In Self-Authentication Scheme

A B S T R A C T

This article focuses on software methods for detecting and preventing the introduction of hardware trojans into the topology of integrated circuits at the design stage. An additional recommendation for typical procedures of receiving GDSII files that significantly complicates the introduction of HTs into the topology is to facilitate detection methods by including special self-test structures created after performing transformations of the initial topology and ensuring the effective examination of ICs at the post-manufacture test stage. The proposed method is based on the analysis of synthesis results at the level of netlist description, its subsequent software conversion into a topological drawing, shifting standard cells and filling created voids with special self-test cells. The method is simple to implement during the topology design stage and can be recommended for use in design centres for machine learning of computer-aided design systems.

1. INTRODUCTION

Intentional and malicious modifications of integrated circuits designed to change the behaviour of a circuit and achieve a certain goal are called hardware trojans (HTs) (Bhunia & Tehranipoor, 2018; Kuznetsov & Saurov, 2016a). Trojan attacks can be carried out by changing a technology (for example, by replacing the doping type), introducing analog and digital components (for example, by switching on an external chain when charging a parasitic capacitor or adding functions of a finite-state machine that are actuated from unused conditions). The article (Chaduvula et al., 2018) discusses different security techniques in the field of digital technologies, such as encryption and shared use of know-how, as well as in areas related to design and manufacture, such as physically unclonable functions (PUFs) and watermarks attached to a physical part to protect it against counterfeiting and unauthorised access. It is proposed to classify these methods depending on their effectiveness for various competition models at different stages of product development. This classification can help developers make reasoned decisions on security practices during product development.

[1] Corresponding author: Tatiana Demenkova
Email: demenkova@mirea.ru

One of the first methods of synthesising reliable microchips using unreliable CAD tools is proposed in the work (Knechtel et al., 2020). This approach allows us to use CAD tools for complex synthesis tasks and very simple reliable tools that can be used by designers to verify results and change specifications. The main idea is to limit the design tools used so that an unreliable tool has no room to add any malicious changes to the design.

Short-term and medium-term possibilities for integrating design methods based on the safety of EDA tools are presented in the work (Potkonjak, 2010). The article considers conceptual problems for the safe organisation of countermeasures against different threat vectors and security indicators.

Since there are different types and size of hardware trojans, a combination of protection methods is required to ensure an acceptable level of security, both in design at different levels of abstraction and in IC testing after manufacture. In the latter case, verification will require a combination of operation logic verification and side channel analysis to cover trojans of different types and sizes with large variations in parameters. The main development problems in this area include: mechanisms for detecting analog trojans that can carry out numerous types of activation and observation conditions, as well as a comprehensive metrics for quantifying the level of confidence that combines the analysis of design and testing methods.

In recent years, trends have included more advanced functions, such as temperature activation, radio channel and power activation, optical side channel reading, accelerated chip aging, and denial of service (DoS). According to the currently established classification (Chakraborty et al., 2009; Pevtsov et al., 2019), it is customary to distinguish hardware trojans by introduction phases, abstraction levels, activation methods, effects and locations. This classification implies that each of the above categories contains several attributes and is based on the fact that a classification should cover all trojans and should describe all methods of how they cause harm.

The classification by introduction method describes the stages of the design route where a project is vulnerable to malicious modifications.

The classification by abstraction level describes the different stages of IC development, including IC introduction before their manufacture. This level covers both the design of topology (i.e. at the physical level) and the determination of a netlist and communication protocols used in an IC (i.e. at the system level).

The activation method describes the means by which an undeclared function is activated. This includes permanently active backdoors, as well as trojans that require certain triggers to be activated, such as internal sequential counters or external triggers from input data flows.

The category of harmful effects describes the undesirable result of a malicious change, which can manifest itself as a change in functions or even a complete failure of a microchip.

The location category describes where a change in the initial design can be physically implemented within a mi-crochip or equipment. In this case, the target of an attack may be either one component, for example, a system clock with introduced timing errors, or several complex components, for example, processors for changing the order of command execution.

A new classification is also proposed, based on the location of IC protection means (the chip itself or its packaging) and the type of protection: protection against unauthorised access or authentication (Fievrea et al., 2015). It is concluded that security methods should be aimed at meeting the requirements of new technologies, such as 3-D systems-on-chip and wearable devices. It is noted that it is promising to use new optical means, such as arrays of nanowires or a method where technical defects are used to create “fingerprints” as information encoded in the form of electromagnetic radiation from a working circuit.

2. PREVIOUS WORK

Since all modern VLIC developments are carried out with a certain degree of participation of other companies, a prerequisite for ensuring the reliable operation of these products is to develop and perform special organisational and technical activities to analyse, identify and eliminate possible design distortions in the form of software and/or hard-ware backdoors directly during the design or manufacture stages.

The website https://www.trust-hub.org regularly updates a collection of hardware trojan tests that have been developed and updated by researchers in the hardware security and trusted design community.

The works regarding IC countermeasures consider the issues of both detecting (Huang & Wang, 2020; Khairallah et al., 2018; Kuznetsov Saurov, 2016b; Kuznetsov Saurov, 2017a; Moein et al., 2017; Yuan et al., 2021) and preventing hardware backdoors to protect ICs (Chakraborty & Bhunia, 2009; Herder et al., 2014; Kulkarni et al., 2016; Kuznetsov & Saurov, 2017b; Shekarian et al., 2013).

The authors of the publication (Moein et al., 2017) note that hardware trojans have a number of properties that can be used to systematically develop detection methods. Based on this concept, they present a detailed overview of modern methods for detecting trojans and

the characteristics of existing hardware trojans, and also propose a new approach to identifying and classifying hardware trojans. This identification can be used to compare the risk or seriousness of trojan programmes with the effectiveness of trojan detection. Identification vectors are generated for each hardware trojan and trojan detection method based on corresponding attributes: insertion methods, abstraction levels, influence effectiveness, type of logic, influence method, activation method, physical implementation at the topology level, location within the IC, etc. Based on the system of quantitative assessment of these attributes, vectors that reflect the risk or seriousness of trojan attacks and the effectiveness of their detection are determined.

Another method of detecting hardware backdoors is proposed in the work (Khairallah et al., 2018). For hardware encryption systems, there is a design method based on the integration of a specially developed sensor circuit based on a ring oscillator located around potential targets of a trojan attack without any loss of circuit performance.

Unlike the existing solutions at the IP core level, solutions presented in the work (Huang & Wang, 2020) also take into account architecture-level security threats and use a strategy of introducing distributed IP security units to ensure reliable SoC operations with unreliable IP units. Special side channel attack protection modules built into the IC, physically unclonable modules and circuit abnormal behaviour monitoring modules were installed on a FPGA platform. The experimental results show the effectiveness of the proposed approach for ensuring that a system is protected against various attacks. Built-in IP units bring low overhead expenses, do not affect main IP units and have such characteristics as flexibility, scalability and diversity.

In the context of HT detection measures, works aimed at ensuring the security of built-in processor cores deserve special attention. In particular, the work (Yuan et al., 2021) notes that hardware trojans are one of the main hardware security threats for general-purpose registers (GPRs) of processors. This article presents a new method for detecting HT attacks by comparing the conditions of GPRs with a built-in reference model in real time. The authors used this method in the design of a RISC-V core and studied its effectiveness. The experimental results show that all accidentally inserted HT attacks can be detected in real time with a delay of two cycles.

The article (Kulkarni et al., 2016) presents a method for detecting hardware trojans during the operation of a multi-core processor using machine learning. The support-vector method (SVM) is used for machine learning. A data set is generated based on the behaviour of a multi-core router in a normal mode and when a hardware trojan is activated. The article discusses various communication attacks initiated by hardware trojans, namely: spoofing, redirection of one traffic packet to a random core, routing loops. An algorithm based on the support-vector method has the detection accuracy of 94% ÷ 97%. The implemented structure for detecting trojan attacks increases the chip area by 2%.

Software and hardware methods for counteracting HT introduction are considered in the work (Kuznetsov & Saurov, 2017b). In one option in the article (Herder et al., 2014), it is suggested to use physical unclonable functions (PUFs) in low-cost applications for the authentication and generation of encryption keys. It discusses the weaknesses of PUF implementations and their use in key generation applications associated with constructing error correction schemes based on sample matching and index-based coding.

Another protection option is to use a key-based obfuscation method to ensure protection against hardware trojans (Chakraborty & Bhunia, 2009). The obfuscation method is based on changing the state transition function of this scheme by extending its reachable state space, allowing it to operate in two different modes—normal mode and obfuscated mode. This modification makes it difficult to insert difficult-to-detect trojan backdoors into the design at both the software and hardware levels. The authors presented a functional OBISA circuit (obfuscated built-in self-authentication). This circuit is connected to the initial design circuit and achieves the goal of deceiving the attacker by reducing the empty chip space and making it difficult to interpret the functional purpose of a particular circuit element. The mechanism for selectively disabling various circuit sections and the method for selecting the path of a useful signal proposed in this article reduce resources potentially expended for filling.

The work (Shekarian et al., 2013) suggests a method called DfHT (Design-for-Hardware-Trust). The purpose of this method is to construct a scheme with maximum density. Instead of non-functional filling cells, it uses functional testing ones. However, functional cells can greatly limit design traceability. This paper also presents an improvement algorithm to minimise limitations imposed on tracing.

This method, in particular, was developed in the works (Xiao & Tehranipoor, 2013; Shiet al., 2019). The work (Xiao & Tehranipoor, 2013) suggests to include built-in self-authentication (BISA) circuits in the design for protection against HT introduction. The idea of this method is to fill unused spaces in a chip with functional cells—fillers that form logic devices of combinational or sequential logic, the correctness of which is checked by the BISA device itself, as a result of which a digital signature is formed. Any change in BISA chains will result in a different signature. Thus, BISA can be used to prevent or extremely obstruct the introduction of

trojan programmes. BISA is applicable to any single-module or hierarchical design.

The work (Shi et al., 2019) suggests the joint use of the BISA concept and split manufacturing. It is noted that split manufacturing is designed to prevent IP piracy and IC cloning, but it does not prevent the untargeted introduction of hardware trojans and results in considerable overhead expenses when a high level of security is required. Built-in self-authentication (BISA) is an inexpensive method of preventing and detecting the introduction of hardware trojans, but it is vulnerable to IP piracy, microchip cloning or attempts to hack original circuit features. This article suggests an obfuscated BISA method that combines and optimises both methods so that they complement and enhance protection against both vulnerabilities, while minimising design overhead expenses to such an extent that the proposed method does not require excessive expenses for industrial-level designs. According to the authors' estimate, the proposed method more than doubles the level of security, while reducing overhead expenses from hundreds of percent to less than 13% in capacity, 5% in delay and zero percent in area compared to the best declared performance in the existing technologies.
A number of publications address the application of symmetry principles for detecting and preventing HTs.

The article (Vaikuntapu et al., 2016) suggests a method for detecting trojans inserted after the completion of development, i.e. a trojan is inserted at the topology level. Since golden ICs, guaranteed to be free of trojans, are not always available in all cases, it is relevant to develop a detection methodology that does not require any gold microchips. This work uses the concept of asymmetric path delays to detect trojans, taking into account changes in delays of symmetric pairs due to the insertion of a trojan. Proposals are formulated on the methods of detecting suspicious ICs by comparing the metrics of two symmetric signal propagation paths. It has been shown that the proposed method is quite resistant to the influence of changes in the manufacturing process. In particular, the modelling results show that the total probability of trojan detection is close to 100% with a maximum change in threshold voltage and gate length of 8% in one chip and with a variation of 10% between batch chips.

The article (Xue & Ren, 2018) suggests a novel microelectronic HT detection circuit based on timing analysis. This detection method can be applied in both combinational and sequential circuits. The proposed technique is implemented in the IBM 90 nm CMOS processor and Xilinx ISE PLD. Experiments have shown that one detection circuit embedded in the test-path can detect a HT with a size that is 2.81% of the host-circuit size with a detection probability of 90%. The probability of false positives is controlled effectively by the testing clock frequency. For 90nm CMOS ASIC tests, the ratio of the detectable HT size to the host-circuit size ranged from 2.81% to 3.37% with a detection probability of 90% at 10% FP. For PLD implementation, the ratio of the detectable HT size to the host-circuit size ranged from approximated 0.5% to 0.9% with a detection probability of 90%. The detection probability decreases with a decrease in the HT size, but can be additionally improved by applying a larger number of detection circuits on a test-path. Moreover, a ring oscillator can be introduced to estimate operating temperature and changes in the test-path process for calibrating detection parameters, which eventually increases detection probability.

A similar method based on delay analysis is proposed in the work (Jin & Makris, 2008). To reduce the overhead expenses of analysis, the authors proposed to divide the design into separate sections, each of which has much smaller dimensions than the initial dataset, but reflects the main characteristics in the initial datasets. The results of experiments with circuit sections with comparators and counters show that the detection rate of payload trojans is 100%. However, the authors note that this method does not work well in the case of trojans, the effect of which is activated only after the occurrence of their activation event.

The articles (Cui et al., 2018; Yoshimizu, 2014) suggest a two-phase technique based on signal symmetry analysis that determines delays in path pairs for HT detection. At the design stage, a full-cover path set that covers all the nets of the design is created. At the test stage, the actual path delay in the full-cover set is extracted from manufactured circuits, and the travel of signals in path pairs is compared to a reference order generated at the design stage. A mismatch between them indicates the existence of HTs. Both process changes and measurement noise are taken into account. The efficiency and accuracy of the proposed technique are confirmed by a series of experiments, including the examination of both violated path pairs caused by HTs and their false activation rate.
Therefore, currently there is no clear trend in the development of HT detection methods. The variety of detection and anti-implementation techniques proposed so far provide the opportunity to combat various types of HTs in various hardware platforms. An obvious condition for IC design for trust (DfT or DfS—Design for Security) is the preliminary analysis of possible trojan attack models and the development of appropriate detection and/or prevention measures.

Such measures, in particular, are:

- Analysis of standard element libraries used in the design, with a full disclosure of their specifications, i.e. descriptions at the level of topology, circuit diagrams and methods for checking design and verification rules.
- Application of trusted complex functional units, control of the design route, introduction

of the designed node circuit into the topology, which perform the functions of obfuscation, camouflaging, filling of the topology free spaces. Obfuscation is a method of hiding circuit functionality by inserting an additional logic-locking circuit into the design in order to conceal its functions and intended topology. Camouflaging is a method of creating indistinguishable layouts of gates using additional dummy contacts and fake connections between layers, which prevents the attacker from re-designing the circuit's netlist. When filling empty topology spaces, additional functional elements are intentionally added to the design, which form combinational logic that can be tested during the design process and whose mal-function can serve as evidence that the topology has been distorted during manufacture.

- Implementation of split manufacturing where the factory with design rates of 40...7 nm produces the main part of a circuit without revealing its functions, and the final stage of IC manufacture is carried out at a trusted factory.
- Creation of VLIC nodes based on new principles of digital electronics, in particular, based on quantum effects and microelectromechanical nodes.
- Development of special hardware measurement methods and analysis of their results for monitoring the functioning of manufactured circuits and their parameters.

3. PROBLEM STATEMENT

Depending on the attacker's potential possibilities, a HT can be introduced, in one way or another, at any stage of IC creation, however, this action can be relatively easily carried out and relatively easily detected or prevented during the design stage. At the same time, the cost of localisation and control of the development and design process is millions of times lower than the cost of localisation and control of the IC manufacture process, which makes the task of counteracting the introduction of HTs at the manufacture stage a high priority.

In order to choose the optimal method of counteracting the introduction of hardware backdoors, it is necessary to separately consider the characteristic advantages and existing disadvantages of the above methods. We can divide the methods of counteracting the introduction of hardware backdoors into two groups: DfT (design for trust) and SMfT (split manufacturing for trust).

The methods of the DfT group usually involve inserting additional circuit components, obfuscation functions into the design or reducing the empty space of a chip. This also results in additional overhead expenses in the form of increased chip area, increased dissipated power and signal propagation delays, which is a major limitation of the DfT approach. However, for some specialised mission-critical ICs produced in small batches, this method is an excellent protection option. The method can significantly improve the security of ICs.

The main idea of the SMfT methods is to share out the stages of IC manufacture among several factories. These methods allow to effectively implement mutual physical isolation of untrusted and trusted manufacturers, thereby minimising the probability of HT introduction. At the same time, the SMfT methods have a long production cycle and higher production costs than the DfT methods, and cannot be used for large-scale production. It is necessary to separately consider the logistics problems during the movement of plates between factories, and the issues of economic efficiency in case of a sharp drop in the release of suitable products in case of violating the standard technological process.

In the current conditions of extreme technological dependence in the field of semiconductor production with nanometre topological norms and the absence of appropriate trusted manufacturers in the country, technological methods of counteracting the introduction of hardware backdoors are still not applicable.

Based on the assessment of the attacker's high potential (capabilities of the enterprise/group of enterprises/state's level to develop and use special means of exploiting vulnerabilities), measures to counteract the introduction of hardware backdoors that make it difficult to analyse initial circuits and topological drawings may be ineffective due to the potential attacker's ability to extract initial design documents.

Therefore, the only group of methods applicable in this situation are methods of increasing testability and filling the area of a topological drawing. Testable design and testability improvement are currently a standard approach to VLIC design, are well studied and will not be discussed in this work.

A universal approach to counteracting the introduction of HTs into a topological drawing is to limit the free area on the chip (filling the empty space of the chip): the filling is associated with minimising the use of non-functional cells in the circuit, inserting a specially designed detection circuit (BISA) or removing the empty space, in order to limit the space where HTs can be inserted.

Summarising the known approaches to counteracting the introduction of HTs into IC topological drawings after obtaining a final design solution in the form of a topological drawing, we have formulated the following main requirements for solving the problem of ensuring protection against hardware backdoors in the IC design:

- filling the entire area of an IC digital core with standard library elements that are part of the

circuit suitable for functional testing (the main functional diagram of the design and the BISA scheme);
- fully controlling the integrity of the protection circuit using functional testing methods;
- minimising the effect of the protection circuit on the functional characteristics of the main device circuit;
- implementation of the protection circuit within the standard IC design route.

The above procedures are universal and can be effectively used in existing design centres to train and instruct staff in basic HT countermeasures during the design phase.

3.1 Implementing the self-testing circuit of an IC digital core

The idea of the BISA circuit implementation proposed in this work is to perform symmetric topology transformations in the obtained design solution of the IC topology that provides all the required functions in accordance with the technical specifications. Moreover, these symmetric topology transformations should not impair the circuit functionality, but allow the installation of additional BISA cells. This intermediate topological solution is then symmetrically displayed in a new drawing where empty spaces are filled with standard cells that form BISA circuits. The empty spaces should be filled with BISA cells in such a way that it is impossible to add a cell to any part of the topology that has the smallest size of all standard cells in the design library. The symmetry of the topology display ensures that all required functions of the main circuit are maintained in accordance with the specifications.

Assuming that the functionality of a HT circuit should not be less than a logic function with three inputs, the equivalent area occupied by a HT circuit in the topological drawing should not be less than the area of four inverters in this process. Therefore, the total area available for locating active semiconductor devices within the boundaries of a digital core should not exceed this value. Despite the fact that in the process of locating standard library elements within the boundaries of the digital core, the location pitch can be several times less than the width of the smallest inverter, we assume that an attacker can change the topological drawing in such a way as to combine unoccupied areas, as well as locate HT parts (up to one transistor) in unoccupied areas in different parts of the digital core.
Simply eliminating the free area by compacting the topological drawing of the main IC may lead to a fatal violation of design rules and design limitations:
- A violation of topological design rules, namely the inability to provide the required minimum distances between metallisation buses with a high-density location of input and output ports of library elements.
- A violation of electrical design rules, namely exceeding the permissible dissipated power per area unit in highly loaded parts of the circuit.
- A violation of design time limits, namely increasing parasitic capacities of metallisation buses with an increase of their location density.

Based on this, additional elements that are not part of the main circuit and do not have dynamic consumption during its operation should be used to fill the unused space, while the ratio of the area of these elements to the area of the required metal layout should be maximum.
Free space can be filled by various circuits, if only they can be controlled by non-invasive methods. In general, they can be divided into two types:
- digital ones (based on logic gates and memory circuits), during the control of which their logical function is checked;
- analog ones (based on passive and active elements), during the control of which their electrical and time parameters are checked.

At the same time, the protection circuit should not have a destructive effect on the main circuit. For example, it is permissible to increase the static current consumption of the circuit, but it is not permissible to reduce the maximum clock frequency of the main circuit or limit its any other functions.

This work uses GSCL (general standard cell library) as an example. It is a library of standard 150 nm process cells containing 42 elements of different widths that implement basic logical operations and auxiliary functions.

Names and types of the library cells are shown in Table 1.

To meet all requirements, we analysed the composition of the GSCL standard cell library to assess the effect of each of its functional elements (except for buffers and inverters) on the traceability of the main digital core circuit. Elements with the same functional purpose and different output stage power were also excluded from the analysis (only single-power elements were assessed). The assessment results are presented in Table 2.

Based on this assessment, a D-trigger named DFFX1 is selected as the main element of the integrity control circuit of the topological drawing of the IC digital core. This element is the simplest one, so its circuit cannot be optimised to reduce the area of its topological drawing, while the minimum number of involved ports without loss of control can be reduced to 3, which allows to achieve the best ratio of the controlled area and the spent tracing resources.

The proposed circuitry and algorithmic solutions were tested in this work using the example of the core of the RISC-V Steel Core RTL microprocessor whose code is open for free use.

Table 1. Elements of the GSCL standard cell library

Cell type	Cell name
Inversion	INVX8, INVX4, INVX2, INVX1
“AND”	AND2X1
“OR”	OR4X1, OR2X1
“AND-NO”	NAND4X1, NAND3X1, NAND2X2, NAND2X1
“OR-NO”	NOR4X1, NOR3X1, NOR2X1
“EXCLUSIVE OR”	XOR2X1
Two-level logic	OAI33X1, OAI22X1, OAI21X1, AOI22X1, AOI21X1
Multiplexer	MX2X1
D-triggers	DFFX1, DFFSRX1, SDFFSRX1
Clock buffers	CLKBUFX1, CLKBUFX3, CLKBUFX2
Three-state buffers	TBUFX1, TBUFX2, TBUFX4, TBUFX8
Adders	ADDHX1, ADDFX1
Buffers	BUFX1, BUFX3
Filling cells	FILL1, FILL2, FILL4

Table 2. Analysis of the GSCL library topological parameters

Element	Linear size, μm	Number of vertical tracks (pitch of 0.6 μm)	Number of ports	Number of effective vertical tracks
NAND2X1	3.3	5.5	3	2.5
NOR2X1	3.3	5.5	3	2.5
AOI21X1	3.96	6.6	4	2.6
AND2X1	3.96	6.6	3	3.6
OR2X1	3.96	6.6	3	3.6
NAND3X1	4.62	7.7	4	3.7
AOI22X1	5.28	8.8	5	3.8
NAND4X1	5.28	8.8	5	3.8
OAI21X1	5.28	8.8	4	4.8
XOR2X1	7.26	12.1	3	9.1
MX2X1	7.92	13.2	4	9.2
NOR3X1	7.92	13.2	4	9.2
OAI22X1	8.58	14.3	5	9.3
OR4X1	9.24	15.4	5	10.4
OAI33X1	11.22	18.7	7	11.7
TLATX1	11.22	18.7	4	14.7
NOR4X1	12.54	20.9	5	15.9
DFFX1	18.48	30.8	4	26.8
DFFSRX1	20.46	34.1	6	28.1
TLATSRX1	21.12	35.2	6	29.2
ADDFX1	24.42	40.7	5	35.7
DFFSRX1	35.64	59.4	8	51.4

This core contains three pipelining stages: extracting instructions, decoding instructions, executing instructions. The small number of pipelining stages eliminates the need to implement conflict resolution units and other complex microarchitectural solutions. Figure 1 presents a microarchitecture diagram of this processor at the level of register transmissions.

The main functional units of this processor are as follows:

- A decoder that decrypts instructions and generates signals to monitor the processor data path.
- An arithmetic-logic unit (ALU) that implements 10 logical and arithmetic operations on two 32-bit operands.
- Register files (IRF, CSR) that contain 32 general-purpose registers and support writing and reading operations.
- A branch unit that predicts when to perform the operation of moving to another address in the executed programme.
- A load unit that is required to load data into a register file.
- A store unit that controls signals of the memory interface.
- An immediate generator that expands immediate operands to 32 bits if necessary.
- A special machine control that controls the processor instruction counter.

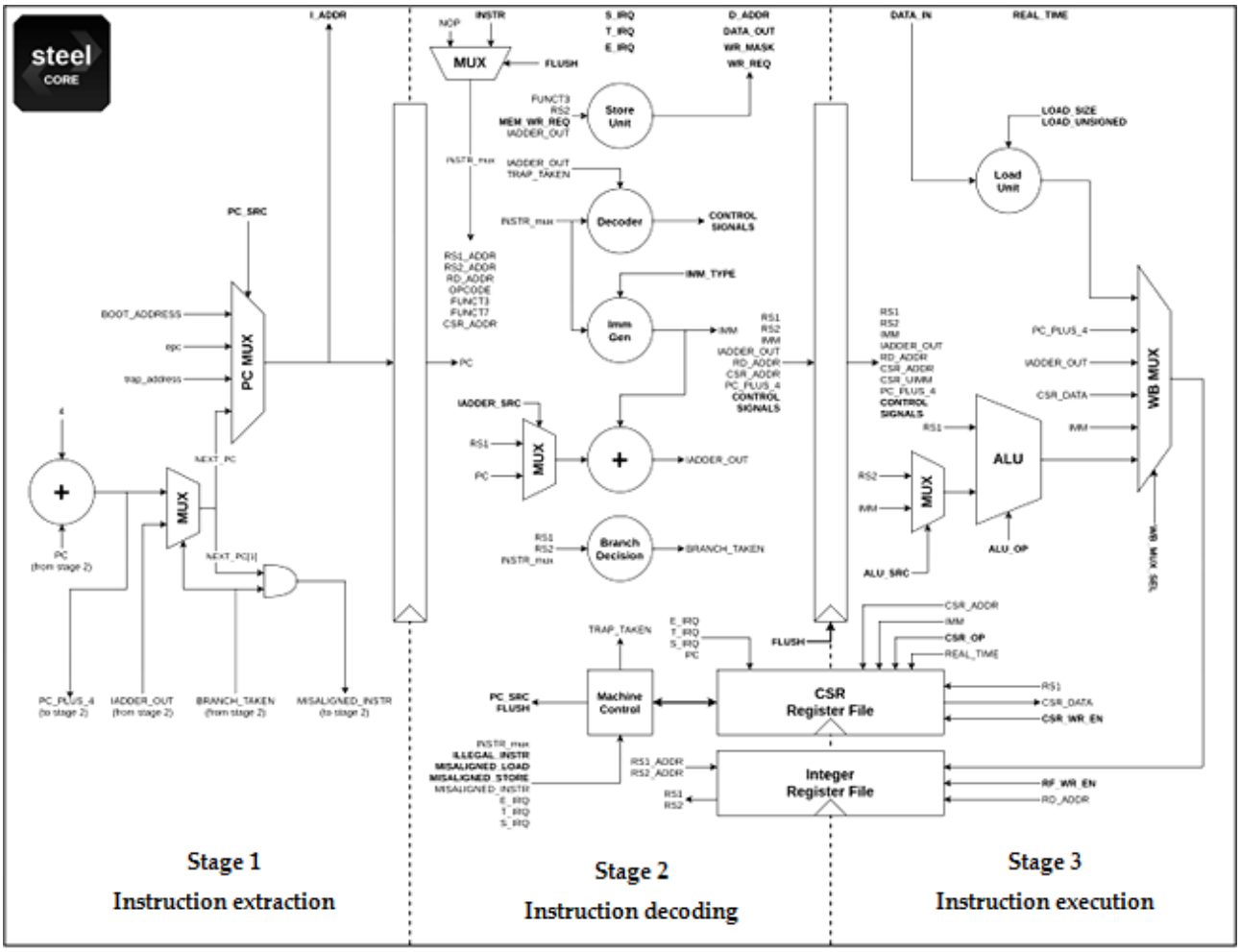

Figure 1. The RISC-V Steel microprocessor core at the level of register transmissions

In general, the design route is shown in Figure 6. Parallel branches of the circuit show that the design of analog and digital parts can be carried out parallelly and independently.

The entire design route can be nominally divided into 5 stages: architecture development, development of non-standard library elements (analog components and SF units), logical synthesis (digital components), chip planning and physical synthesis (combining all parts of the design), mixed modelling and verification. At the same time, the introduction of BISA elements is a separate stage, the role and place of which in the design route are given in Figure 2.

At this stage, DEF (design exchange format) files that contain information about cell coordinates, chip sizes, power lines, etc., and LEF (library exchange format) files that contain data about the size of standard cells, the location of their ports and the metallisation paths within a cell should already be retrieved from the design database. These files are required to implement the BISA filling cell location algorithm.

Based on D-triggers, when they are switched on sequentially, a shift register circuit can be implemented (trigger chains, Figure 3), which allows to monitor the presence of all its elements (links), for which it is enough to use only three additional ports (Si, So, Clock).

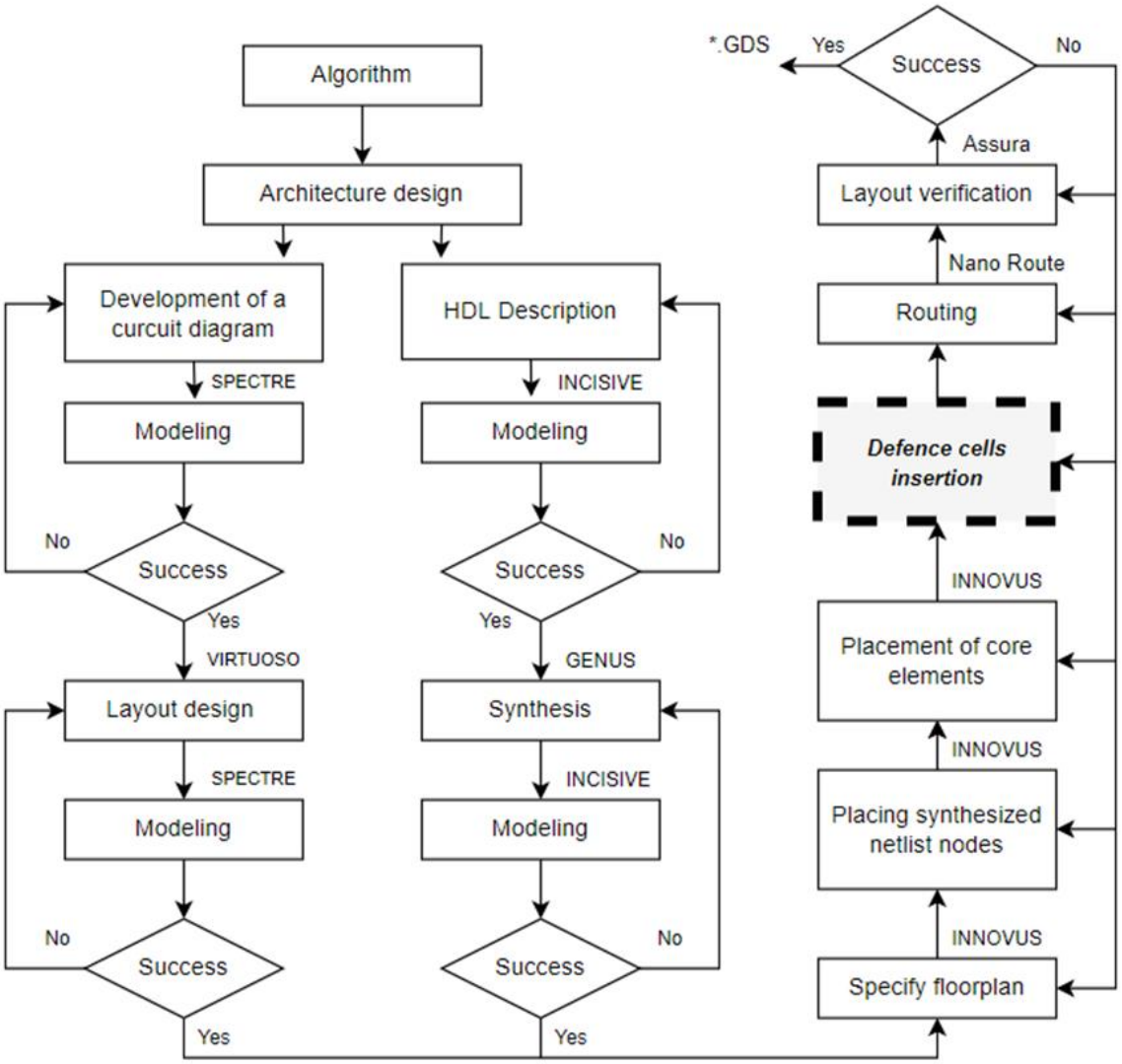

Figure 2. The design route of the topology of trusted ICs

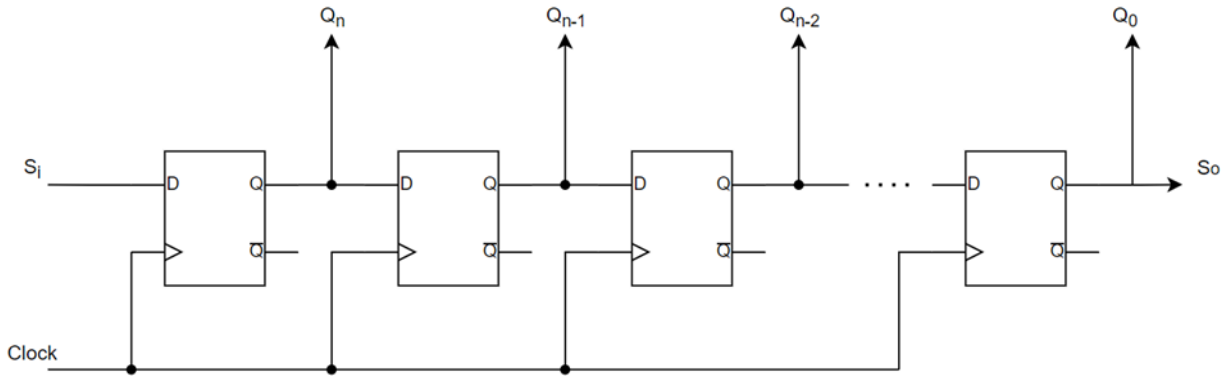

Figure 3. The diagram of a shift register

By replacing all the free areas of the IC digital core with D-trigger cells and then combining them into a shift register of any length, the location density of standard cells approaching 100% can be achieved, while the integrity control of the chain itself, which is performed once, can be carried out in a number of cycles equal to twice the number of triggers in the chain. By supplying an arbitrary, unknown binary sequence to the input of the shift register and reading it from the output of the shift register through the number of cycles equal to that of triggers in the circuit, the integrity of the proposed

integrity control circuit of the topological drawing of the IC digital core can be precisely estimated.

One of the disadvantages of the previous methods is the inability to achieve 100% use of the free circuit space, since possible voids may be smaller or be not a multiple of the dimensions of the test circuit elements inserted into them. The remaining voids can be smaller in size than the NAND gate. But the total area of all such voids may be critical. And when a HT is introduced, this space can be used, for example, by combining individual voids into one due to small movements of the main circuit elements (displays symmetrical with respect to the main functions of the circuit). Insignificant movements (comparable to the size of the simplest gate), for the most part, have a minimal effect on the main circuit, since the average location density of the elements does not change. Therefore, if an additional location optimisation cycle is performed prior to the introduction of the protection circuit, which provides voids equal to or being a multiple of the dimensions of the filling elements of the protection circuit, 100% use of the free area can be achieved. After such optimisation, re-checking the time characteristics of the main circuit allows to verify that its parameters meet the requirements. If the requirements are not met, several optimisation stages can be performed to achieve the required results. After the voids are compensated through optimisation, the previously described integrity control circuit can be implemented.

To ensure 100% filling of the free space inside the digital part of the design, it is necessary to set the dimensions of the voids to multiple sizes of the filling elements. Initially, the dimensions of the voids are arbitrary and are chosen by means of locating the main circuit, based on the requirements of the optimal design tracing. Setting the voids to multiple dimensions results in the movement of the main circuit elements, while their significant movement may negatively affect the parameters of the main circuit. Therefore, setting the voids to multiple dimensions should be carried out while maintaining the average density of the design. To construct a clock tree for the shift register, the circuit should have room for clock signal buffers. The location of these buffers should be strictly consistent with the direction of signal propagation (towards the data flow) and meet a number of requirements for the maximum distance between the buffers and the maximum number of the triggers per buffer. Since the operation is carried out with the digital circuit elements, only the width of the elements is taken into account, since the height of all elements is the same. When levelling the voids, the following actions can be carried out:

- decreasing the size to a multiple size
- increasing the size to a multiple size
- combining adjacent voids up to a multiple size
- creating voids of specified size in places required for setting clock buffers

The cell location programme is implemented in the Python programming language and contains three main stages: reading data from .def, .v and .lef files; an algorithm for optimising and locating cells; recording data to new files with the same extensions.

An example of text data contained in the LEF and DEF files is shown in Figure 4.

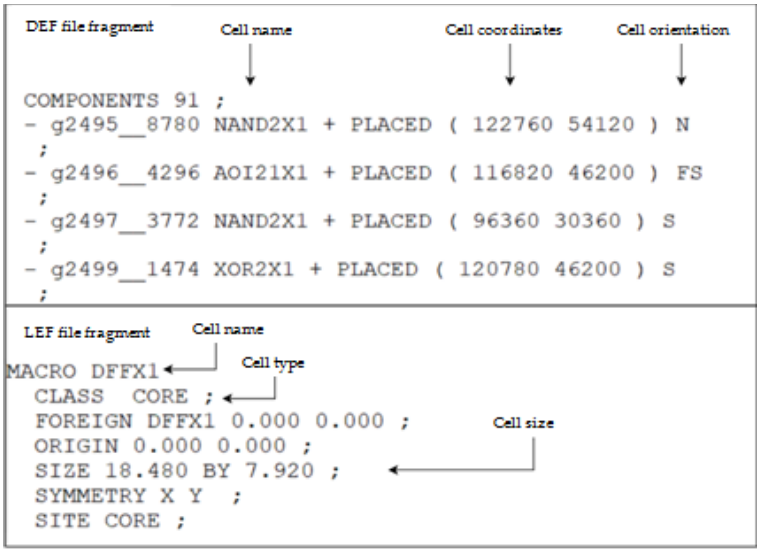

Figure 4. The fragments of LEF and DEF files describing the parameters and coordinates of the location of the standard cells used in the design

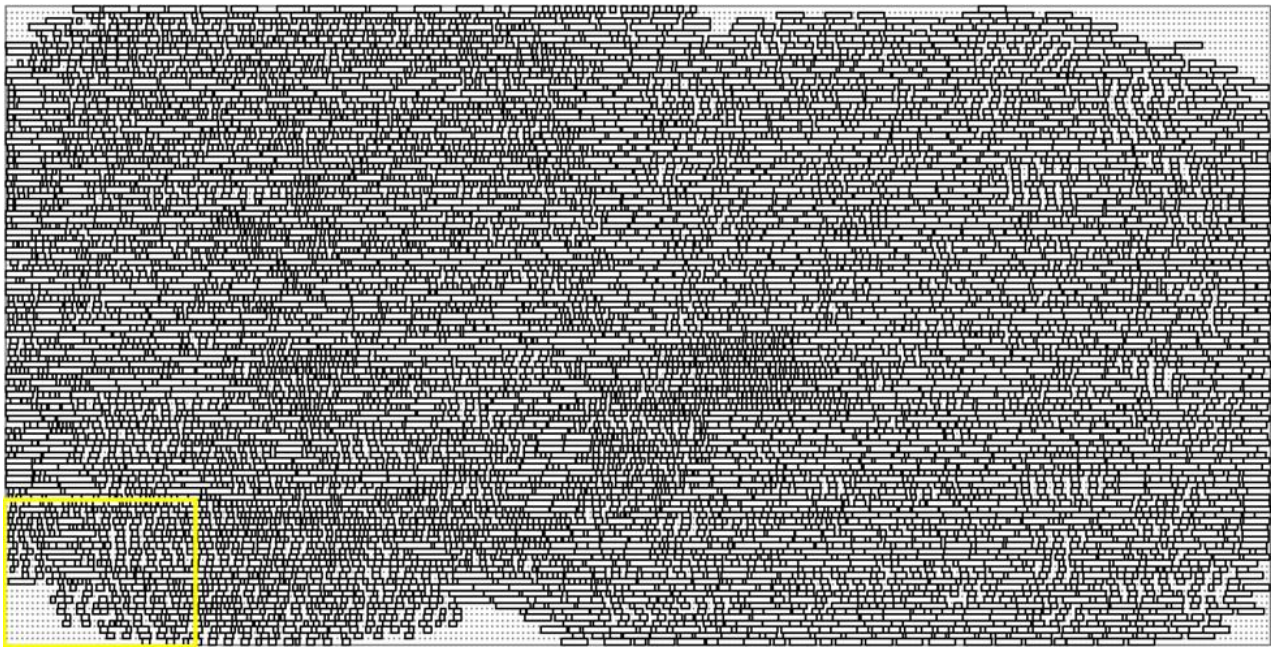

Figure 5. The IC core topology drawing after converting DEF file data

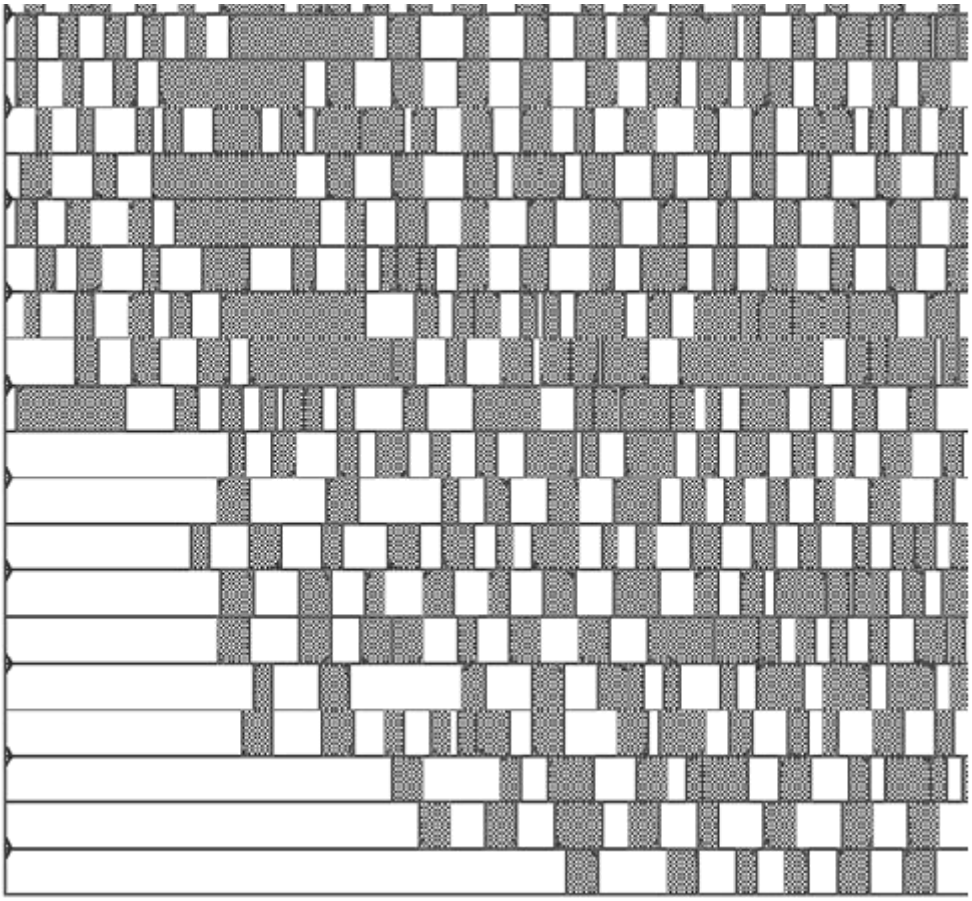

Figure 6. The fragment of the IC digital core after locating elements of the synthesised netlist (the initial back end)

All stages of the unit design route, including the location of specialised elements of the digital core, are carried out in the standard route using standard tools. At the preparatory stage, a special Python programme converted data from a DEF file format into a cell topology drawing. The result of the conversion programme is shown in Figure 5.

An enlarged fragment of the digital core drawing highlighted in the lower left corner of Figure 5 is presented in Figure 6.

This topology is initial for subsequent symmetric transformation in order to introduce BISA cells, which is performed in two stages: 1) analysis of free areas and symmetric shift of standard cells, 2) location of BISA cells. This optimisation algorithm can be implemented as follows. Since all logical elements of the digital core are located sequentially in strings, optimisation should also be carried out string by string. At the first step, the string start and end coordinates should be obtained. Next, the centre of the string is determined, relative to which the cell shift algorithm will be implemented. After marking all reference points, the optimisation of cell space in strings is carried out by shifting them and filling the freed space with filling elements, in our case with D-triggers. In this case, it is necessary to take into account the orientation of the filling cell in space, since the orientation of the power lines alternates in each string.

A block diagram of the location algorithm is shown in Figure 7.

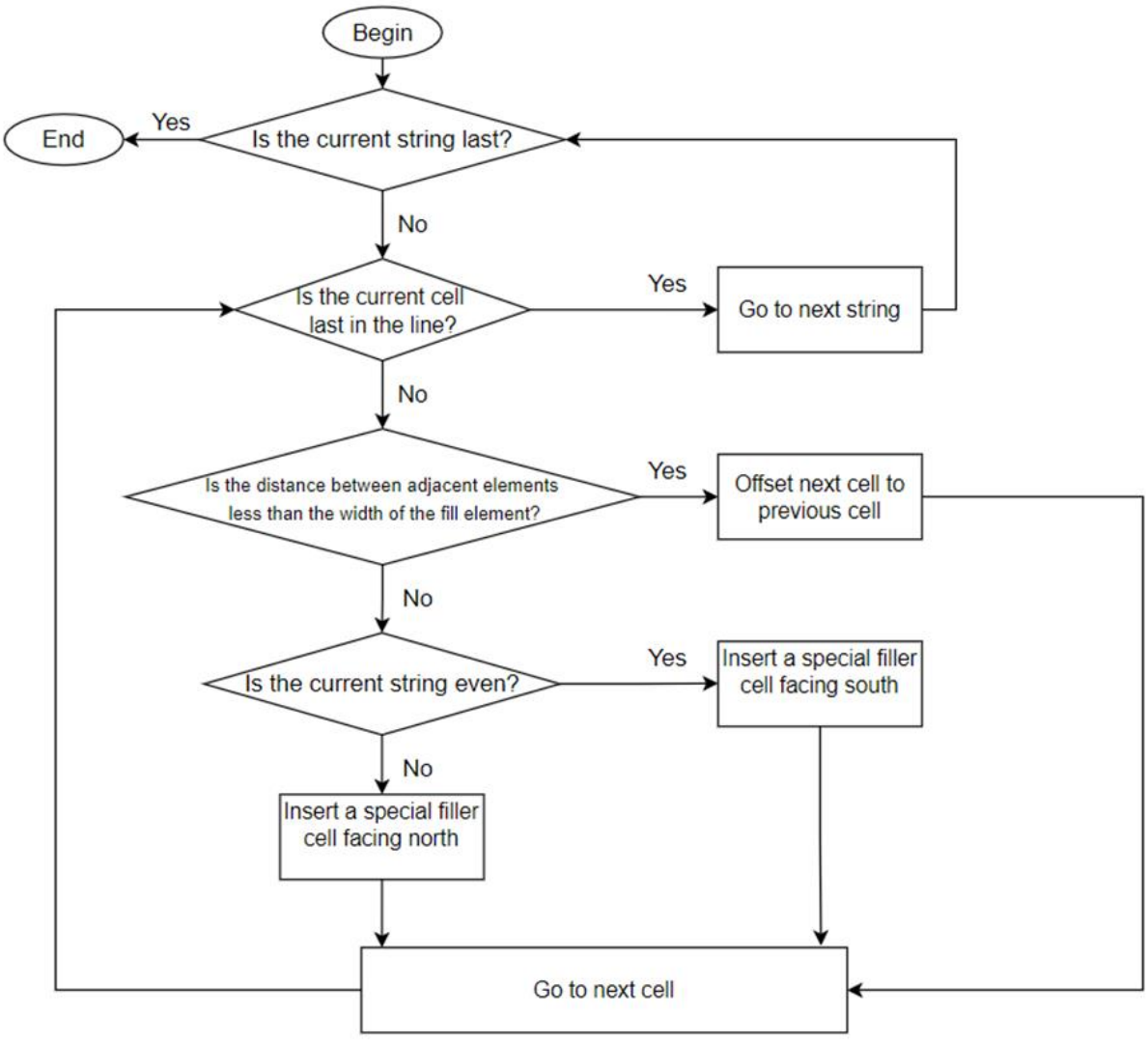

Figure 7. Block diagram of the location algorithm for filling cells

The programme that implements this algorithm is also written in Python. An example that illustrates the operation of the standard cell symmetric shift programme is shown in Figure 8. The filling cell is marked as DFFX1. At the first step, the distance between adjacent elements is checked, including the line start coordinate. If the D_0 distance is less than the width of the filling cell, the right cell is shifted to the left one; in this case, the cell with coordinates (X_1,Y_0) to that with coordinates (X_0,Y_0). At the next step, the distance between the next two elements is checked; if it is larger than the width of the filling element, it is located in this place, otherwise a verification similar to the first step is carried out. At the last step, the filling cell extends the array of the string, then the same conditions are checked on the string as in the previous two steps.

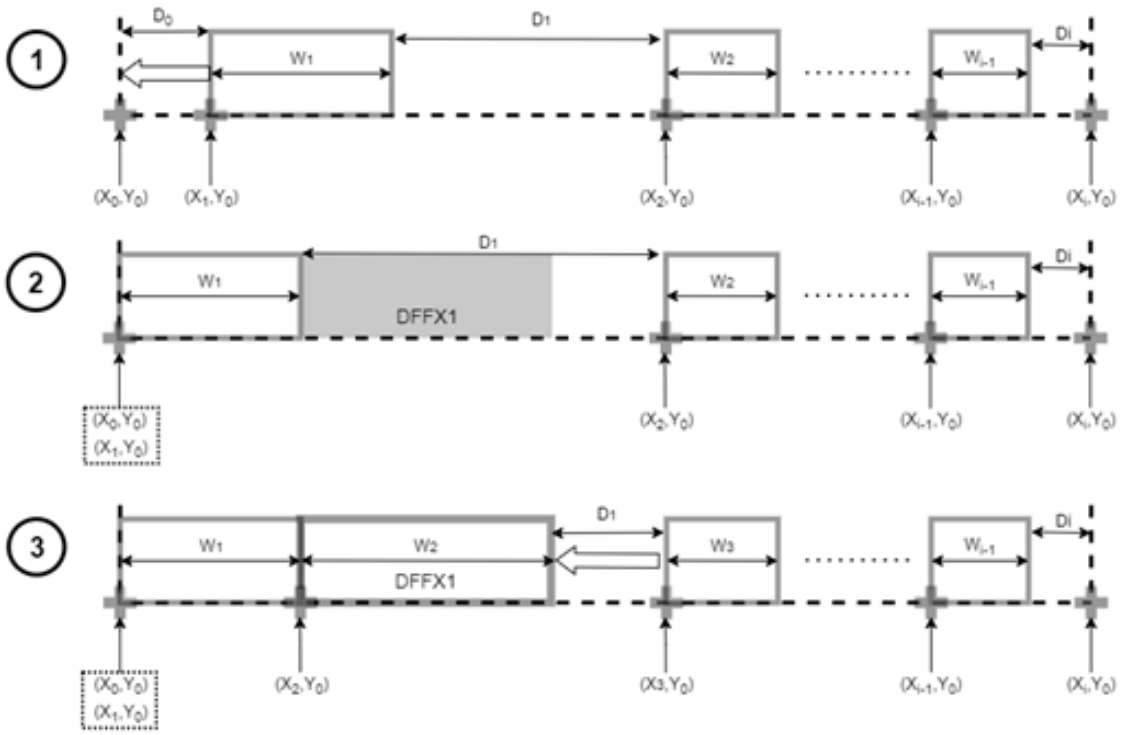

Figure 8. Visualisation of the filling cell location algorithm

At the stage of introducing protection elements for the IC digital core into the design, the following sequence of actions is performed:

1) Uploading information on the location of library cells of standard digital elements of the synthesised netlist and specialised elements.
2) Optimising the location of library cells of standard digital elements of the synthesised netlist, taking into account the requirements for free areas.
3) Locating the triggers of the specialised scan chain in the voids of the optimised digital core.
4) Finding the best way to connect the triggers of the specialised scan chain.
5) Downloading information on the location of library cells of standard digital elements of the synthesised netlist and triggers of the specialised scan chain.
6) Downloading information on the interconnections of triggers of the specialised scan chain in the format of a synthesised netlist for subsequent tracing.

The algorithm for optimising the location of library cells consists in shifting elements within strings so that, after displacement, the voids or free areas between adjacent elements have certain width.

Elements in a string can move in an arbitrary order to the left or right, but the elements cannot swap places.

After optimising all free areas in a string, one can proceed to the next string and so on until the strings are complete. After all strings have been optimised, new element coordinates are recorded, their dimensions and the list of elements are downloaded in an input format. At this stage, the percentage of the occupied core area is 69.99%.

As a result of changing the input data, only the coordinates of the logical cells change, the remaining input data do not change, namely, the size of the cells, their names and the dimensions of the workspace should remain in their original form. As a result, the digital core takes the form shown in Figure 9.

As a result of locating the triggers of the specialised scan chain in the free areas of the design strings, the location of library cells of standard digital elements of the synthesised netlist and specialised elements is optimised so that the voids between the elements in the string are multiple of the size of the trigger of the specialised scanning chain.

After replacing all voids with the cells of the trigger of the specialised scan chain, new element coordinates are recorded, as well as their dimensions and the supplemented list of elements are downloaded in an input format.

As a result, the digital core takes the form shown in Figures 10 and 11 (a highlighted fragment).

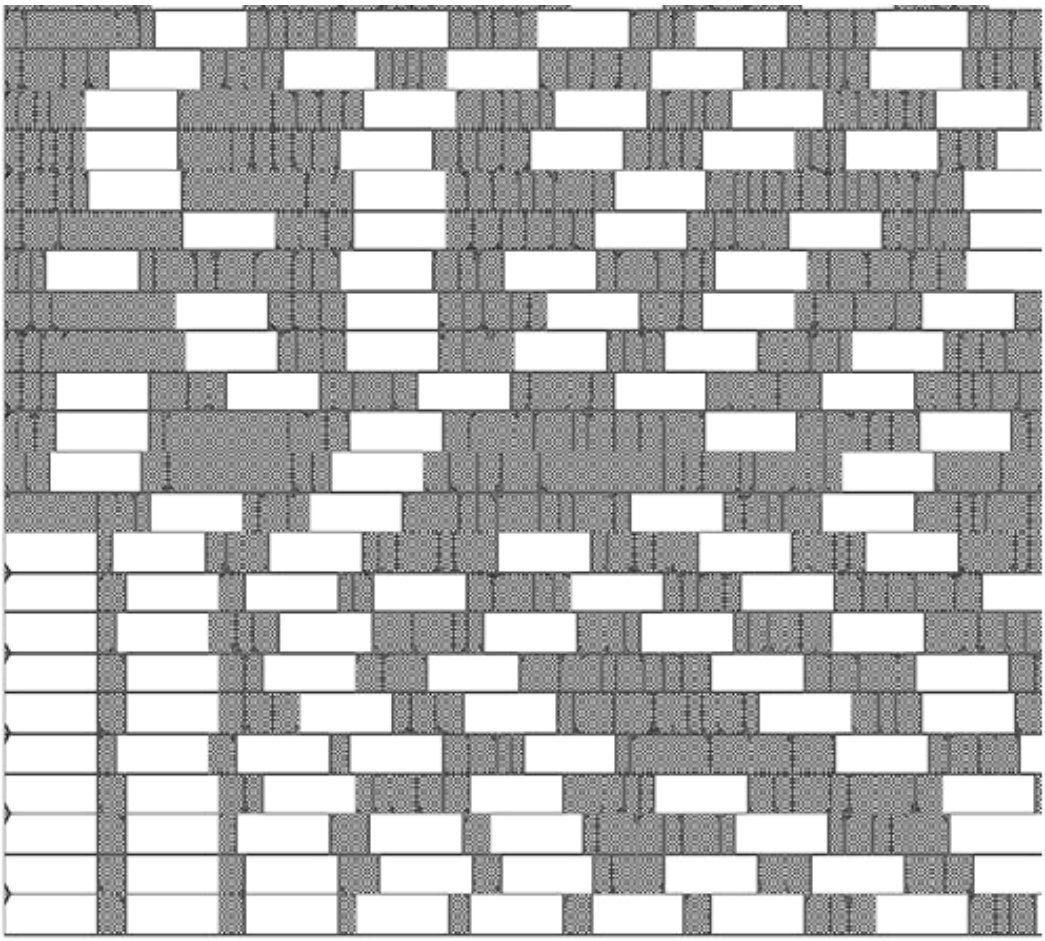

Figure 9. A fragment of the IC digital core after the stage of locating the elements of the synthesised netlist

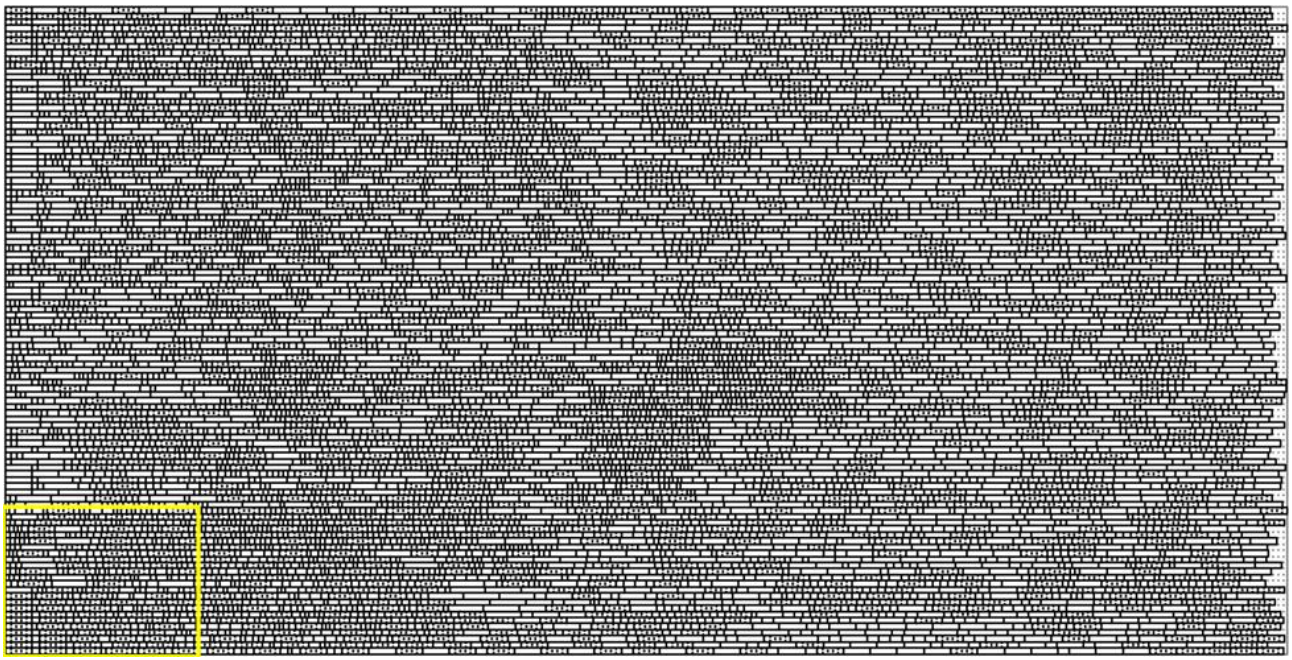

Figure 10. The IC digital core with located protection elements

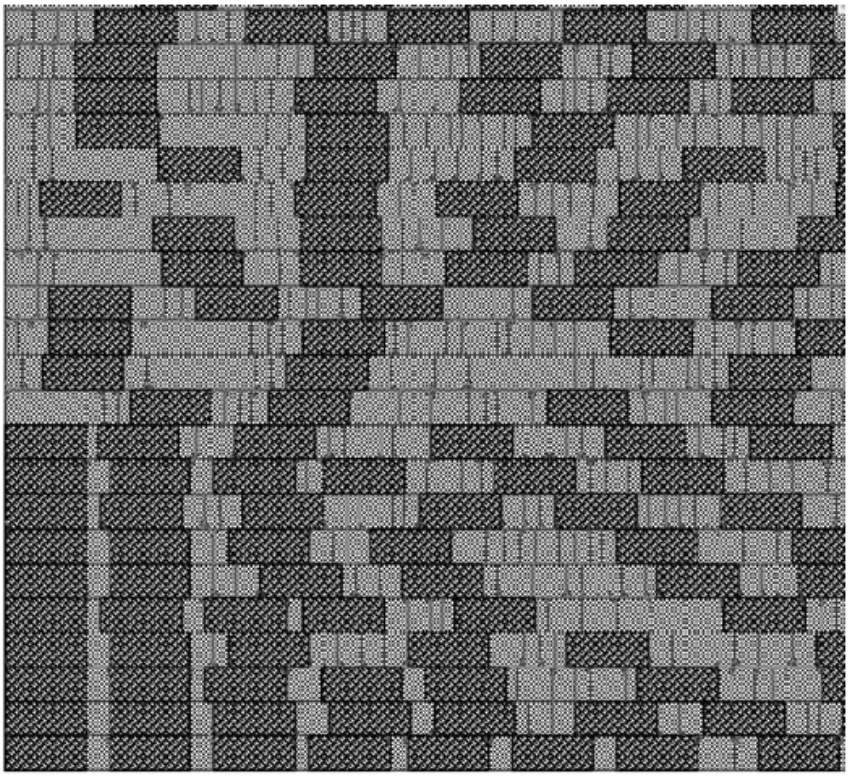

Figure 11. A fragment of the IC digital core with located protection elements (inserted standard elements are highlighted in dark grey)

In this case, the percentage of the occupied chip area is 98.95%.

At the next design stage, the process of sequentially forming logical interconnections between the nearest triggers of the specialised scan chain takes place, as a result of which the input of each subsequent trigger is connected to the output of the previous one independently of the location string. The result is a netlist file for the triggers of the spe-cialised scan chain in the Verilog hardware description language. The resulting netlist is included in the main netlist file of the design, and then we return to the stage of tracing the standard design route.

4. CONCLUSION

Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Research has been carried out on software methods for detecting and preventing the introduction of hardware Trojans into the topology of integrated circuits at the stage of their design. An additional recommendation for typical design procedures for obtaining GDSII files, which significantly complicates the introduction of hardware Trojans into the topology, is to facilitate detection methods by including special self-testing structures in the project, formed after performing symmetrical transformations of the original topology and providing effective verification of integrated circuits at the testing stage after manufacturing. A method is proposed for introducing BISA self-testing cells into the project, which is based on the analysis of the results of synthesis at the level of description of the list of circuits, its subsequent program transformation into a topology drawing, symmetrical shift of standard cells and filling the voids formed in this way with special self-testing cells. The result is a netlist file for custom scan chain flip-flops in the Verilog hardware description language. The resulting netlist is included in the project's main netlist file, after which it returns to the tracing phase of the standard design route.

The proposed method makes it possible to build a very simple and efficient protection scheme based on the operation of symmetrical transformation of the topology drawing and shift of standard design elements.

Advantages of the method:

- the simplicity of the scheme and its strict determinism, when controlling the scheme, you can accurately determine the number of protection triggers and, if they are absent, you can talk about possible changes to the scheme
- the minimum necessary amount of required additional tracing resources.

The method is simple to implement at the topology design stage and can be recommended for use in design centers for machine learning of computer-aided design systems.

Acknowledgement: This study has been supported by the Ministry of Science and Higher Education of the Russian Federation (Project No. FSFZ-0706-2020-0022).

References:

Bhunia, S. , & Tehranipoor, M. M. (2018). *The Hardware Trojan War*. Cham, Springer International Publishing AG. doi: 10.1007/978-3-319-68511-3_2

Chaduvula, S. Ch., Dachowicz, A., Atallah, M. J., & Panchal, J. H. (2018). Security in Cyber-Enabled Design and Manufacturing: A Survey. *Journal of Computing and Information Science in Engineering, 18*(4), 040802. doi:10.1115/1.4040341

Chakraborty, R. S., & Bhunia, S. (2009). Security against Hardware Trojan through a Novel Application of Design Obfuscation. *Proceedings of the 2009 International Conference on Computer-Aided Design (ICCAD)* (pp. 113-116). New York, NY, United States, Association for Computing Machinery. DOI: 10.1145/1687399.1687424

Chakraborty, R. S. , Narasimhan, S., & Bhunia, S. (2009). Hardware Trojan: Threats and emerging solutions. *Proceedings of the IEEE International High Level Design Validation and Test Workshop* (pp. 166-171). San Francisco, CA, US, IEEE. doi: 10.1109/HLDVT.2009.5340158

Cui, X., Koopahi, E., Wu, K., & Karri, R. (2018). Hardware Trojan Detection Using the Order of Path Delay.

J. Emerg. Technol. Comput. Syst., 14(3), 33. https://doi.org/10.1145/3229050

Fievrea, A. M. P., Al-Aakhir, Rogersb, A., & Bhansalia, Sh. (2015). Integrated circuit security: an overview. *Journal of Institute of Smart Structures and Systems (ISSS), 4*(1), 18-37.

Herder, C., Yu, M., Koushanfar, F., & Devadas, S. (2014). Physical Unclonable Functions and Applications: A Tutorial. *Proceedings of the IEEE, 102*(8), 1126-1141. doi: 10.1109/JPROC.2014.2320516.

Huang, Z., & Wang, Q. (2020). Enhancing Architecture-level Security of SoC Designs via the Distributed Security IPs Deployment Methodology. *Journal of information science and engineering, 36*, 387-421.

Jin, Y., & Makris, Y. (2008). Hardware Trojan detection using path delay fingerprint. *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust* (pp. 51-57). Anaheim, CA, IEEE. doi: 10.1109/HST.2008.4559049

Khairallah, M., Sadhukhan, R., Samanta, R., Breier, J., Bhasin, Sh., Chakraborty, R. S., Chattopadhyay, A., & Mukhopadhyay, D. (2018). DFARPA: Differential Fault Attack Resistant Physical Design Automation Design. *Proceedings of the Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1171-1174). Dresden, Germany, IEEE. DOI: 10.23919/DATE.2018.8342190.

Knechtel, J. et al. (2020). "Towards Secure Composition of Integrated Circuits and Electronic Systems: On the Role of EDA," 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 508-513, doi: 10.23919/DATE48585.2020.9116483.

Kulkarni, A., Pino, Y., & Mohsenin, T. (2016). SVM-based real-time hardware Trojan detection for many-core platform. *Proceedings of the 17th International Symposium on Quality Electronic Design (ISQED)* (pp. 362-367). Santa Clara, CA, US, IEEE. DOI: 10.1109/ISQED.2016.7479228

Kuznetsov, E., & Saurov, A. (2016a). Hardware Trojans. Part 1: new threats to cyber security. *Nanoindustry, 7*, 17-26.

Kuznetsov, E. & Saurov, A. (2016b). Hardware Trojans. Part2: examples of implementation, methods of insertion and activation. *Nanoindustry 8*(70), 12-20.

Kuznetsov, E. & Saurov, A. (2017a). Hardware Trojans. Part 3: methods for prevention and detection. *Nanoindustry 1*(71), 30-40.

Kuznetsov, E., & Saurov, A. (2017b). Hardware Trojans. Part 4: software and hardware countermeasures. *Nanoindustry, 2*(72), 42-56.

Pevtsov, E. Ph., Demenkova, T. A., & Shnyakin, A. A. (2019). Design for testability of integrated circuits and project protection difficulties. *Russian Technological Journal, 7*, 4, 60-70. https://doi.org/10.32362/2500-316X-2019-7-4-60-70

Potkonjak, M. (2010). Synthesis of trustable ICs using untrusted CAD tools. *Proceedings DAC '10: Proceedings of the 47th Design Automation Conference* (pp. 633-634). New York, NY, United States, Association for Computing Machinery. doi: 10.1145/1837274.1837435

Moein, S., Gebali, F., Gulliver, T. A., & Alkandari, A. (2017). Hardware Trojan Identification and Detection. *International Journal on Cryptography and Information Security 7*(2), 1-20. doi:10.5121/ijcis.2017.7201.

Shekarian, S. M. H., Zamani, M. S., & Alami, S. (2013). Neutralizing a design-for-hardware-trust technique. *Proceedings of the 17th CSI International Symposium on Computer Architecture & Digital Systems (CADS 2013)* (pp. 73-78). Tehran, Iran, IEEE. doi: 10.1109/CADS.2013.6714240

Shi, Q., Tehranipoor, M. M., & Forte, D. (2019). Obfuscated Built-In Self-Authentication with Secure and Efficient Wire-Lifting. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 38*(11), 1981-1994, doi: 10.1109/TCAD.2018.2877012

Vaikuntapu, R., Bhargava, L., & Sahula, V. (2016). Golden IC free methodology for hardware Trojan detection using symmetric path delays. *Proceedings of the 20th International Symposium on VLSI Design and Test (VDAT)* (pp. 1-2). Guwahati, India, IEEE. doi: 10.1109/ISVDAT.2016.8064895

Xiao, K., & Tehranipoor, M. (2013). BISA: Built-In Self-Authentication for Preventing Hardware Trojan Insertion. *Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (pp. 45-50). Austin, TX, USA, IEEE. doi: 10.1109/HST.2013.6581564

Xue, H., & Ren, S. (2018). Hardware Trojan detection by timing measurement: Theory and implementation. *Microelectronics Journal, 77*, 16-25.

Yoshimizu, N. (2014). Hardware trojan detection by symmetry breaking in path delays. *Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (pp. 107-111). Arlington, VA, USA, IEEE. doi: 10.1109/HST.2014.6855579

Yuan, Sh. W., Li, L., Yang, J., He, Y., Zhou, W. T., & Li, J. (2021). Real-time detection of hardware trojan attacks on General-Purpose Registers in a RISC-V processor. *IEICE Electronics Express, 18*(10), 1-3. doi: 10.1587/elex.18.20210098

Evgeniy Pevtsov
MIREA—Russian Technological University (RTU MIREA),
Moscow,
Russian Federation
pevtsov@mirea.ru
ORCID 0000-0001-6264-1231

Tatiana Demenkova
MIREA—Russian Technological University (RTU MIREA),
Moscow,
Russian Federation
demenkova@mirea.ru
ORCID 0000-0003-3519-6683

Alexander Sigov
MIREA—Russian Technological University (RTU MIREA),
Moscow,
Russian Federation
sigov@mirea.ru
ORCID 0000-0002-1216-3339

Alexander Shnyakin
MIREA—Russian Technological University (RTU MIREA),
Moscow,
Russian Federation
shniks@mail.ru
ORCID 0000-0002-4716-2547

Semyon Moskolenko
MIREA—Russian Technological University (RTU MIREA),
Moscow,
Russian Federation
sabblesteaks@yandex.ru
ORCID 0000-0003-4518-5590