

ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ, ԳԻՏՈՒԹՅԱՆ,  
ՄՇԱԿՈՒՅԹԻ ԵՎ ՍՊՈՐՏԻ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

## Բարայան Արմեն Վրեժի

ՆԱՆՈՉԱՓԱԿԱՆ ԲՅՈՒՐԵՂԻ ՎՐԱ ՏԵՂԱԳՐՎԱԾ  
ՀԱՄԱԿԱՐԳԵՐԻ ԹԵՍՏԱՎՈՐՄԱՆ ԼՈՒԾՈՒՄՆԵՐԻ ՈՐԱԿԻ  
ՀԱՎԱՍՏՄԱՆ ԱՐԳՅՈՒՆԱՎԵՏՈՒԹՅՈՒՆԸ ԲԱՐՁՐԱՑՆՈՂ  
ԳՈՐԾԻՔՆԵՐԻ ՄՇԱԿՈՒՄԸ

Ե.27.01 «Էլեկտրոնիկա, միկրո և նանոէլեկտրոնիկա» մասնագիտությամբ  
տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման  
ատենախոսության

### ՍԵՂՄԱԳԻՐ

Երևան 2024

THE MINISTRY OF EDUCATION, SCIENCE, CULTURE AND SPORTS OF  
THE REPUBLIC OF ARMENIA

NATIONAL POLYTECHNIC UNIVERSITY OF ARMENIA

## Babayan Armen Vrezh

DEVELOPMENT OF TOOLS INCREASING QUALITY ASSURANCE  
EFFICIENCY FOR BUILT-IN SELF TEST SOLUTIONS IN NANOSCALE  
SOC

### SYNOPSIS

The dissertation is submitted in fulfillment of the  
requirements for the scientific degree of Candidate of Technical Sciences in  
Specialty 05.27.01 “Electronics, micro and nanoelectronics”

Yerevan 2024

Ատենախոսության թեման հաստատվել է Հայաստանի ազգային պոլիտեխնիկական համալսարանում (ՀԱՊՀ):

Գիտական ղեկավար՝ ֆ. մ. գ. դ. Սամվել Կիմի Շուքուրյան

Պաշտոնական ընդդիմախոսներ՝  
ու. գ. դ. Օլեգ Հարությունի Պետրոսյան  
տ. գ. թ. Արման Ստեփանի Տրդատյան

Առաջատար կազմակերպություն՝ Երևանի կապի միջոցների  
գիտահետազոտական ինստիտուտ

Ատենախոսության պաշտպանությունը կայանալու է 2024թ. հոկտեմբերի 11-ին, ժամը 14<sup>00</sup>-ին, ՀԱՊՀ-ում գործող 046 մասնագիտական խորհրդի նիստում (հասցեն՝ 0009, Երևան, Տերյան փ., 105, 17 մասնաշենք):

Ատենախոսությանը կարելի է ծանոթանալ ՀԱՊՀ-ի գրադարանում:

Սեղմագիրն առաքված է 2024 թ սեպտեմբերի 9-ին:

046 Մասնագիտական խորհրդի  
գիտական քարտուղար, տ. գ. թ.



Բենիամին Ֆելիքսի Բադալյան

---

---

The topic of the dissertation was approved at the National Polytechnic University of Armenia (NPUA)

Scientific supervisor: Dr. Phys.-Math. Sci. Samvel Kim Shoukourian

Official reviewers: Dr. Tech. Sci. Oleg Harutyun Petrosyan  
Cand. Tech. Sci. Arman Stepan Trdatyan

Leading reviewer organization: Yerevan Telecommunication Research  
Institute

The defense will be held at the meeting of the specialized council 046, in the National Polytechnic University of Armenia on October 11<sup>th</sup>, 2024, at 14<sup>00</sup>. Address: Teryan st. 105, building 17, 0009, Yerevan.

The thesis is available at the NPUA library.

The synopsis in English was distributed on 9<sup>th</sup> of September 2024.

Scientific Secretary of the  
Specialized Council 046,  
Candidate of Technical Sciences



Beniamin Felix Badalyan

**Relevance of the topic.** Increasing of the manufacturing yield continues to be one of the most important problems in semiconductor production. Due to possible power, voltage and temperature (PVT) deviations during production, a considerable part of integrated circuits (ICs) located on a semiconductor wafer may have defects<sup>1</sup>, which disrupt the performance of these ICs. Most of these defects in wafers are usually relating to memory devices, which are typically occupying over 70%<sup>2</sup> of the design and are most prone to defects causing failures. Among those defects there are many specific defects which can be sensitized and, thus, can be detected, then localized and finally repaired. This is the reason why embedded built-in test and repair (BIST) solutions for memories (MBIST) which usually occupy a negligible area are inserted in a given design to provide high test coverage and to increase the overall yield<sup>3</sup>.

There are various MBIST solutions which have different architectures and features when seeking a BIST solution for a given system on a chip (SoC)<sup>4</sup>. To make these solutions reusable, MBIST vendors have defined some description standards and have provided automated flows which generate SoC specific register transfer level (RTL) representation/instance of the given MBIST through RTL compilers and then insert it into the SoC basing on the mentioned standards.

On the other hand, this automation of the MBIST development might bring sometimes to drawbacks leading to a non-working RTL instance in the given SoC environment. These drawbacks, particularly, include the case when the SoC design is not synchronized with the mentioned standards and limitations for MBIST functionality. Due to that some SoC vendors consider undesirable inner modifications of their design for MBIST implementation and insertion. Instead, they suggest using of a shared interface<sup>5</sup> within the chip, which is intended to provide an access to the memories in the functional mode and is called a multi-memory bus (MMB). In other words, they suggest using of the MMB for the test mode too.

To separate consideration of the memory BIST for the mentioned case the following abbreviation will be used further: MMB-BIST. For MMB-BIST a description of the interface as well as descriptions of memories connected to the interface should be provided. Meantime, any validation of a generated MMB-BIST RTL instance including its functionality validation, requires generation of multiple instances from mentioned descriptions of the shared interface and connected to the interface memories and their further execution with the considered MMB-BIST instance. The number of possible combinations for mentioned generated instances is much bigger than the number of generated instances for a given memory, and, correspondingly, their exhaustive runs take unacceptable time. This brings to a necessity of an in-depth analysis followed by some

---

<sup>1</sup> G. Harutyunyan, S. Shoukourian, Y. Zorian, "Fault Awareness for Memory BIST Architecture Shaped by Multidimensional Prediction Mechanism", IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 38(3): 562-575 (2019).

<sup>2</sup> 2015 International Technology Roadmap for Semiconductors (ITRS).

<sup>3</sup> Y. Zorian, S. Shoukourian, "Embedded-memory test and repair: infrastructure IP for SoC yield", IEEE Design & Test of Computers, 2003, vol. 20, no. 3, pp. 58-66.

<sup>4</sup> G. Harutyunyan, S. Martirosyan, S. Shoukourian, Y. Zorian, "Memory Physical Aware Multi-Level Fault Diagnosis Flow", IEEE Trans. Emerg. Top. Comput. 8(3): 700-711 (2020)

<sup>5</sup> T. McLaurin, R. Knoth, "The Challenges of Implementing an MBIST Interface: A Practical Application", IEEE International Test Conference (ITC), 2019, pp. 1-6.

actions directed to an essential increase of the MMB-BIST validation efficiency.

The thesis is dedicated to the development and implementation of an approach and methods which increase MMB-BIST validation efficiency.

**Subject of study.** Drawbacks for MMB-BIST compiler validation caused by exhaustive runs during the validation.

**The aim of the thesis.** Development of efficient and justified validation methods for MMB-BIST Engines and their implementation in a tool to be used in testing and quality assurance (QA) of MMB-BIST.

**Methods of study.** Methods for MBIST test and validation flow including modeling and automated validation approaches, decision-making solutions as well as principles of learning methodologies applied to validation flows.

**Scientific novelty.**

- A validation model for MMB-BIST is suggested which describes the main features of a SoCs which have memories accessible and testable through shared interfaces. The model can be the basis for a MMB-BIST validation environment implementation.
- A learning based efficient validation methodology for MMB-BIST configurations is built, justified, and implemented. The corresponding signed certificate of implementation at “Synopsys-Armenia” CJSC is added in the Appendix 1. Particularly:
  - the methodology is flexible and extensible due to parametrization and tuning to given requirements for validation flow which leads to user-friendly generation of a specific validation method according to specific requirements for generation reflected in parameter values;
  - a dynamically built hierarchy of MMB-BIST parameter priorities is used for the validated/tested features based on the knowledge of MMB-BIST internal structure;
  - an efficient exhaustion method is used to validate MBIST functioning within a fixed time slot basing on the built hierarchy.
  - experiments on application of the developed methodology have shown that the number test-cases in average is reduced by 30% in comparison with completely random validation flow.
- A special justified hardware support within the MMB-BIST for the developed validation methodology. Its presence allows to increase the performance of the validation on 37.72%.

**Practical significance and implementation of results.** The obtained results have served as a basis for a mature multilevel/hierarchical solution for MMB-BIST validation flow and test which increases the efficiency, reduces the test and validation time and improves their quality. The done work is not just a basis for further research in this area, but it also induces a direct application of the results to MMB-BIST test.

The obtained results are implemented at “Synopsys Armenia” CJSC and used in validation and QA flow for Synopsys MMB Compiler which is a component of DesignWare STAR Memory System (SMS) product. SMS is widely used by hundreds of customers. Particularly, more than half of top 25 companies in the world using semiconductors are considering SMS as their built-in memory test solution.

The implementation is done in a form of a special tool for automation of validation

and test, named “MMB Validator”. The data on the efficiency of the tool are adduced in the thesis.

The certificate of the implementation at “Synopsys Armenia” CJSC is adduced in the Appendix 1.

**The following scientific theses are presented for the defense:**

- Learning based methodology which generates an efficient validation method for a given MMB-BIST configuration from the methodology template via specific values of parameters and requirements.
- A dynamically built hierarchy for the validation and test of MMB-BIST features which leads to reduction of exhaustion when validating MBIST functionality and which allows to perform considerations for each iteration of the validation within a fixed time slot.
- A special hardware support within the MMB-BIST for the developed validation methodology to increasing the performance of the validation.

**Justification of results.** Justification is done using modeling and statistical methods for processing of experimental results.

**Presentation of the thesis.** The main scientific and practical results of the dissertation were presented at:

- 19th International Conference "East-West Design & Test Symposium (EWDTs)" (Batumi, Georgia, 2023);
- scientific seminars of the Department of Microelectronic Circuits and Systems of NPUA (Yerevan, Armenia, 2021 - 2024);
- research seminars of the Embedded Test and Repair Program at Synopsys Armenia CJSC (Yerevan, Armenia, 2021 - 2024).

**Publications.** The main results are published in 5 papers. These papers are included in the list of used literature of the thesis.

**Structure and the size of the thesis.** The thesis consists of an introduction, 4 sections, conclusions, a list of the used literature and 4 appendices. The list of literature includes 132 papers. The main volume of the dissertation is 104 pages, and together with appendices - 144 pages, including 61 figures and 5 tables.

## MAIN CONTENT OF THE WORK

**The introduction** substantiates the relevance of the dissertation topic, formulates the purpose and main objectives of the research, presents the developed methods, scientific novelty, practical significance, and the main scientific positions submitted for defense.

**The first chapter** shows that built-in self-test (BIST) has found wide application in integrated circuit (IC) industry.

*The section 1.1* describes that no IC can be guaranteed to be fault-free especially those with memory devices which occupy up to 70% of an IC<sup>2</sup>. Memory BIST (MBIST) is an efficient solution to test and repair these memories which can significantly increase IC manufacturing yield<sup>3</sup>. As the studies show<sup>1</sup>, each new node brings new defect types which were not relevant for previous technologies. To handle this, MBIST solutions are continuously updated which cover the faults modeled on these defects using test algorithms. Besides the algorithm creation, the solution is adapted to the given system on a chip (SoC). Meantime, the test solution should not significantly affect to the SoC

performance.

*The section 1.2* introduces existing MBIST solutions<sup>4</sup>. As an intellectual property (IP) block, MBIST solution should be flexible and usable for different SoCs. For that reason special software-based tools/compiler are created which have a generic solution as a template and generate a proper MBIST register transfer level (RTL) behavioral description/instance using provided input parameters' values which determine a configuration of the instance. That configuration contains information about memory features like its port sizes, usage, structure, MBIST features such as algorithms, modes, custom registers' sizes, and their usage, test options, etc. The more detailed are the configuration parameter values described, the more accurate the MBIST engine is created and, correspondingly, the test process of a given SoC is managed. Meantime, the validation of detailed configurations takes much more resources due to the number of validation scenarios which are formed by these configuration parameters. It is important to validate the generated MBIST instance with the given configuration. Some failures which may occur during MBIST generation, can disrupt its functioning and the SoC test will not be complete. A famous MBIST solution which supports detailed descriptions of the memory devices and more is SMS.

SMS is known for more than two decades as comprehensive test solution for SoC memories. It supports different testing and diagnostic capabilities as

- Advanced memory test and repair algorithms that efficiently address new memory defects for the most recent technological nodes,
- A mechanism for new faults prediction in memories for upcoming technologies and its reflection in the architecture of a MBIST engine,
- A unified MBIST interface with memory instances,
- A language for description of memory structural models and methods for its efficient use,
- A hierarchical architecture for test of memory instances in SoC that delivers essential increase of test efficiency,
- Flexibility to schedule individual memory instance for parallel or serial testing to optimize time and power consumption during test,
- New methods and standards of data porting between different levels of the test hierarchy,
- Protection against soft errors with multi-bit upset error correcting codes (ECC),
- Advanced failure diagnosis with physical failed bitmaps and XY coordinates of failing memory bit cells.

Originally, SMS uses memory test interface to apply its operations. A test interface allows to switch a memory from functional to test mode and be tested through test infrastructure. Once there are different memory structures in a SoC, SMS wraps these memories with a wrapper circuit which allows SMS processor to have similar access to those memories (Fig. 1).

One of the validation approaches for SMS MBIST is black-box testing. Regardless the internal implementation of MBIST generator, configurations are created and given to validate the generation of MBIST RTL description. The generation of configurations usually is randomized. This approach is efficient for creating big volume testcases which are hard to do manually.

Because the validation of a MBIST IP is usually done by its vendors which have

implementation details, the white-box testing can be applied. Using this approach, the scope of configurations can be outlined more efficiently and with time saving.

All validation approaches are combined with the regression. To ensure that the current version of MBIST generation tool is compatible with previously used scenarios, regression suit is continuously updated, and corresponding test is applied.

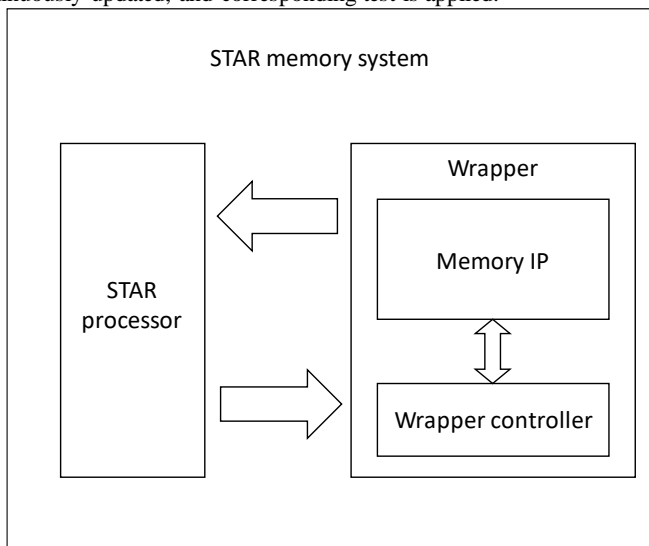


Fig. 1. SMS solution for single memory instance

The section 1.3 introduces another method of testing memories which uses functional paths existing in a SoC<sup>5</sup>. The test interface insertion is not acceptable for these SoCs, and the only approach is to use already inserted paths (Fig. 2). To make these paths usable by the test engine, they are grouped together to form a multi-memory bus (MMB) to which the MBIST engine is connected. This approach saves area compared to the conventional SMS solution, where additional routing is done for inserted test interface.

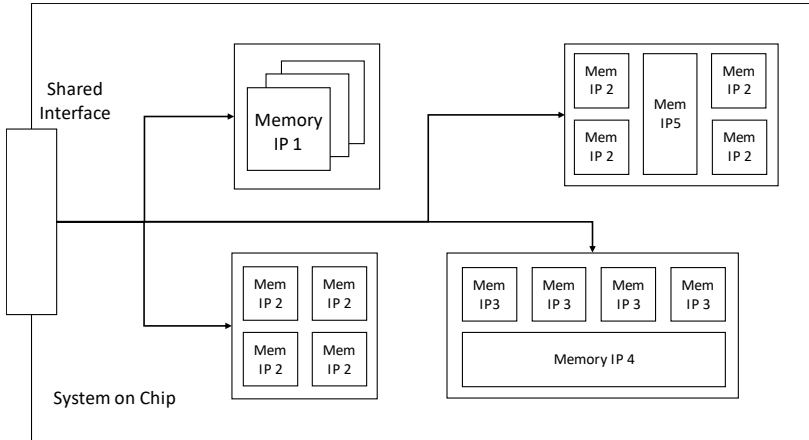
The section 1.4 explains that the validation and test methods which were applicable for SMS solution, rise problems for MBIST engines which use MMB. In particular:

- the exhaustion over possible scenarios is inadmissible time consuming. The number of dynamically grown parameters are more than those for conventional MBIST,
- testing through shared interface consumes more test time, than testing through wrappers as in the case of SMS. All memory instances are driven by the same interface and parallel testing of all these instances is not appropriate. Correspondingly, the validation process will last longer,
- the configuration of MMB-BIST is more complicated in comparison with the conventional MBIST which uses test interface. It contains information about MMB interface and its connections to memories, which are not relevant for other type of MBIST. The description of this information may have some mismatches

that affect the MBIST integration.

Considering pointed challenges, a detailed analysis needs to be performed to effectively validate MMB-BIST generation and functioning.

**The second chapter** presents the developed methods and provides solutions to the problems described in the section 1.4.



*Fig. 2. MBIST application through shared interface*

*The section 2.1* introduces the necessary **definitions and modeling of MMB-BIST features to be validated**.

In general, to validate the functionality of an MMB-BIST engine, the scope of testing features should be defined. The validation of systems which are described with parameters, can be automated implementing an environment specific to validation subject. The preferable approach for such systems is the randomized test which can provide both a big number of possible appropriate combinations and supports reasonable coverage as well. In fact, MMB-BIST configuration generation is much more complex compared to conventional MBIST engines. There are parameters which scopes are dynamically changed considering other parameter values, though there are independent parameters with numeric or string types.

The general structure of MMB configuration is illustrated in Fig. 2. It consists of different number of memory groups which are connected to the shared bus. Each memory group is described by the number of parallel and serially accessible memories, and its structural parameters such as number of words, ability to be repaired, etc. Sufficient grouping of these parameters will increase validation efficiency. Studying the specification of MMB-BIST structure, a hybrid validation approach is suggested which uses these groupings of parameters applying determined and randomized test. The following features/parameters which will further call as primary parameters were defined for determined test;

- Number of test buses – the MMB-BIST solution should handle with multiple



buses existing in the SoC. Considering the number of memory groups, the order of testing and other factors, some SoCs have multiple buses for better test performance.

- Number of memory groups – the main purpose of testing through shared bus is to work with multiple memory instances which number may vary depending on a SoC.
- Repair capabilities – a memory can have different structure of redundant elements which are used for its repair. Correspondingly, MBIST solutions should handle with various redundancy configurations.
- Number of operation cycles – some memory instances can have latencies which should be considered by MMB-BIST engine.
- Masking of memories – memories frequently have masking ports which are used for different purposes. In some SoCs, memory access is modified using this feature.
- Partially connections – MMB-BIST solution should handle the scenarios where there are memories which connections are not fully accessible such as partially data port connections.
- Number of memory ports – memories with multiple ports are driven by separate drivers existing in the bus. Algorithms which are applied to multi-port memories, should be executed in the same way independently of their interaction interface (via a shared or a dedicated test interfaces).
- Number of memory banks – memories which have multi-bank structure, have some specification which should be taken care by MMB-BIST. The functioning for most of the features are changed if a memory has multiple banks.

*In the section 2.2* a way of organizing an **efficient exhaustion of parameters values for validation** is presented. To effectively use values of the described primary parameters, special binary forms of representation are used. The latter determine the presence or absence of corresponding features. For example, the binary representation of the first parameter means that a value of zero corresponds to an SoC with single test bus and one for several. The rest of the parameters which describe MMB configuration, are considered by using random values. These are the parameters which have some dependencies on above pointed ones and assessed as non-primary. Considering all these descriptions, the tree of parameters is formed and shown in Fig. 3.

After tree of parameters is formed, it is necessary to understand which parameters have initially high priorities. For example, it is not sufficient to run redundancy allocation simulations if an MMB-BIST has generation issue on an operation application. Though the redundancy allocation circuit is separate component in an MMB-BIST, it interacts with other components especially with BIST controller. Another important criterion is how easy the issue is to fix. The faster the current issues are resolved, the more efficient the validation process will be.

Considering described criteria primary parameter weights are defined using analytic hierarchy process (AHP)<sup>6</sup>. The calculations performed show that the pre-selected weights

---

<sup>6</sup> T.L. Saaty, Decision Making with the Analytic Hierarchy Process, International Journal of Services Sciences, 2008, 1, 83.

correspond to each other, from which the primary parameters can be arranged in accordance with the selected priority. The AHP application is useful especially when the number of overall testcases is huge and can be executed in multiple iterations.

In the section 2.3 a learning-based methodology for MMB-BIST validation is described.

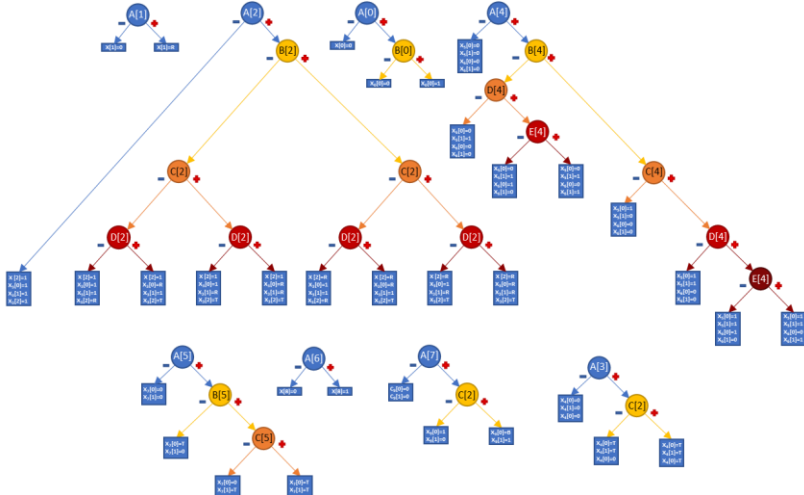


Fig.3. Tree of the MMB configuration parameters

Validation of an automated system is considered as exhaustive execution of scenarios composed with parameters which describe that system. These parameters have different types and ranges of values which are defined and documented.

To generate a testcase, a configuration is formed which contains some values of these parameters. Here, some challenges appear if there are non-permitted combinations of these parameters. The automation of configuration generation using such parameter sets is hard to implement. On the other hand, classification and analyzing problems appeared when generating configurations regardless of whether the parameters are valid or not.

Mentioned challenges are overcome by implementing random testing environment which generates random configurations within a given scope of parameter values. Naturally, this approach cannot cover all possible scenarios, and some important scenarios are executed after multiple iterations of testing environment usage. Here, parameter classification will simplify configuration generation process. This classification should be done considering the structure of the subject under validation.

The configuration parameters which describe MMB-BIST generally describe existence or absence of certain feature, i.e., these are binary parameters. Having the set of these binary parameters, exhaustive search can be applied. But, as mentioned before, the full exhaustion of these parameters takes unacceptable time, and randomization of configurations is preferred. Experiments show that combining randomization with some exhaustion yields better results.

The Fig. 4 represents an approach which allows to localize failures and reduce number

of testcase executions during validation. The localization is done by classifying binary parameters' occurrences in failing testcases. For the current iteration of generation the failure percentage is calculated, and binary parameters are sorted by their relevance. The parameters which occur often in the failed cases, are firstly fixed the value which sensitize failures and then inverted for analysis the rest of parameters.

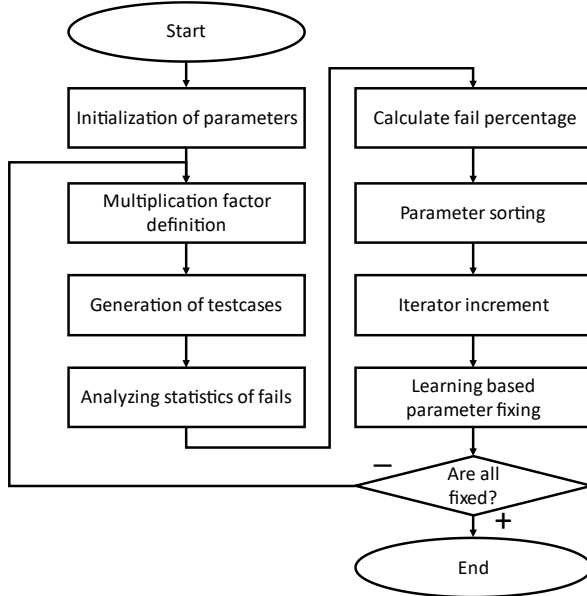


Fig. 4. Learning based validation block-diagram

In fact, the suggested approach emits a set of formulas for MMB-BIST solutions. Each of these formulas differs by coefficients that are unique to a given solution. These coefficients describe comparison threshold values and can be determined in two ways. One of the methods is to collect statistics by continuous execution of QA environment with this approach giving random values to the coefficients. The collected statistics can be a dataset for a machine learning environment which will be trained and predict how to adjust these coefficients. Another way is to classify MMB-BIST configuration parameters considering structural specifications. There are some features that need to be validated first to accelerate increasing of validation coverage.

Due to the described approach the hierarchy of considered parameters becomes dynamically built basing on results of each iteration step. In its turn it brings to an essential increase for the efficiency for test case generation in comparison with using a direct exhaustion for generation.

**The third chapter** introduces extension and optimization approaches of MMB-BIST validation environments (VE).

*The section 3.1* describes the differences in test solutions by memory device types. It

is shown that the classification and prioritization of features can be applied not only to MMB-BIST but also to other test solutions. One of these solutions is testing magnetic RAMs (MRAM).

Due to its characteristics MRAM has a potential to become universal memory device<sup>7</sup>. Meantime, the challenges related to its operating margin are still relevant. To overcome these challenges dedicated calibration mechanisms are suggested and implemented<sup>8</sup>.

In respect that MRAM has its own peculiarities, the validation model of MRAM test solution slightly differs compared to MMB. These models are represented in Fig. 5.

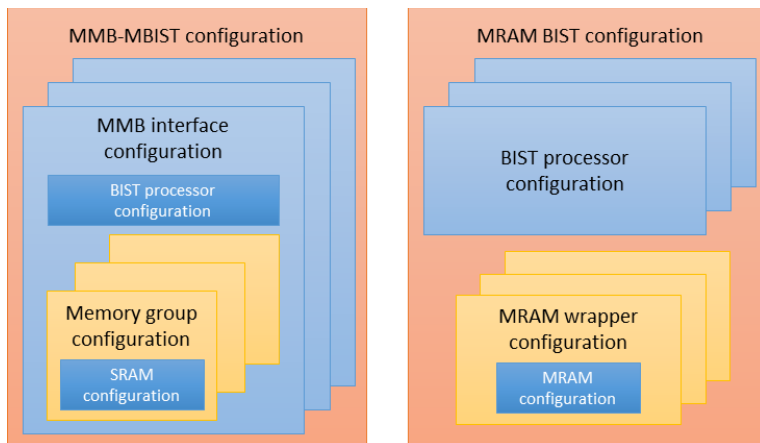


Fig. 5. MMB-BIST and MRAM BIST validation models

The section 3.2 discusses **validation reduction method by MMB-BIST structure modification**.

During the application of the VE it was found that the validation improvements can be achieved by the improvements of characteristics MMB-BIST. Due to specific structure of MMB-BIST there can be large configurations and corresponding simulations of generated RTL descriptions will take long time. In general, execution time of an algorithm takes most of the simulation time. Therefore, reducing testing time will directly reduce validation time.

MMB-BIST test time can be reduced considering its architectural specification. For each memory port corresponding driver exists on the shared bus (Fig. 6). The operations

<sup>7</sup> Y. -D. Chih et al., "13.3 A 22nm 32Mb Embedded STT-MRAM with 10ns Read Speed, 1M Cycle Write Endurance, 10 Years Retention at 150°C and High Immunity to Magnetic Field Interference", IEEE International Solid-State Circuits Conference - (ISSCC), San Francisco, CA, USA, 2020, pp. 222-224.

<sup>8</sup> X. Liu, C. Xu, P. Yu, Y. Jiang, "Calibration and Parameter Extraction of STT-MTJ Device at High Frequency by Using De-Embedding Approach Based on TRL Calibration", IEEE Transactions on Electron Devices, 2021, vol. 68, no. 1, pp. 271-278,

described in test algorithm can be applied using one or more group of drivers. A simple group of drivers consists of memory read/write enable, clock, data, and address ports. Test time of an algorithm can be reduced if its operations use only one port group.

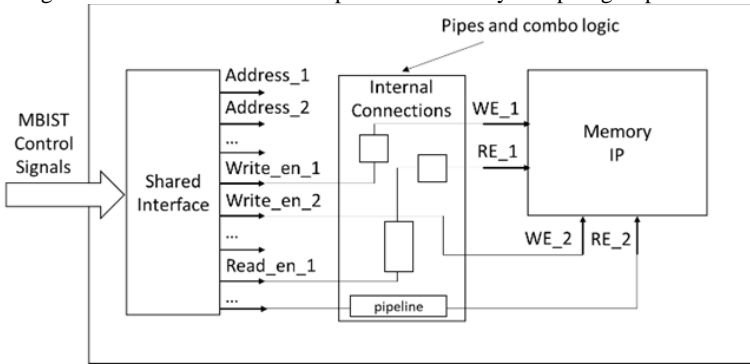


Fig. 6. Shared interface to memory connection

The main idea of test time reduction is to apply operations for all ports a memory has. That is, if a memory has two read-write ports, the operations can be shared between these two (Fig. 7). This approach shows its advantage when there are multiple memory instances with multiple groups of ports.

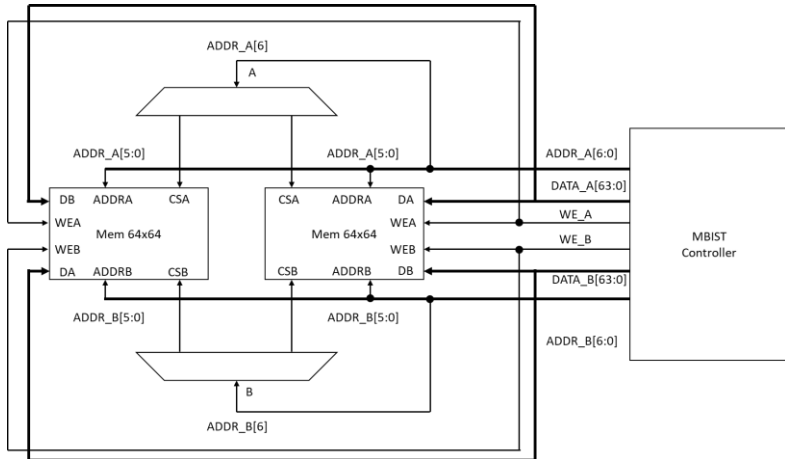


Fig. 7. Connection of two dual-port memory instances

In Fig. 7, the connection of two dual-port memory instances is represented. Suppose that these memories cannot be tested in parallel because the bus data size is enough for only one instance. It is possible to access each memory with one of the port groups modifying the structure of MBIST address counter structure. For the example shown in the

figure, when read operation applied on the left instance by A port, the right one can be with B port. The address bit, which is responsible for memory access, should be different for each group, i.e., ADDR\_A[6] = 1, ADDR\_B[6] = 0. With this approach, the execution of the algorithm will be halved.

This approach can be applied for more port groups. If there are three memories with three read-write ports (Fig. 8), the application of the suggested approach should be performed with the sequence described in Table 2.

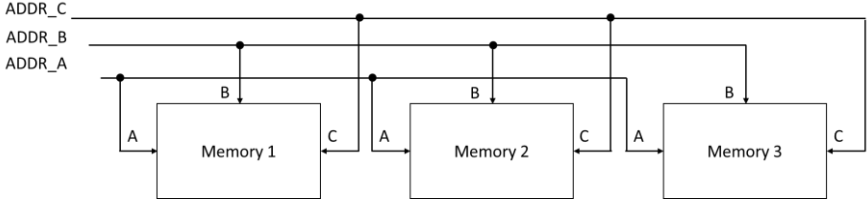


Fig. 8. Connection of tree tree-port memory instances

Table 2

Algorithm execution for three-port memories

STEP	Chip Select	Memory 1	Memory 2	Memory 3
1	CSA	Active	Inactive	Inactive
	CSB	Inactive	Active	Inactive
	CSC	Inactive	Inactive	Active
2	CSA	Inactive	Active	Inactive
	CSB	Inactive	Inactive	Active
	CSC	Active	Inactive	Inactive
3	CSA	Inactive	Inactive	Active
	CSB	Active	Inactive	Inactive
	CSC	Inactive	Active	Inactive

The section 3.3 discusses **validation challenges of programmable features of the MMB-BIST.**

It is shown in the previous subchapter that some structural improvements can reduce MMB-BIST validation time. Actually, not all structural improvements are helpful for the validation. The addition of programmable features is one of these which increases exhaustion by its variety of states. Though the programmability of a feature may increase test coverage after production, it should be properly validated in the scope of its usage. The need for a programmable feature increases when there are limitations such as MMB-BIST has.

A feature which may increase test coverage is in field operation programmability. The main idea is to program bus to memory port connections through specific operation box

which contains information about operation application (Fig. 9). Each row of the operation box has a specific and carries information for one operation. It is crucial to distinct validation scenarios which have high priority.

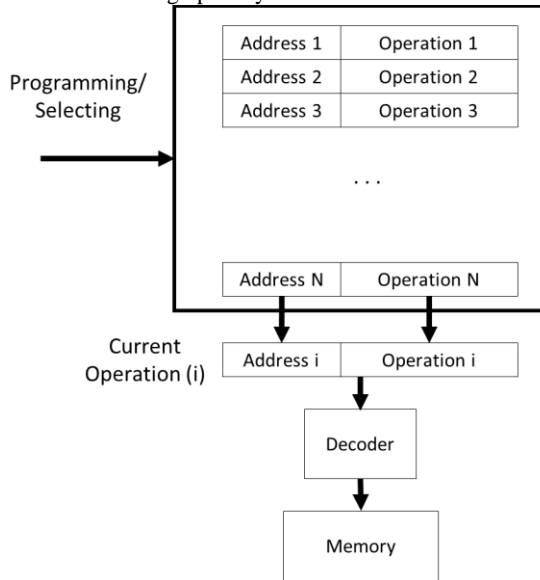


Fig. 9. Programmable operation box

**The fourth chapter** presents methodology implementation details and corresponding experimental results.

*The section 4.1* introduces a software tool “MMB Validator” implemented for the testcase generation of MMB-BIST engine using Python programming language and its PyQt5 library. The tool’s main window is shown in Fig. 10. The tool enables the following features;

- Generation of MMB configurations based on described parameters.
- Configuration based design generation in Verilog HDL
- Ability to define maximum values for design parameters
- Visualization of configurations

The MMB configuration is described on JSON format, which is widely used in different areas and supported by various programming languages. The description contains information about shared buses, memory groups and some test specific options. Each bus consists of signals and for each signal number of properties are defined. The description a memory group contains the description of its ports, redundancy structure, address scrambling and how they are connected to a bus. The generalized structure of a configuration is represented in Fig. 11.

MMB Validator enables definition of the randomization scopes for bus configurations. The maximum values of these parameters are given through MMB Generator configuration file

- Number of buses (interfaces)
- Number of memory groups
- Number of parallel working memories in a group
- Number of serially working memories in a group
- Read/Write latencies
- Number of pipelines

Some string parameters such as bus signal names are parameterized. Memories are chosen depending on the specified testcase scenario. In case of absence corresponding memory, a warning message is printed. Each testcase schematic diagram can be opened after generation (Fig. 12).

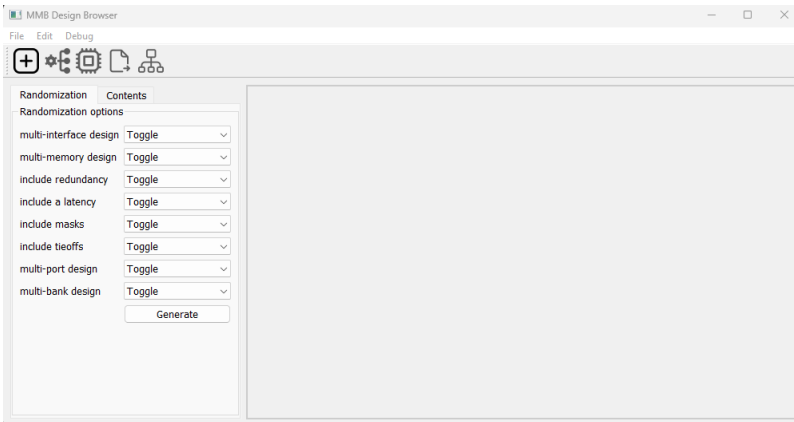


Fig. 10. MMB Generator GUI

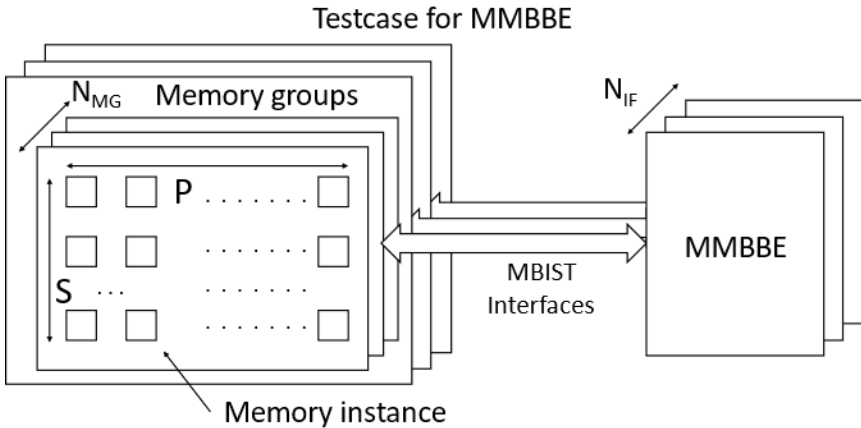
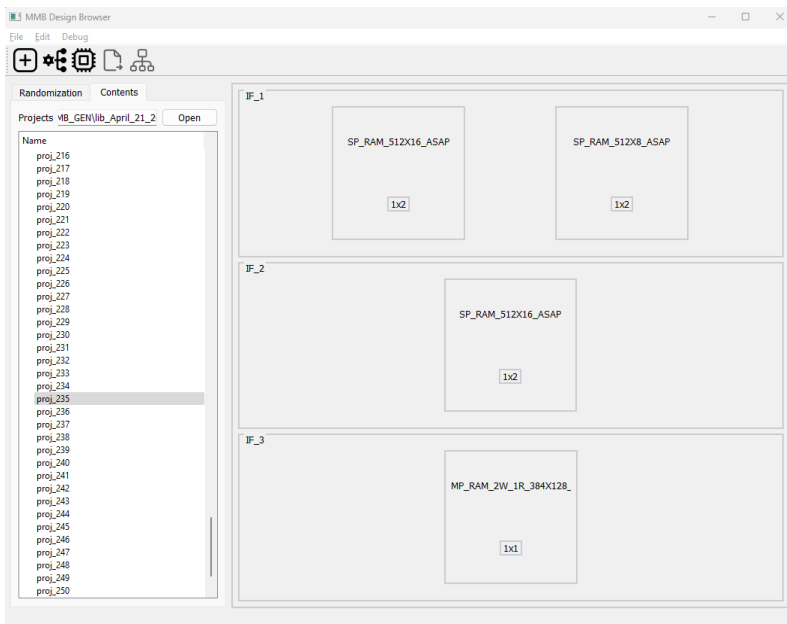


Fig. 11. MMB configuration general model



*In the section 4.2* it is shown how the MMB-BIST engine configuration dataset is formed. It is known that the efficiency of predictions depends on how effectively the dataset is formed. It is shown in the thesis that generic artificial intelligence (AI) tools can be used for that purpose and the domestic AI tool developed for SMS product test and QA is chosen. Mostly these tools accept fixed number of columns which correspond to input parameters. As MMB-BIST engine has a hierarchical structure which is dynamically grown, some acceptable ranges for MMB configuration parameters should be determined. Some parameters such as bus names are dropped because their numeric parameters are provided as well. On the other hand, the number of certain buses such as address or data cannot be endlessly increased.



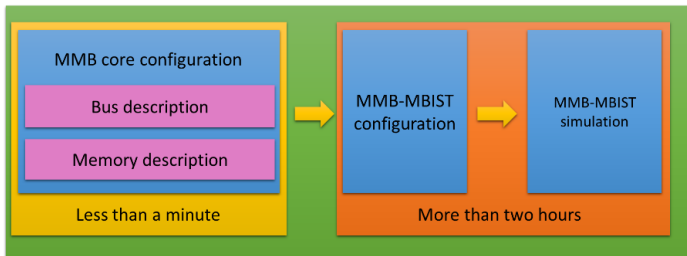
*Fig. 12. MMB configuration visualization*

Use of the mentioned AI tool leads to acceleration of the MMB-BIST validation time. As it is shown in the Fig. 13, the most of time is taken by MMB-BIST engine generation and simulation. The generic AI tool can be used to predict the MMB configurations which are most likely to have issues. The experiments show that the number of failing cases does not exceed 30% of all generated cases. Due to that the prediction efficiency for MMB configurations will directly affect validation time.

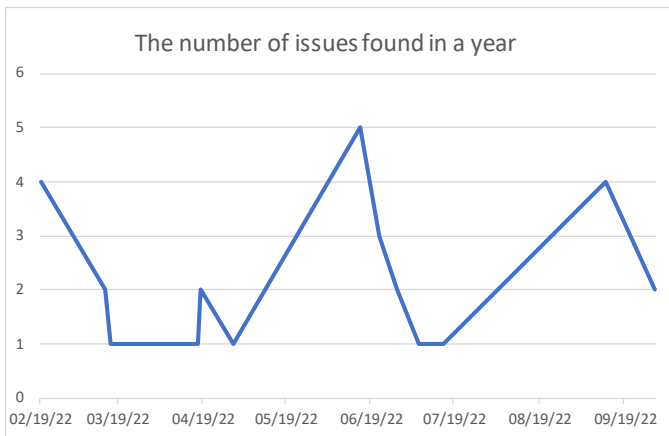
*The section 4.3* summarizes MMB-BIST validation methodology application results. The described classification and prioritization were basis for development of a MMB-BIST validation environment (VE) which allows to generate different MMB configurations. All

methods proposed in the thesis have been applied using this VE which is continuously improved. Particularly

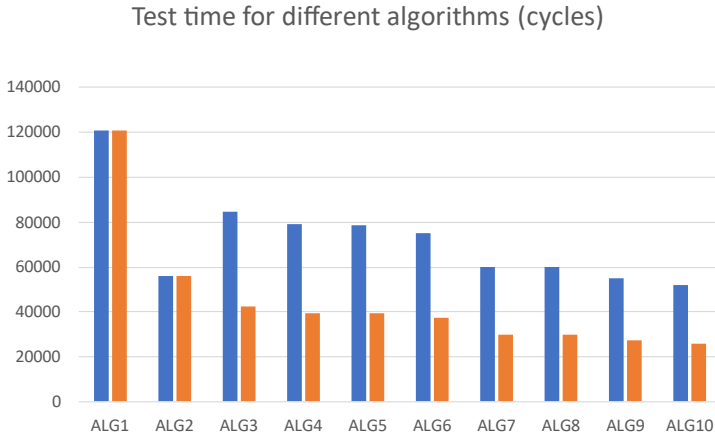
- VE effectively finds MMB-BIST errors which were not detected before. Meantime, it concentrates on the defined primary parameters keeping validation time relatively fixed. The statistics of the bugs found in a year is represented in Fig. 14.
- Each validation example which is parameterized Verilog description, allows to get another scenario making minor changes. Parameterized description is useful especially for the MMB-BIST development process because it helps to consider near scenarios without creating separate testcases for each.
- validation time was reduced by an average of 30% by applying the learning-based methodology. Before, all iterations of VE executions have considered all combinations of primary parameter values.
- The application of the approach suggested in 3.2 reduces overall MMB-BIST test time by 37.72%. The comparison of some algorithms is represented in Fig. 15.



*Fig. 13. MMB-BIST validation steps*



*Fig. 14. Failure statistics using hybrid validation*



*Fig. 15. Test time reduction approach comparison*

### MAIN CONCLUSIONS OF THE THESIS

1. A solution for validation is suggested for MMB-BIST engines validation basing on analysis of validation problems for multi-memory bus MBIST (MMB-BIST) engines which are not overcome via existing approaches. The efficiency of the approach is proved experimentally.
2. A validation model is introduced which describes the main features of a SoCs which have memories accessible and testable through shared interfaces. Using this model, a validation environment is implemented.
3. A learning-based validation methodology is suggested and justified comprised of the following steps:
  - outlining of primary validation parameters for an iterative exhaustion of their values;
  - random selection of values for the rest of parameters;
  - reduction of the number of iterations via finding dependencies between primary parameters and
  - reduction of the randomly selected values for the rest of parameters via preserving the number of test cases considered for each step of the iteration invariant;
  - at the end of an iteration step a comparison of detected bugs with the initially defined threshold value is performed. If the number is over the mentioned threshold, then the next step of the iteration is started, otherwise inverting of values engaged in the considered parameter dependencies is performed and a process of finding new dependencies of parameters starts.
4. A validation environment is built which combines full and randomized exhaustion according to the suggested methodology and built hierarchy. An estimation of the efficiency for the implemented environment is obtained experimentally. Due to its application:

- the total number of considered random test-cases was reduced in average by 30% preserving the list of bugs obtained during a purely random generation of test-cases;
  - the generation of testcases was accelerated more than 10 times;
  - the number of bugs detected by customers within a fixed time interval was reduced in average by 3 times.
5. Special extensions of the methodology are considered for other types of memory devices and additional accelerating of the validation process as well as slowdown factors.
  6. A suggestion on further investigations which extends more the approach considered in the thesis.

**The main results** of the thesis are published in the following papers:

1. Babayan A. BIST Architecture for Magnetic Memories // Mathematical Problems of Computer Science (MPCS). – December, 2020. – Vol. 54. – P. 88-95.
2. Babayan A. Enabling In-Field Operation Programmability for Built-In Self-Test // Vestnik RAU: Physics, Mathematics and Natural Sciences. – 2021. – no. 1. – P 142-149.
3. Babayan A. Validation and Test Time Reduction Approach for Built-In-Self-Test and Repair of Memories Via a Shared Interface // Proceedings of the NAS RA and NPUA. Series of Technical Sciences. – 2022. – Vol. 75, no. 3. – P. 415-423.
4. Babayan A. Validation and Test Challenges for Multi-Memory Bus BIST Engines // IEEE East-West Design & Test Symposium (EWDTS). – September, 2023. – P. 1-6.
5. Babayan A., Shoukourian S. Learning Methodology for Validation of Memory Bist Solutions via a Shared Interface // Proceedings of the NAS RA and NPUA. Series of Technical Sciences. – 2024. – Vol. 77, no. 1. – P. 69-76.

## ԱՄՓՈՓԱԳԻՐ

Օրեցօր զարգացող ինտագրալ սխեմաներին (ԻՍ) ներկայացվում են նոր մարտահրավերներ, որոնց հաղթահարման համար մշակվում են նոր մոտեցումներ: Տրանզիտորի չափերի փոքրացումից, տեխնոլոգիական գործընթացի, լարման և ջերմաստիճանի փոփոխություններից զատ, փոխվում են ԻՍ-երում առկա լուծումների մոտեցումները, մասնավորապես, դրանց ճարտարապետությունը: Այս լուծումներն իրականացնելիս կարևոր է հաշվի առնել դրանց հուսալիությունը, որից կախված է արտադրության պիտանի ելքի տոկոսը:

ԻՍ-երի գերակշիռ մասը հիշող սարքերն (<Ն) են, և հիմնականում դրանց անխափան աշխատանքից է կախված այդ սխեմաների հուսալիությունը: Մյուս կողմից՝ <Մ-երը նախագծվում և արտադրվում են նվազագույն ծախսերով և առավել հակված են անսարքությունների: Այս խնդիրները լուծելու համար, կիրառում են <Մ-երի ներկառուցված թեստավորման և նորոգման համակարգեր (ՆԹՆՀ), որոնք գտնում և վերանորոգում են առաջացող անսարքությունները: Արտադրական տեխնոլոգիայի զարգացումը ցույց է տվել, որ նոր տեխնոլոգիայով արտադրված <Մ-երում հանդիպում են արատներ, որոնք չեն եղել նախկինում, իսկ առկա թեստավորման լուծումները բավարար չեն այդ արատների հետևանքով առաջացած անսարքությունների հայտնաբերման համար: Ուստի շարունակաբար բարելավվում են ՆԹՆՀ-երում կիրառվող լուծումները, որոնք դարձնում են այդ համակարգերն ավելի ճկուն և արդյունավետ:

ՆԹՆՀ-երը պատվիրատուների տարատեսակ ԻՍ-երում օգտագործվելու համար իրականացվում են ավտոմատացված հոսքուղիների միջոցով, որոնք տրված <Մ-երի նկարագրությունից գեներացնում են տրված միջուկին համապատասխան սխեմա: ՆԹՆՀ-ի՝ միջուկում ինտեգրման դասական մոտեցումը թեստային միջերեսի ներմուծումն է, որի միջոցով ամբողջական հասանելիություն է ապահովվում <Մ-երին: Որոշ պատվիրատուների համար ընդունելի չէ թեստային միջերեսի ներմուծումը, քանի որ ներմուծված միջմիացումներն ու հարակից տրամաբանությունը կարող են ազդել միջուկի այն պարամետրերի վրա, որոնք առաջնային են: Սակայն այդ միջուկներում <Մ-երն իրար են միացված որոշակի կապուղիներով, որոնց օգտագործում են միջուկում առկա մյուս հանգույցները: Որպես կանոն, ՆԹՆՀ-ը ևս միացվում է այս կապուղիներին, իսկ թեստավորումը կատարվում է համապատասխան ռեժիմում, երբ դադարում է մյուս հանգույցների աշխատանքը: Ընդհանուր կապուղով թեստավորող համակարգերը (ԸԿԹՀ) նույնպես գեներացվում են համապատասխան հոսքուղու միջոցով, որոնց նկարագրության համար անհրաժեշտ է ոչ միայն <Մ-երի և դրանց ԸԿԹՀ-երի գործառնություններ, այլև միջուկում այս երկու խմբերի կապերը նկարագրող պարամետրերի բազմությունը: Կապուղիների և <Մ-երի նկարագրության թերի լինելը, ԸԿԹՀ-ի գեներատորում մարդկային և այլ գործոններով պայմանավորված հնարավոր սխալները կարող են առաջացնել գեներացված սխեմայի որոշակի գործառնությունների սխալներ, որոնց հետևանքով կխափանվի միջուկի թեստավորումը: Մյուս կողմից՝ ԸԿԹՀ-ն նկարագրող պարամետրերի և դրանց հնարավոր արժեքների քանակը բավականին մեծ է, իսկ դրանց հատարկման ժամանակը՝ անընդունելի: Նշված խնդիրները լուծելու համար կիրառվում են այսպիսի համակարգերին հատուկ վավերացման միջավայրերը:

Ատենախտությունը նվիրված է ԸԿԹՀ-երը և դրանք գեներացնող ավտոմատացված հոսքուղիները վավերացնող միջավայրի հետազոտմանը և

վավերացման արդյունավետությունը բարձրացնող միջոցների մշակմանը:

Առաջարկվել են ԸԿԹՀ-ները վավերացնող մոտեցումներ, որոնք, հաշվի առնելով ԸԿԹՀ-երի կառուցվածքային առանձնահատկությունները, ապահովում են վավերացման ծածկույթի ան և տևողության կրճատում:

Առաջարկվել է վավերացման ուսուցանվող մոտեցում, որը տեղայնացնում է ԸԿԹՀ-եր գեներացնող միջավայրում առաջացող սխալները: Մեթոդաբանության կիրառմամբ առաջարկվել է ԸԿԹՀ-ի վավերացման մոդել, որով սահմանվում են վերջինիս հիմնական հատկությունները, և դասակարգվում դրանք նկարագրող պարամետրերը: Այդ հատկությունները ցույց են տալիս միջուկում ՀՄ-երի և դրանց միացված կապուղիների քանակական և որակական ցուցանիշները, որոնցից են ՀՄ-երի և կապուղիների պորտերի կարգայնությունը, կապուղիներին միացված ՀՄ-երի զուգահեռ կամ հաջորդաբար միացումները և դրանց թիվը, կապուղիներին միացված համակցական և հաջորդական տրամաբանությունը և այլն: Հաշվի առնելով ԸԿԹՀ-ի և դրա գեներատորի պարամետրերը՝ սահմանվում է գեներատորին հատուկ վավերացման մեթոդ, որը որոշում է այդ գեներատորով գեներացված ԸԿԹՀ-ի հատկությունների վավերացման առաջնահերթությունները: Այս առաջնահերթությունները հետադարձ կապով կարգաբերվում են ընթացակարգի ժամանակ հայտնաբերված սխալների վիճակագրության վերլուծությամբ:

Առաջարկված մոդելի հիման վրա նախագծվել է վավերացման միջավայր, որը ստեղծում է տարատեսակ ՀՄ-երն ընդհանուր կապուղով թեստավորման հնարավորությամբ վիրտուալ միջուկներ: Միջավայրը կատարում է սահմանված հիմնական պարամետրերի լրիվ հատարկում, և ոչ հիմնական պարամետրերի պատահական ընտրություններով հատարկում: Վավերացման միջավայրը թույլ է տալիս օգտագործողին որոշել ոչ հիմնական պարամետրերի արժեքների տիրույթը, որոնցից պատահականության սկզբունքով ընտրվում են այդ պարամետրերի արժեքները: Միջավայրի կիրառությունն արագացրել է ԸԿԹՀ-ի վավերացման օրինակների ստեղծումն ավելի քան 10 անգամ՝ վավերացման առկա մեթոդների համեմատ: Պատվիրատուների կողմից հայտնաբերված սխալների քանակը նվազել է միջինը 3 անգամ: Միջավայրում ներդրված ուսուցանվող մեթոդաբանությունը կրճատում է վավերացման ժամանակը միջինը 30%-ով՝ մինչ այդ առկա լրիվ հատարկմամբ եղանակի համեմատ:

Ուսումնասիրվել է առաջարկված մեթոդների ընդլայնման հնարավորությունը այլ տեսակի ՀՄ-երի և դրանց թեստավորող համակարգերի դեպքում:

Առաջարկվել է ԸԿԹՀ-ի հասցեավորման մոտեցում, որը, սխեմայում փոքր փոփոխության շնորհիվ, զուգահեռացնում է մի քանի պորտ ունեցող ՀՄ-երի թեստավորումը՝ կրճատելով համակարգի թեստավորման ժամանակը: Արդյունքում՝ ԸԿԹՀ-ի վավերացման ժամանակը կրճատվում է միջինը 37.72 %-ով: Դիտարկվել են նաև այն սխեմատեխնիկական փոփոխությունները, որոնք բարդացնում են ԸԿԹՀ-ի վավերացման գործընթացը:

Նախագծվել է ծրագրային միջոց, որը ներառում է վավերացման միջավայրը և թույլ է տալիս օգտագործողին գեներացնել ցանկացած քանակով վավերացման օրինակներ և ուսումնասիրել վավերացման ժամանակ առաջացող սխալները: Ծրագրային միջոցը կարող են կիրառել ինչպես վավերացմամբ զբաղվող, այնպես էլ նախագծող նարտարագետները:

**РАЗРАБОТКА СРЕДСТВ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ  
ОБЕСПЕЧЕНИЯ КАЧЕСТВА РЕШЕНИЙ СИСТЕМ ТЕСТИРОВАНИЯ,  
ВСТРОЕННЫХ В НАНОРАЗМЕРНОМ КРИСТАЛЛЕ**

**ЗАКЛЮЧЕНИЕ**

С течением времени разработчики интегральных схем (ИС) сталкиваются с новыми проблемами, и для их преодоления разрабатываются новые подходы. Наряду с уменьшением размеров транзистора, изменением технологического процесса, напряжения и температуры меняются подходы к существующим решениям в ИС, в частности, их архитектура. При реализации этих решений важное значение имеет учёт их надёжности, которая зависит от процента годного выпуска продукции.

Большинство ИС являются запоминающие устройства (ЗУ), и, надёжность этих схем, главным образом зависит от надёжности ЗУ. С другой стороны, ЗУ проектируются и производятся с минимальными затратами, что делает их более склонными к неисправностям. Для решения этих проблем используются встроенные системы тестирования и ремонта (ВСТР) для ЗУ, которые находят и устраняют возникающие неисправности. Развитие технологии производства показало, что в ЗУ, изготовленных по новой технологии, встречаются дефекты, не существовавшие ранее, и имеющиеся решения по контролю недостаточны для выявления неисправностей, вызванных этими дефектами. Поэтому решения, используемые в ВСТР, постоянно совершенствуются, что делает эти системы более гибкими и эффективными.

Для использования заказчиком ВСТР применительно к различным ИС их реализуют посредством автоматизированных рабочих процессов, формирующих на основе описания данных ИС, соответствующую заданному ядру. Классический подход к интеграции ВСТР в ядро заключается во внедрении тестового интерфейса, обеспечивающего полный доступ к ЗУ. Для некоторых заказчиков внедрение тестового интерфейса неприемлемо, поскольку внедренные интерфейсы и связанная с ними логика могут повлиять на основные параметры ядра. Однако ЗУ в этих ядрах соединены друг с другом определенными каналами, которые используются другими узлами ядра. Как правило, к этим каналам подключается и ВСТР, и тестирование проводится в соответствующем режиме, во время которого работа других узлов прекращается. Системы тестирования с разделяемым интерфейсом (СТРИ) также генерируются соответствующим потоком, для описания которого необходимы не только функции ЗУ и их СТРИ, но и набор параметров, описывающих связи этих двух групп в ядре. Неполное описание интерфейсов и ЗУ, возможные ошибки, вызванные человеческим фактором и другими факторами в генераторе СТРИ, могут вызвать погрешности в отдельных функциях генерируемой схемы, что приведет к сбою тестирования ядра. С другой стороны, количество параметров, описывающих СТРИ, и их возможных значений достаточно велико, а время их пересечения неприемлемо. Для решения упомянутых проблем используются среды валидации, характерные для этих типов систем.

Диссертация посвящена исследованию среды валидации СТРИ и их генерирующих автоматизированных потоков, а также разработке средств повышения эффективности валидации.

Предложены подходы к валидации СТРИ, которые с учетом структурных особенностей последней обеспечивают увеличение покрытия и сокращение времени.

Предложен подход к валидации на основе обучения, который локализует ошибки, возникающие в среде, генерирующей СТРИ. С использованием предлагаемой методики разработана модель валидации СТРИ, с помощью которой определены основные свойства последней и классифицированы описывающие их параметры. Эти свойства отображают количественные и качественные показатели ЗУ и подключенных к ним интерфейсов в ядре, в частности, порядок портов ЗУ и интерфейсов, параллельные или последовательные соединения ЗУ, подключенных к интерфейсам, и их количество, комбинационная и последовательная логика, подключенная к интерфейсам и т.д. Учитывая параметры СТРИ и её генератора, получается метод валидации для конкретного генератора, который определяет приоритеты валидации свойств СТРИ, генерируемых этим генератором. Эти приоритеты корректируются в обратной связи путем анализа статистики ошибок, обнаруженных в ходе процедуры.

На основе предложенной модели была спроектирована среда валидации, создающая виртуальные ядра с возможностью тестирования различных ЗУ с общим интерфейсом. Среда выполняет полный перебор по заданным основным параметрам и случайный перебор по неосновным параметрам. Среда валидации позволяет пользователю определить диапазон значений неосновных параметров, из которого значения этих параметров выбираются случайным образом. Использование среды ускорило процесс создания примеров проверки СТРИ более чем в 10 раз по сравнению с существующими методами валидации. Количество ошибок, обнаруженных заказчиками, сократилось в среднем в три раза. Обучаемая методология, реализованная в среде, сокращает время валидации в среднем на 30% по сравнению с полным случайным перебором.

Изучена возможность распространения предложенных методов на другие типы ЗУ и системы их тестирования.

Предложен подход к адресации СТРИ, позволяющий за счет небольшого изменения схемы распараллеливать тестирование ЗУ с несколькими портами, сокращая время тестирования системы. В результате время валидации СТРИ сокращается в среднем на 37,72%. Также рассмотрены схематические изменения, усложняющие процесс ратификации СТРИ.

Разработан программный инструмент, который включает в себя среду валидации и позволяет пользователю генерировать любое количество примеров валидации и исследовать ошибки. Программное обеспечение может использоваться как инженерами по валидации, так и инженерами-разработчиками.

