KHARKOV NATIONAL UNIVERSITY OF RADIOELECTRONICS

Proceedings of IEEE East-West Design & Test Symposium (EWDTS'09)

SPONSORED BY

IEEE Computer Society Test Technology Technical Council

Moscow, Russia, September 18 – 21, 2009

IEEE EAST-WEST DESIGN AND TEST SYMPOSIUM 2009
ORGANISING COMMITTEE

General Chairs
V. Hahanov – Ukraine
Y. Zorian – USA

General Vice-Chairs
D. Bikov - Russia
R. Ubar – Estonia

Program Chairs
S. Shoukourian – Armenia
D. Speranskiy – Russia

Program Vice-Chairs
M. Renovell – France
Z. Navabi – Iran

Steering Committee
M. Bondarenko – Ukraine
V. Hahanov – Ukraine
R. Ubar – Estonia
Y. Zorian – USA

Publicity Chairs
R.Ubar - Estonia
S. Mosin – Russia

Program Committee
E. Evdokimov – Ukraine
A. Chaterjee – USA
E. Gramatova – Slovakia
S. Hellebrand – Germany
A. Ivanov – Canada
M. Karavay – Russia
V. Kharchenko – Ukraine
K. Kuchukjan – Armenia
A. Matrosova – Russia
V. Melikyan - Armenia
O. Novak – Czech Republic
A. Orailoglu – USA
Z. Peng – Sweden
A. Petrenko – Ukraine
P. Prinetto – Italy
J. Raik – Estonia
A. Romankevich – Ukraine
A. Ryjov – Russia
R. Seinauskas – Lithuania
S. Sharshunov – Russia
A. Singh – USA
J. Skobtsov – Ukraine
A. Stempkovsky – Russia
V. Tverdokhlebov – Russia
V. Vardanian – Armenia
V. Yarmolik – Byelorussia
E. J. Aas – Norway
J. Abraham – USA
M. Adamski – Poland
A . Barkalov – Poland
R. Bazylevych – Ukraine
V. Djigan – Russia
A. Drozd – Ukraine
W. Kuzmicz – Poland

Organizing Committee
S. Chumachenko – Ukraine
N. Kulbakova – Ukraine
V. Obrizan – Ukraine
A. Kamkin – Russia
K.Petrosyanz – Russia
A.Sokolov – Russia
Y.Gubenko – Russia
M.Chupilko – Russia
E. Litvinova – Ukraine
O. Guz – Ukraine
G. Markosyan – Armenia

EWDTS CONTACT INFORMATION

Prof. Vladimir Hahanov
Design Automation Department
Kharkov National University of Radio Electronics,
14 Lenin ave,
Kharkov, 61166, Ukraine.

Tel.: +380 (57)-702-13-26
E-mail: hahanov@kture.kharkov.ua
Web: www.ewdtest.com/conf/

7th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS 2009)

Moscow, Russia, September 18-21, 2009

The main target of the **IEEE East-West Design & Test Symposium** (EWDTS) is to exchange experiences between scientists and technologies of Eastern and Western Europe, as well as North America and other parts of the world, in the field of design, design automation and test of electronic circuits and systems. The symposium is typically held in countries around the Black Sea, the Baltic Sea and Central Asia region. We cordially invite you to participate and submit your contribution(s) to EWDTS'09 which covers (but is not limited to) the following topics:

- Analog, Mixed-Signal and RF Test
- Analysis and Optimization
- ATPG and High-Level Test
- Built-In Self Test
- Debug and Diagnosis
- Defect/Fault Tolerance and Reliability
- Design for Testability
- Design Verification and Validation
- EDA Tools for Design and Test
- Embedded Software Performance
- Failure Analysis, Defect and Fault
- FPGA Test
- HDL in test and test languages
- High-level Synthesis
- High-Performance Networks and Systems on a Chip
- Low-power Design
- Memory and Processor Test
- Modeling & Fault Simulation
- Network-on-Chip Design & Test
- Modeling and Synthesis of Embedded Systems
- Object-Oriented System Specification and Design
- On-Line Testing
- Power Issues in Design & Test
- Real Time Embedded Systems
- Reliability of Digital Systems
- Scan-Based Techniques
- Self-Repair and Reconfigurable Architectures
- System Level Modeling, Simulation & Test Generation
- System-in-Package and 3D Design & Test
- Using UML for Embedded System Specification
- CAD and EDA Tools, Methods and Algorithms
- Design and Process Engineering
- Logic, Schematic and System Synthesis
- Place and Route
- Thermal, Timing and Electrostatic Analysis of SoCs and Systems on Board
- Wireless and RFID Systems Synthesis
- Digital Satellite Television
- Signal and Information Processing in RF and Communication

The symposium is organized by Kharkov National University of Radio Electronics, in cooperation with Tallinn University of Technology, Institute for System Programming of RAS, and Moscow Institute of Electronics and Mathematics. It is sponsored by the IEEE Computer Society Test Technology Technical Council (TTTC) and financially supported by Cadence, JTAG Technologies, Kaspersky Lab, Synopsys, Mentor Graphics, Tallinn Technical University, Donetsk Institute of Road Transport, Moscow Institute of Electronics and Mathematics, Virage Logic, Echostar, Aldec, Teprocomp, DataArt Lab.

CONTENTS

Technology for Faulty Blocks Coverage by Spares

Hahanov Vladimir, Chumachenko Svetlana, Litvinova Eugenia,
Zakharchenko Oleg, Kulbakova Natalka
Kharkov National University of Radioelectronics
Ukraine, 61166, Kharkov, Lenine Ave, 14, of. 321, tel/fax: (8-057)-70-21-326
E-mail: hahanov@kture.kharkov.ua, ri@kture.kharkov.ua, kiu@kture.kharkov.ua

Abstract

The technology for the minimum coverage of faulty blocks by spares when repairing the logic part of digital system-on-chip is proposed. The general provisions and rules of coverage for the matrix of configurable logic blocks (CLB) with faulty cells are considered. Coverage criteria for faulty cells are developed. Examples of the algorithm implementation are made.

1. Introduction

Billions of digital systems-on-chips, used in the world, containing up to 16 types of various components (processor, memory, logic, buses, dedicated computers), which can be divided into 2 subsets: the memory (90%) and logic (10%). At that faults, detected in memory, are repaired successfully by the on-chip facilities of the leading companies (Virage Logic, Intel).

But almost 10% of logic is unamenable to regular solutions in the on-chip repair. Today, the world's biggest problem in the market of electronic technology is repairing the logic part of digital system-on-chip.

Due to high market appeal the problem of diagnosis and repair of memory and logic cells is considered in the paper. It is one of the Gartner's Top 10 Strategic Technologies for 2009 that is solved by readdressing faulty cells to faultless components from the spare rows, columns and tiles.

The strategy works on the logic blocks, which must be addressed (and should be provided with repair blocks) or reprogrammable on the faultless space of the chip for the embedded repair. Repair models for SiP memory modules are considered in the papers [1-6].

It should also take into account that the level of sales of computers has fallen in the 2 quarter of 2009 by 8% and amounted to 66 million pieces, but sale of laptops has grown by 20%. With regard to market of chips, there is the highest rise of sales in the last 13 years.

This fact confirms Moore's Law – transistor today is worthless, a user will pay for power. The whole world sees the future of digital systems-on-chips. Conclusion – all market-based ideas will be implemented in the chip with a dedicated functionality. So, Infrastructure IP creation in a chip is important problem, because it is capable to realize the embedded diagnosis and repairing, which will significantly improve the yield and extend the life cycle of digital product. Therefore, any new solution in this area might be interesting for the market of electronic technology, which determines the urgency of the proposed technology for quasi-optimal faulty blocks coverage by spare components.

The papers [7-9] are devoted to the development of the theory and methods for optimization of geometric design, in particular, the research of optimization placement problem for rectangular objects. The optimization placement problem for rectangular objects with variable metric characterizations in a given area is considered in [7, 8]. The analysis of advanced technologies for embedded Functional Intellectual Property of digital system-in-package is shown in [10]. Features of the architecture «System-in-Package» and present repair strategies for digital systems-on-chips, as well as the method of evaluation the reliability of their performance are considered. The problem of SoC testing technologies adaptation to new digital system embodiment System-in-Package (SiP) that allows implementing on-chip sophisticated specialty computers and RF devices is considered in [10]. System-in-Package forms new objectives and goals of Infrastructure IP for real time SiP functionality, which differ from embedded SoC diagnosis essentially. Structure-logical diagnosis and repair methods for FPGA functional logic blocks based on real time fault detection table analysis are proposed. A fault coverage method for digital system-on-chip by means of traversal the logic block matrix to repair the FPGA components is proposed in [10]. A method enables to obtain the solution in the form of quasioptimal coverage for all faulty blocks by minimum number of spare tiles. A choice one of two traversal strategies for rows or columns of a logic

block matrix on the basis of the structurization criteria, which determine a number of faulty blocks, reduced to the unit modified matrix of rows or columns is realized.

The objective of the research is the development of technology for the optimal faulty blocks coverage by spares when repairing the logic of digital SoC.

Research tasks are:

1) The development of generalities and rules to cover the matrix of configurable logic blocks with faulty cells.

2) The development of coverage criteria for faulty cells.

3) Flowcharting for the bypassing the matrix of configurable logic blocks to obtain coverage.

4) Flowchart examples.

2. Generalities and rules of coverage

A matrix of configurable logic blocks with marked faulty cells is considered. In the appropriate adjacency matrix the identifier 1 answers to a fault. In detecting the faulty cell its coverage is carried out. Covering element is a block of 9 cells, which form a square of size 3×3.

Methods cover the faulty cell. Coverage of the faulty cell, which is regarded as a base, can be performed by 9 ways (Fig. 1).

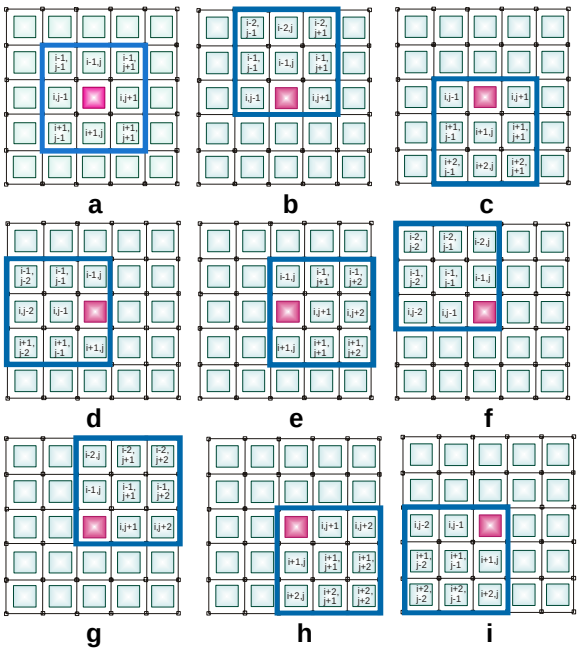

Figure 1. Coverage methods for faulty cell a_{ij}

The alternate coverages are represented in Fig. 1:

a – $a_{ij} \cup a_{i,j-1} \cup a_{i,j+1} \cup a_{i-1,j-1} \cup a_{i-1,j} \cup$
$\cup a_{i-1,j+1} \cup a_{i+1,j-1} \cup a_{i+1,j} \cup a_{i+1,j+1}$;

b – $a_{ij} \cup a_{i,j-1} \cup a_{i,j+1} \cup a_{i-1,j-1} \cup a_{i-1,j} \cup$
$\cup a_{i-1,j+1} \cup a_{i-2,j-1} \cup a_{i-2,j} \cup a_{i-2,j+1}$;

c – $a_{ij} \cup a_{i,j-1} \cup a_{i,j+1} \cup a_{i+1,j-1} \cup a_{i+1,j} \cup$
$\cup a_{i+1,j+1} \cup a_{i+2,j-1} \cup a_{i+2,j} \cup a_{i+2,j+1}$;

d – $a_{ij} \cup a_{i,j-1} \cup a_{i,j-2} \cup a_{i-1,j} \cup a_{i-1,j-1} \cup$
$\cup a_{i-1,j-2} \cup a_{i+1,j} \cup a_{i+1,j-1} \cup a_{i+1,j-2}$;

e – $a_{ij} \cup a_{i,j+1} \cup a_{i,j+2} \cup a_{i-1,j} \cup a_{i-1,j+1} \cup$
$\cup a_{i-1,j+2} \cup a_{i+1,j} \cup a_{i+1,j+1} \cup a_{i+1,j+2}$;

f – $a_{ij} \cup a_{i,j-1} \cup a_{i,j-2} \cup a_{i-1,j} \cup a_{i-1,j-1} \cup$
$\cup a_{i-1,j-2} \cup a_{i-2,j} \cup a_{i-2,j-1} \cup a_{i-2,j-2}$;

g – $a_{ij} \cup a_{i,j+1} \cup a_{i,j+2} \cup a_{i-1,j} \cup a_{i-1,j+1} \cup$
$\cup a_{i-1,j+2} \cup a_{i-2,j} \cup a_{i-2,j+1} \cup a_{i-2,j+2}$;

h – $a_{ij} \cup a_{i,j+1} \cup a_{i,j+2} \cup a_{i+1,j} \cup a_{i+1,j+1} \cup$
$\cup a_{i+1,j+2} \cup a_{i+2,j} \cup a_{i+2,j+1} \cup a_{i+2,j+2}$;

i – $a_{ij} \cup a_{i,j-1} \cup a_{i,j-2} \cup a_{i+1,j} \cup a_{i+1,j-1} \cup$
$\cup a_{i+1,j-2} \cup a_{i+2,j} \cup a_{i+2,j-1} \cup a_{i+2,j-2}$.

Coverage constrains for the faulty cell. If a faulty cell is located in first/last two rows/columns, obviously, the number of ways to cover for it is limited to the alternate solutions shown in Fig. 2.

3. Criteria for choice and obtainment of coverage

When choosing a covering element the square with the largest weight is preferred. Weight is determined by the number of faulty cells, which are in the covering square. For example, the coverage solutions for the faulty cell a_{ij}, when choice of the covering element with maximum weight is possible, are shown in Fig. 3.

Row-wise matrix traversal. In implementing the coverage algorithm the traversal of a matrix by rows is performed starting from the first one in order from left to right. Then a jump to the second row and traversal of the matrix elements in reverse order are carried out. Thus, the matrix rows with odd numbers are always traversed from the left to right, then jump to the bottom row with an even number and traversal in the reverse order from the right to left are performed.

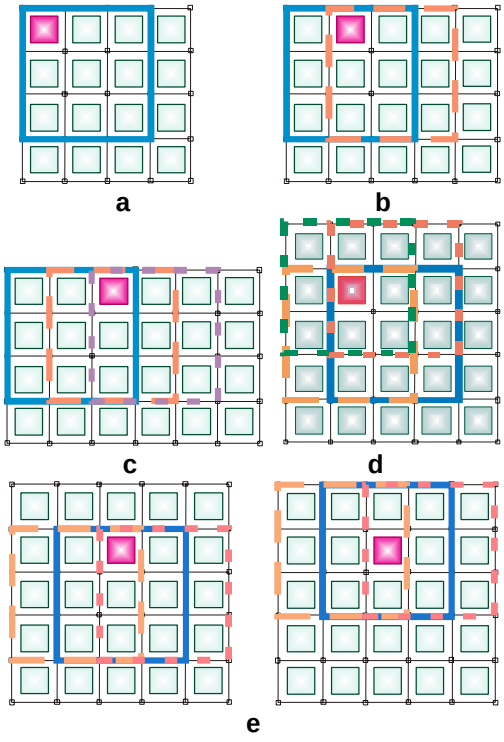

Figure 2. Coverage constraints for faulty cell, when it is located in two extreme rows/columns: a – the only possible way of obtainment the coverage, when a fault is in the corner cells $a_{11}, a_{1,N}, a_{N,1}, a_{N,N}$; b – two coverage ways, when a fault is in the cells a_{12}, a_{21}, $a_{1,N-1}$, $a_{N-1,1}$, $a_{N-1,N}$, $a_{N,N-1}$, $a_{2,N}$, $a_{N,2}$; c – three coverage ways for fault, when it is in the cells a_{13}, a_{31}, $a_{1,N-2}$, $a_{N-2,1}$, $a_{3,N}$, $a_{N,3}$, $a_{N,N-2}$, $a_{N-2,N}$; d – four coverage ways, when a fault is in the cells a_{22}, $a_{2,N-1}$, $a_{N-1,2}$, $a_{N-1,N-1}$; e – six coverage ways, when a fault is in the cells a_{23}, a_{32}, $a_{2,N-2}$, $a_{N-2,2}$, $a_{3,N-1}$, $a_{N-1,3}$, $a_{N-1,N-2}$, $a_{N-2,N-1}$

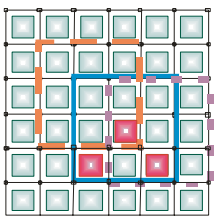

Figure 3. Choice of covering element – a square with the highest weight (solid line)

For the faulty cell, found in a row, the covering square with maximum weight is chosen. After that, traversal of a row to choose the coverage squares for the rest faulty cells is continued.

When traversing the next row a part of faulty cells is already covered, so it is necessary to choose the covering squares for uncovered faulty cells. At that earlier obtained coverage limits the choice of covering squares for the rest faulty cells that reduces their number in each case and reduces the search time. In the case where there are several ways to cover the faulty cells by squares with the same maximum weight, any of them is chosen.

If there is a situation where it is impossible to cover the faulty cell, it should be back to the previous faulty one to change its coverage on a square with smaller weight.

When searching a new coverage a square with less weight is preferred if it covers two neighboring cells (located on the diagonal or side by side in a row/column) and it is adjacent to the left/right side (when traversing from left to right or right to left respectively) in order to retain a larger number of free cells on the right/left, respectively, for other coverage. In other words, a covering square is moved in the direction already seen to this time rows/columns.

Columnwise matrix traversal. Along with traversing by rows the pass of the matrix from top to bottom followed by jump to the right of the neighboring column and then up can be used. In this case the odd-numbered columns are traversed from top to bottom, but even – from left to right. At that coverage is selected similarly to previous one, as well as subject to the shift of covering square to the side of traversed cells.

Example 1. Consider a matrix of configurable logic blocks with marked faulty cells presented in Fig. 4.

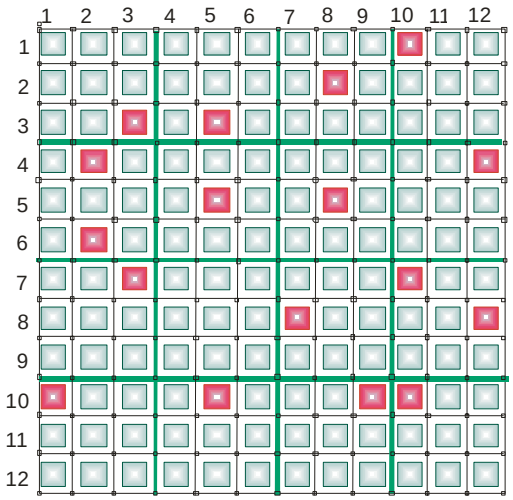

Figure 4. Matrix of logic blocks with faulty cells

According to the algorithm, the traversal of a matrix is performed by rows.

For any faulty element the maximum weight of its neighborhood is determined and grouping of cells in clusters is performed subject to the above rules.

Pass of the row with number $i=1$ indicates that the evaluation of neighborhood weight for the faulty element $a_{1,10}$ is equal to $w(a_{1,10})=2$. Then the faulty cells $a_{1,10}$ and a_{28} should be joined in a single covering block: $a_{18} \cup a_{19} \cup a_{1,10} \cup a_{28} \cup a_{29} \cup a_{2,10} \cup \cup a_{38} \cup a_{39} \cup a_{3,10}$. When traversing the row $i=3$ one can see that for the cell a_{33} the coverage $a_{33} \cup a_{34} \cup a_{35} \cup a_{43} \cup a_{44} \cup a_{45} \cup a_{53} \cup a_{54} \cup a_{55}$ with maximum weight $w(a_{33})=3$ can be chosen. But after traversal the row $i=4$ from right to left would be that the faulty cell a_{42} can not be covered by any of the square (Fig. 5). In this case it is necessary to return to the previous cell, which was covered. If there are no alternate solutions for changing its coverage, it is necessary to return to the previous cell and change its coverage. This is the cell a_{33} in Fig. 5.

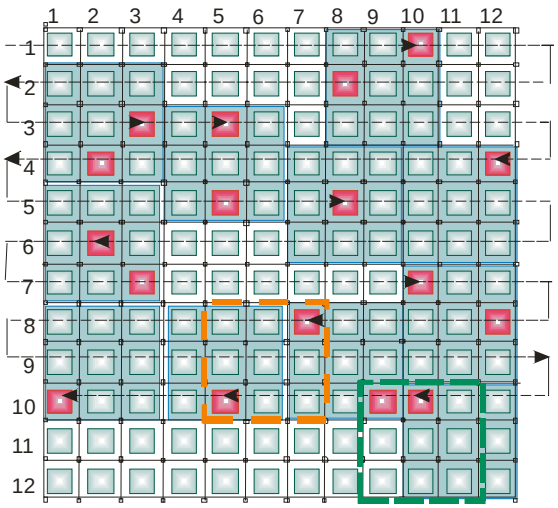

Figure 5. Choice of coverage at row-wise traversal

As a result, the coverage consisting of 11 squares is obtained, at that 6 squares have the weight 2 and 5 ones have the weight 1. The adjacent faulty cells $a_{10,9}$ and $a_{10,10}$ can be covered by the element (Fig. 5):

$$a_{10,9} \cup a_{10,10} \cup a_{10,11} \cup a_{11,9} \cup a_{11,10} \cup \cup a_{11,11} \cup a_{12,9} \cup a_{12,10} \cup a_{12,11}$$

or

$$a_{10,8} \cup a_{10,9} \cup a_{10,10} \cup a_{11,8} \cup a_{11,9} \cup \cup a_{11,10} \cup a_{12,8} \cup a_{12,9} \cup a_{12,10}.$$

In this case if to change coverage of the faulty cell a_{87} on the square $a_{8,5} \cup a_{8,6} \cup a_{8,7} \cup a_{9,5} \cup a_{9,6} \cup a_{9,7} \cup a_{10,5} \cup a_{10,6} \cup a_{10,7}$, the coverage of 10 squares will be obtained, from which 7 squares have weight 2 and 3 ones – weight 1 (Fig. 6). That coverage is obtained at row-wise matrix traversal (Fig. 6).

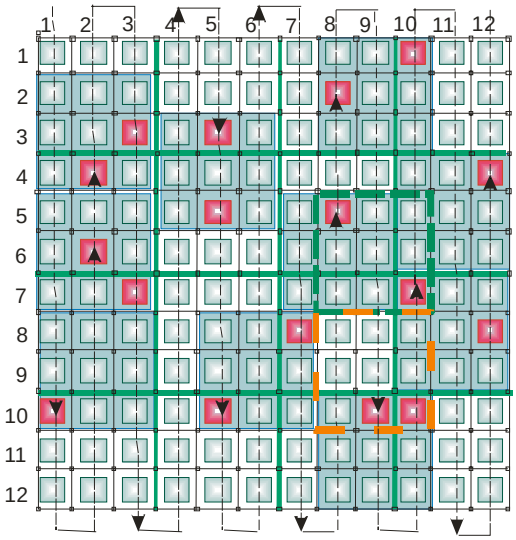

Figure 6. Optimal coverage for a matrix of configurable logic blocks

Thus, quasi-optimal coverage was obtained at traversing the matrix by rows (as it is shown in Example 1) and optimal one – by columns. As it will be shown below, the traversal of a matrix by rows almost always gives optimal coverage.

4. Description of the algorithm

1. Enter the matrix $A=[a_{ij}]_1^n$.

2. A loop by rows (for i=1 up to N).

2.1. If row number is odd i=2k+1, go to item 3 (a loop by columns).

2.2. If row number is even i=2k, go to item 4 (a loop by columns – column traversal in reverse order from N up to 1).

3. A loop by columns (for j=1 up to N).

4. A loop by columns (for j=N up to 1).

4.1. Calculate the weight of adjacent squares with respect to the base cell (call procedure 1/procedure 1 return).

4.2. Choose square with a maximum weight as covering one.

4.3. Eliminate from consideration the cells, forming a covering square (call procedure 2/procedure 2 return).

4.4. Return to item 4 (continue of the traversal by columns).

4.5. Return to item 2 (go to nest row).

5. Output results – optimal (quasi-optimal) coverage as a set covering squares, each of which is a union of 9 cells.

PROCEDURE 1 evaluates the weight of adjacent squares with respect to the base cell.

PROCEDURE 2 eliminates from consideration the cells, forming a covering square, but it memorizes them in order to return, if it is required.

5. Examples of the algorithm implementation

Consider the implementation of the algorithm on examples where the logic blocks matrices with faulty elements are generated randomly.

Example 2. Consider a matrix of configurable logic blocks with marked faulty cells presented in Fig. 7.

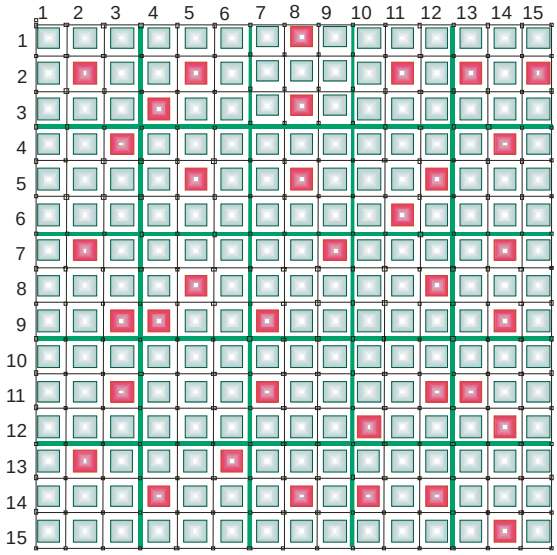

Figure 7. Matrix of logic blocks with faulty cells

In order to construct coverage the row-wise matrix traversal is carried out according to the rules, criteria and algorithms. The route of matrix traversal is shown in Fig. 8. The arrows on the faulty cells indicate on which basic cell the coverage is carried out.

Optimal matrix coverage, presented in Fig. 8, consists of 17 blocks: 3 ones have weight 1, 9 – weight 2 and 5 – weight 3.

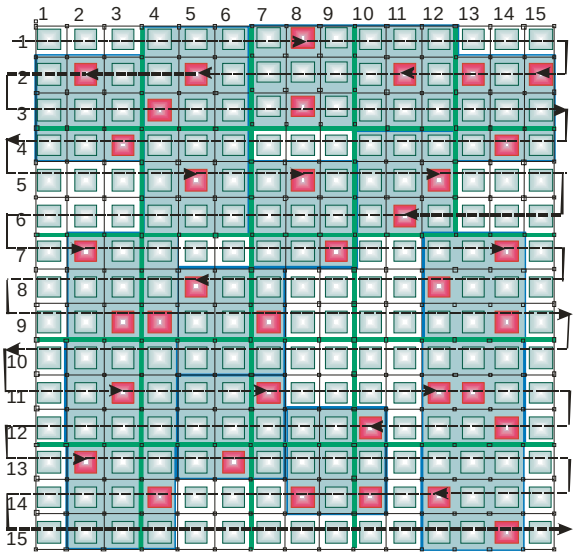

Figure 8. Route of matrix traversal

Example 3. Consider the alternate coverage solutions for a matrix of configurable logic blocks 15×15 with faulty cells. The coverage obtained by row-wise traversal of a matrix is shown in Fig. 9. It consists of 16 squares: 4 ones have the weight 1, 7 – weight 2, 2 – weight 3, 3 – weight 4. The coverage obtained by columnwise matrix traversal from left to right is shown in Fig. 10. The number of covering elements is equal to 16: 4 ones have the weight 1, 7 – weight 2, 2 – weight 3, 3 – weight 4. Thus, two different ways of traversal (by rows and columns) are quantitatively the same result.

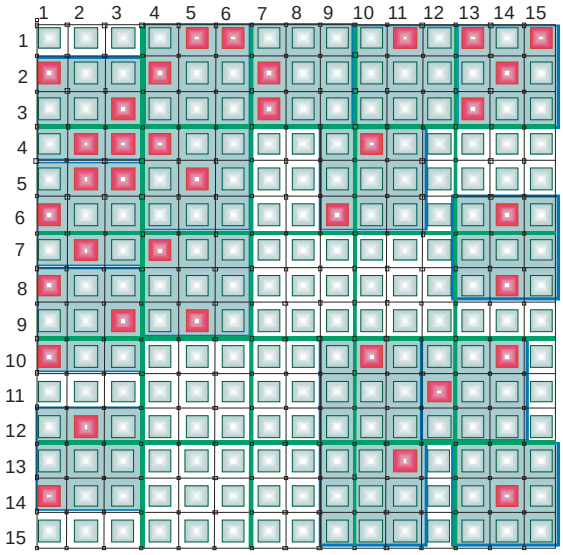

Figure 9. Matrix coverage for example 3

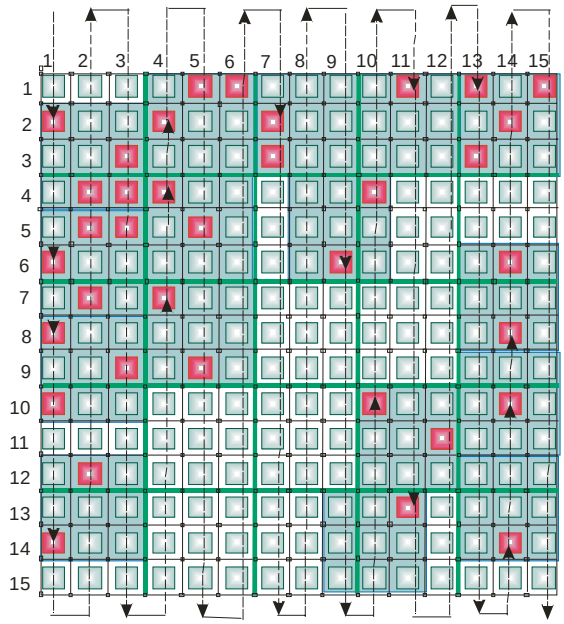

Figure 10. Columnwise traversal for example 3

Example 4. Consider the coverage for a matrix of configurable logic blocks 15×15 with marked faulty cells. To construct coverage the row-wise traversal is carried out (Fig. 11). The coverages, which were considered but were not chosen, are shown in Fig. 11 by dotted lines. For example, to cover the faulty cell a_{19} when traversing the first row the block $a_{17} \cup a_{18} \cup a_{19} \cup a_{27} \cup a_{28} \cup a_{29} \cup a_{37} \cup a_{38} \cup a_{39}$ is preferred among two blocks with weight 2 (not block $a_{18} \cup a_{19} \cup a_{1,10} \cup a_{28} \cup a_{29} \cup a_{2,10} \cup a_{38} \cup a_{39} \cup a_{3,10}$); first one joins some cells which were traversed in the first row (shift to the left on the route of row traversal).

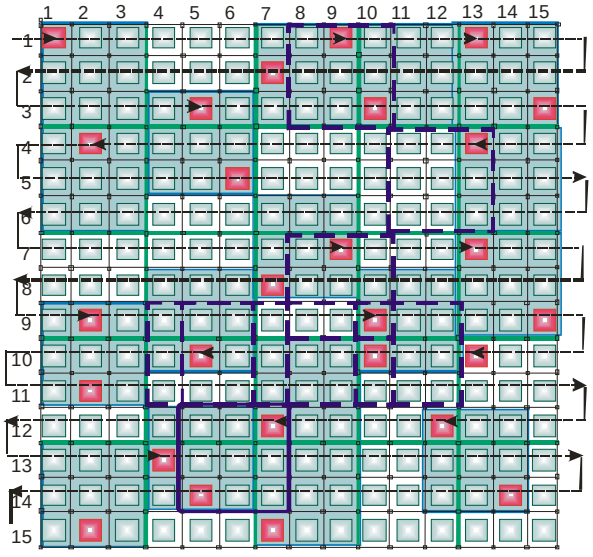

Figure 11. Row-wise traversal for example 4

When covering the cell $a_{4,13}$ the block $a_{4,13} \cup a_{4,14} \cup a_{4,15} \cup a_{5,13} \cup a_{5,14} \cup a_{5,15} \cup a_{6,13} \cup \cup a_{6,14} \cup a_{6,15}$ with the weight 1 is selected (not $a_{4,11} \cup a_{4,12} \cup a_{4,13} \cup a_{5,11} \cup a_{5,12} \cup a_{5,13} \cup a_{6,11} \cup \cup a_{6,12} \cup a_{6,13}$ with the same weight), because the traversing of the fourth row is performed from right to left and the first block includes the already traversed cells, it means that the block is shifted in the direction of the cells have already passed – to the right on the route.

Thus, the optimal coverage of the matrix presented in Fig. 11, consists of 17 squares: 8 ones have the weight 1, 9 – weight 2.

6. Conclusion

Quasi-optimal coverage method for the addressable faulty cells of digital systems-on-chips by spares is developed. It enables to increase yield of FPGA.

Intercomparison with analogs. Existing analogs are typically focused on optimizing the placement of components in 3D or 2D space. The difference of the proposed method is coverage optimization for faulty cells by spares. Therefore, the objective function is minimization of the number of spares, which completely cover all faulty addressable cells of the chip.

Scientific novelty. Formulation of the proposed method is original. A feature of a quasi-optimal coverage algorithm, implemented in the software, is traversal of all matrix cells by rows (columns) to determine and choose the optimal solution of the coverage problem. The choice of covering elements with the maximum weight and constructed before coverages constraint the number of covering squares for the rest faulty cells, which reduces the search time.

Practical significance of the proposed method is the possibility of applying the method to on-chip repair components of digital systems-on-chips, including the addressable memory cells or addressable logic blocks. In general, this method can be applied for repairing any addressable components, located in the plane.

Prospects for research are implementation of the technology for repair of digital systems-on-chips to enhance the reliability and yield of products, operating in critical environments such as space, aviation.

7. References

[1] *Memory* Repair Primer – A guide to understanding embedded memory Repair options and issues. Logic Vision. 2007.

[2] *Youngs L., Paramanandam S.* Mapping and Repairing Embedded-Memory Defects//IEEE Design and Test. 1997. P. 18–24.

[3] *Zorian Y., Shoukourian S.* Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield // Design and Test. 2003. P. 58–66.
[4] *Huang R., Chen Ch., Wu Ch.* Economic Aspects of Memory Built-in Self-Repair // IEEE Design & Test. 2007. P. 164–172.
[5] *Choi M., Park N., Lombardi F., Kim Y. B., Piuri V.* Optimal Spare Utilization in Repairable and Reliable Memory Cores // 2003 International Workshop on Memory Technology, Design and Testing (MTDT'03). 2003. P. 64–71.
[6] *Ohler Ph., Hellebrand S., Wunderlich H.-J.* An Integrated Built-In Test and Repair Approach for Memories with 2D Redundancy // 12th IEEE European Test Symposium (ETS'07). 2007. P. 91–96.
[7] *Novozshilova M.V., Chub I.A, Murin M.N.* Method of solving the problem of rectangles placement with variable metric characteristics // Radioelectronics and informatics. 2008. №4. P. 134-141.
[8] *Chub I.A, Novozshilova M.V.* Modification of the exact method for solving the problem of rectangular objects placement // Automated Control Systems and A D. 2008. Issue. 145. P.57-63.
[9] *Stoyan Yu.G., Yakovlev S.V.* Mathematical models and optimization methods for geometric design. K.: Naukova dumka, 1986. 266 p.
[10] *Litvinova E.I.* Embedded technologies of SiP components repair // Automated Control Systems and A D. 2008. Issue. 145. P. 40-48.

Camera-ready was prepared in Kharkov National University of Radio Electronics
by Dr. Svetlana Chumachenko
Lenin ave, 14, KNURE, Kharkov, 61166, Ukraine

Approved for publication: 31.08.2009. Format 60×841/8.
Relative printer's sheets: . Circulation: 150 copies.
Published by SPD FL Stepanov V.V.
Ukraine, 61168, Kharkov, Ak. Pavlova st., 311

Матеріали симпозіуму «Схід-Захід Проектування та Діагностування – 2009»
Макет підготовлено у Харківському національному університеті радіоелектроніки
Редактори: Володимир Хаханов, Світлана Чумаченко
Пр. Леніна, 14, ХНУРЕ, Харків, 61166, Україна

Підписано до публікації: 31.08.2009. Формат 60×84$^{1}/_{8}$.
Умов. друк. арк. . Тираж: 150 прим.
Видано: СПД ФЛ Степанов В.В.
Вул. Ак. Павлова, 311, Харків, 61168, Україна