Optimization of Memory Faults Coverage by Spares

V. Hahanov, E. Litvinova, K. Mostovaya

Computer Engineering Faculty, Kharkov National University of Radioelectronics, Lenin Ave. 14, Kharkov, Ukraine, 61166, phone: (057) 70-21-421, (057) 70-21-326, e-mail: hahanov@ktek.kharkov.ua; kiu@ktek.kharkov.ua

Introduction

Modern tendencies of semiconductor industry development involve permanent decrease of silicon chip area, increase of quantity and various elements on area unit. Embedded memory elements make up the bulk of components, placed on a chip. Now there are hundreds and even thousands kinds of memory, which are used in Systems on Chip (SoC) [1-4]. As illustrated on Fig. 1 increase of memory share on chip will bring about its complete dominance for data and program storage over 5 years and towards the 2014 year this characteristic will amount to 94% [2]. It will provide not only high performance of functionality realization but also flexibility that is inherent in software by design error correction. Feature of memory elements is that which in the manufacturing and operation processes of them some cells under the influence of failures can go out of functioning state. But given circumstance not always brings memory element to critical or extreme state, when repair is impossible. At that it is considered such technical state of object, when total quantity of faulty cells does not exceed reserve repair resources of memory.

Aim of given research is memory repair by means of cover task solution of faulty cells by minimal quantity of reserved components. At that memory module is considered as matrix of elementary cells. Chip development designer uses two kinds of memory: functional cells, used for data and program storage directly when module use in SoC, as well as reserve or duplicate elements designed for memory repair in case of failure of functional cells. Functional and reserve cells are combined in columns and rows. At detection of defect a row (a column) containing defective element is disconnected from functional memory structure and a row (a column) from crystal reserve is connected on its position (Fig. 2).

As a rule testing procedure is realized by BIST-block (Built-In Self Test) that is hardware fast-operating generator of tests, as well as memory outputs reactions analyzer (signature) applied for test sequences.

Repair analysis consists in determination faulty memory elements cover ability by reserve components. At that memory module is considered as matrix of elementary cells. Chip development designer uses two kinds of memory: functional cells, used for data and program storage directly when module use in SoC, as well as reserve or duplicate elements designed for memory repair in case of failure of functional cells. Functional and reserve cells are combined in columns and rows. At detection of defect a row (a column) containing defective element is disconnected from functional memory structure and a row (a column) from crystal reserve is connected on its position (Fig. 2). But as quantity of reserved components is limited, it is
necessary special procedure enabling to allocate additional resources efficiently to provide cover of defective memory elements by minimal quantity of reserved rows and columns.

![Fig. 2. Memory repair by means of reserved elements: a – memory with defective cells and reserved elements; b – nonoptimal cover; c – optimal cover](image)

Described above cover search procedure of defective cells by minimal quantity of reserved rows and columns can be realized as built-in repair module or external one with respect to chip. In second case data about errors income from without, they are processed and transmitted to controller that provides memory repair. It results in considerable time waste. So, as a rule, preference is given to on-chip module realization, when data about defects are transmitted from BIST directly. This procedure is called as Built-In Repair Analysis (BIRA). Memory repair is realized by means of disconnection defective elements (rows and columns) by electrical fuse that involves faulty rows and columns disconnection and connection of reserved ones. Fuse can be electrical or laser. [9]. Electrical fuse device has smaller dimension then laser one and it is used more frequently. Fuse is carried out by means of instruction set, which can store in permanent memory inside chip (hard repair) or in random-access memory (soft repair). Soft repair has several advantages: at defect appearance new corrected instruction can be recorded in memory easily; there provide economic use of chip area and sufficient reliability [4]. As a rule instructions are not stored in chip permanently. So they are additionally recorded in memory with respect to chip or they are generated automatically. Now combined method of instruction storage is used. Hard repair enables to use simplified manufacturing test and provides detection of errors, which can not be fixed by soft repair under certain circumstances (for instance, overheating). Set of processes of built-in memory repair analysis and its direct repair (usually soft repair) is called as Built-In Self Repair (BISR) [8] that is represented on Fig. 3.

![Fig. 3. Built-In Self Repair process](image)

Model of built-in self repair process is represented below.

1. Activization of chip. Filling of BISR register by zero values.
2. Running the BIST controller. Memory testing and information reservation about defective cells in BIRA register.
3. Information transmission about defective cells in BISR register for further use.
4. Scanning BIRA registers, which have repair status, by BIST controller in order to get information about defects.
5. Running fuse controller in write mode and transmission repair instructions from BISR.
6. Restarting the chip. Writing fuse information in BISR register, replacing defective rows and columns by reserved components.
7. Running the BIST controller. Memory retesting for to check repair result.

Aim function of given research on the assumption of modern achievements in the on-the-fly memory repair field can be defined in the following way: minimization of memory repair cost (hardware cost) on-stream SoC by means of use cover optimization method of defective memory cells set by system of reserved components under limitation conditions N on component quantity:

$$Z = \min_i Q_i(F) \left| \begin{array}{c} Q_i(F) \in N_{\text{max}} = N_r + N_c \end{array} \right.,$$

where $Q_i(F)$ – cost of i-th solution variant of memory $M = \left[ M_{ij} \right]$ repair, by minimal rows and columns subset $R = \{ R_r, R_c \}$ of reserved chip elements that covers set $F$ of defective memory cells $R \cap F = F$, $Z^* = \max |F_i|, F_i \in F \leftarrow \forall R_i$.

The general idea of defective cells cover by reserved rows and columns can be represented by the following strategy:

1. $M_{i(j)} \subseteq R \leftarrow \max \left( M_{i(j)} \right) \wedge \min \left( M_{i(j)} \right) \cup R$;
2. $M = M \setminus M_{i(j)}$;
3. $N_{r(c)} = N_{r(c)} - 1$;
4. $(M^w) = \emptyset \wedge (N_{r(c)} \geq 0) \rightarrow$ is solution.

Here $M_{i(j)}^w$ – cost of row (column) that is equal to quantity of defective cells in it; $N_{r(c)}$ – quantity of reserved rows (columns).

Represented above four items are repeated iteratively up to gaining of result. Fixation of value one in a parameter $N_{r(c)} \leq 0$ is evidence of solution absence – quantity of reserved elements (rows or columns) is not enough for covering of all defective cells.

Flow chart of built-in self repair process based on the strategy described above is shown on Fig. 4.
Memory repair method by means of reserved elements

Memory repair problem is reduced to covering of finite defective cells by means of reserved element system $R = \{R^r, R^c\}$, where $R^r$ and $R^c$ – subsets of reserved rows and columns ($|R^r| = N_r$, $|R^c| = N_c$) of memory matrix, which can be used for it repair under operating maintenance, at that $R = R^r \cup R^c$. Set of all possible solutions of covering problem is reduced to variant family $R = \{R_1, \ldots, R_k\}$, every of which is represented by subsets of rows and columns $R_i = \{R_i^r, R_i^c\}$. Memory matrix $M = [M_{ij}]$ can contain failures, which are on intersection of a row and a column $F_{ij} = M_i \cap M_j$, where $F_{ij}$ – failure identifier.

Metric or selection criterion of minimal complete cover is defined in the following way: every element $F_{ij} \in F$ can be covered by a single component or two ones from family $\{R_i^r, R_i^c\}$, but at that set $N_F$ of defective cells can be covered by nonadditive row and column subset power $|R_i^r| + |R_i^c| \leq |F|$. In this case functional corresponding to criterion of solution minimality of covering problem is represented as follows:

$$Z_c = \min_{i \leq N_F} \left| R_i^r \cup R_i^c \right| \leftarrow (\forall F_{ij} \in F) \rightarrow$$

$$\Rightarrow \exists i, j (F_{ij} \cap [R_i^r \cup R_i^c] = F_{ij})].$$  

(2)

Model of minimal cover getting process

Base of model of minimal cover getting process of defective cells set by collection of reserved rows and columns is greedy algorithm:

1. Rating of rows and columns of matrix $M = [M_{ij}]$ by quantity of defective cells and assignment of cost to every component (row and column):

$$M_i^w = \left( M_{ij}^w \right); M_j^w = \left( M_{ij}^w \right).$$  

(3)

2. On every step it is realized choice of row $M_i$ (column $M_j$) with maximal cost $M_i^w$ ($M_j^w$) and minimal quantity of intersections with elements $M_i$ ($M_j$), chose earlier, for which reserved row $R_i^r$ (column $R_j^c$) is chosen. Then covered rows (columns) are excepted from consideration by cost nulling $M_i^w$ ($M_j^w$).

3. Solution exists when next condition is true: $\{R_i^r, R_j^c\} \cap F = F$. Otherwise defective cells set can not be covered by given set of reserved rows and columns and there not exists a solution.

Then to minimize the covering it is carried out analysis and correction of a solution:

4. Fixation of last solution element: a row $M_i$ (a column $M_j$).

5. Backward scan of solution elements, beginning at the number $(Z_c - 1)$. Check, if removal of given row $M_i$ (column $M_j$) from solution reduces to appearance of uncovered faulty elements of the matrix $M = [M_{ij}]$. If «No», it means that given row $M_i$ (column $M_j$) is redundant element and it can be removed. The process is repeated for all solution elements.
Computational complexity of memory repair process model depends on dimension of memory matrix \((n \times n)\), quantity of defective elements \((2n – 1)\), identification of defective rows and columns) and power of available reserve of rows and columns \((2n – 1)\) in the limit:

\[
Q = n^2 + 2n + 2n + 2n + 2n = n^2 + 8n. \tag{4}
\]

Flow chart of model of minimal cover getting process is represented on Fig. 5.

Verification of cover process model

To verification of model of minimal cover getting process hereinafter an instance of task solution for memory matrix with ten defective cells, two reserved rows and five reserved column is considered (Fig. 6). Every reserved component (row or column) is able to repair from one up to \(5\) defective cells, which belong to row or column.

![Fig. 6. Memory matrix with defective elements](image)

A set \(F\) of cover elements has the appearance:

\[
F = \{(F_{2,5});(F_{2,10});(F_{3,3});(F_{3,8});(F_{5,4});(F_{5,7});(F_{5,10});(F_{7,3});(F_{8,7});(F_{8,10})\}. \tag{5}
\]

Rows and columns of the matrix \(M\), ranged by quantity of defective cells:

\[
M^r_{[3,4,5,7,8,10]} = \{(M_{3,3},M_{7,3})^2;(M_{5,4})^1;(M_{2,5})^3;\]
\[
(M_{5,7},M_{8,7})^2;(M_{3,8})^1;(M_{2,10},M_{5,10},M_{8,10})^3\};
\]

\[
M^c_{[2,3,5,7,8]} = \{(M_{2,5},M_{2,10})^2;(M_{3,3},M_{3,8})^2;\]
\[
(M_{5,4},M_{5,7},M_{5,10})^3;(M_{7,3})^1;(M_{8,7},M_{8,10})^2\}. \tag{6}
\]

In the sets \(M^r_{[3,4,5,7,8,10]}\) and \(M^c_{[2,3,5,7,8]}\) there are two elements with the same cost that is equal to \(w = 3\): a row of the memory matrix \(\{(M_{2,10},M_{5,10},M_{8,10})^3\}\) in \(M^r_{[3,4,5,7,8,10]}\) and a column \(\{(M_{5,4},M_{5,7},M_{5,10})^3\}\) in \(M^c_{[2,3,5,7,8]}\). By random choice the column \(\{(M_{5,4},M_{5,7},M_{5,10})^3\}\) is defined for covering by reserved component (Fig. 7). Chose column is excepted from consideration by cost nulling.

From residual rows and columns of the matrix \(M\) with nonzero cost

\[
M^r_{[3,4,5,7,8,10]} = \{(M_{3,3},M_{7,3})^2;(M_{5,4})^1;(M_{2,5})^3;\]
\[
(M_{5,7},M_{8,7})^2;(M_{3,8})^1;(M_{2,10},M_{5,10},M_{8,10})^3\};
\]

\[
M^c_{[2,3,5,7,8]} = \{(M_{2,5},M_{2,10})^2;(M_{3,3},M_{3,8})^2;\]
\[
(M_{5,4},M_{5,7},M_{5,10})^3;(M_{7,3})^1;(M_{8,7},M_{8,10})^2\} \tag{7}
\]

the row \(\{(M_{2,10},M_{5,10},M_{8,10})^3\}\) that has minimal quantity of intersections with chose above column is chosen to cover by reserved element (Fig. 8). Used row is excepted from consideration by cost nulling.

\[
M^r_{[3,4,5,7,8,10]} = \{(M_{3,3},M_{7,3})^2;(M_{5,4})^1;(M_{2,5})^3;\]
\[
(M_{5,7},M_{8,7})^2;(M_{3,8})^1;(M_{2,10},M_{5,10},M_{8,10})^0\}. \tag{8}
\]

![Fig. 7. Covering of defective cells \{(5,4);(5,7);(5,10)\} by reserved column](image)

![Fig. 8. Covering of defective cells \{(2,10);(5,10);(8,10)\} by reserved row](image)
Fig. 9. Covering of defective cell \(\{2,5\},\{2,10\}\) by reserved column

From residual rows and columns of the matrix \(M\) with nonzero cost to cover of defective cells the column \(\{(M_{2,5},M_{2,10})\}^2\) having minimal quantity of intersections with used above reserved elements is chosen (Fig. 9). Chose column is excepted from consideration by cost nulling.

\[
M_{\{2,5,5,7,8\}}^c = \{(M_{2,5},M_{2,10})^0; (M_{3,3},M_{3,8})^0; (M_{5,4},M_{5,7},M_{5,10})^0; (M_{7,3})^1; (M_{8,7},M_{8,10})^0\}. \tag{9}
\]

In a similar covering of faulty cells in the columns \(\{(M_{3,3},M_{3,8})^2\}\) and \(\{(M_{8,7},M_{8,10})^2\}\) is carried out (Fig. 10).

Fig. 10. Covering of defective cells \(\{(3,3),(3,8)\}\) and \(\{(8,7),(8,10)\}\) by reserved columns

Appropriate cost values are nulled:

\[
M_{\{2,3,5,7,8\}}^c = \{(M_{2,5},M_{2,10})^0; (M_{3,3},M_{3,8})^0; (M_{5,4},M_{5,7},M_{5,10})^0; (M_{7,3})^1; (M_{8,7},M_{8,10})^0\}. \tag{10}
\]

To cover last faulty cell the column \(\{(M_{7,3})^3\}\) is chosen (Fig. 11).

Fig. 11. Covering of defective cell \(\{(7,3)\}\) by reserved column

\[
M_{\{2,3,5,7,8\}}^c = \{(M_{2,5},M_{2,10})^0; (M_{3,3},M_{3,8})^0; (M_{5,4},M_{5,7},M_{5,10})^0; (M_{7,3})^3; (M_{8,7},M_{8,10})^0\}. \tag{11}
\]

Then last solution element – the column \(\{(M_{7,3})^3\}\) is fixed. For the rest rows and columns of a covering removal analysis is realized as it is described below.

Removal of the column \(\{(M_{8,7},M_{8,10})^2\}\) from solution will reduce to appearance uncovered faulty cell \(\{(8,7)\}\). Cells \(\{(3,3),(3,8)\}\) are covered by a single solution element – the column \(\{(M_{3,3},M_{3,8})^2\}\), so given column can not be removed. Removal of the column \(\{(M_{2,5},M_{2,10})^2\}\) will reduce to appearance uncovered faulty cell \(\{(2,5)\}\). The row \(\{(M_{2,10},M_{5,10},M_{8,10})^3\}\) enables to eliminate faulty cells \(\{(2,10),(5,10),(8,10)\}\) only, which are covered by the columns \(\{(M_{2,5},M_{2,10})^2\}, \{(M_{5,4},M_{5,7},M_{5,10})^3\}, \{(M_{8,7},M_{8,10})^2\}\) too. Removal of it will not reduce to truth violation of the condition \(R^c \cap F = F\), so given row is redundant solution element and it can be removed (Fig. 12).

Fig. 12. Removal of the redundant solution element – the row \(\{(M_{2,10},M_{5,10},M_{8,10})^3\}\)
Removal of the column \( \{(M_{5,4}, M_{5,7}, M_{5,10})\} \) will reduce to appearance uncovered faulty cells \( \{(5,4);(5,7);(5,10)\} \).

Therefore actions, carried out in compliance with process model (see Fig.5), enabled to get minimal solution – cover of ten defective cells of memory module that has two reserved rows and five reserved columns by five elements \( R_6 = \{(M_{7,5}, M_{7,10})^2; (M_{4,3}, M_{4,8})^2; (M_{5,4}, M_{5,7}, M_{5,10})^3; (M_{7,3})^3; (M_{8,7}, M_{8,10})^2\}. \) (12)

In comparison with approach that is suggested in [10] given method enables to find optimal solution using less (in two times) quantity of steps.

Conclusions

Scientific novelty of the research is design of new quasioptimal memory repair method by means of solution of defective cells set covering problem by reserved elements. The method has quadratic computational complexity and can have built-in a chip realization or realized outside a chip in the form of additional defect correction module that enables to carry out memory repair automatically.

Practical importance of the research is essential (5-10%) yield increasing on electronic technology market by means of repair of defective memory chips, as well as life cycle time increasing of ones by memory repair on basis of practical realization offered method.

Further research will be directed on design of hardware memory test and repair module at appearance of defects on chip production and operation stages.

References


Memory repair method by means of spares that enables to cover faulty cells by minimal quantity of reserved components was represented. It enables to increase yield rate (by 5-10 %) by means of repair of faulty silicon crystals on the manufacturing stage, as well as to increase life cycle time of silicon crystals during operation phase. Ill. 12, bibl. 10 (in English; summaries in English, Russian and Lithuanian).


Представлен метод восстановления работоспособности модулей памяти с помощью резервных элементов, позволяющий покрыть множество дефектных ячеек минимальным возможным количеством избыточных компонентов. Метод позволяет существенно (на 5–10%) повысить процент выхода годной продукции путем восстановления работоспособности дефектных кристаллов памяти на стадии производства, а также увеличить длительность жизненного цикла кристаллов в процессе эксплуатации. Ил. 12, библ. 10 (на английском языке; рефераты на английском, русском и литовском яз.).


Pateiktas metodas, skirtas atminties modulų darbingumu atkurti naudojant rezervinius elementus. Tokie elementai leidžia ištaisyti defektines laštelese naudojant minimalų perteklinių elementų kiekį. Metodas leidžia gerokai (5–10%) padidinti tinkamos produkcijos išeigą, nes defektinių atminties kristalų darbingumas atkuriamas gamybos ciklo metu. Taip pat padidinama kristalų gyvavimo trukmė eksploatacijos metu. Il. 12, bibl. 10 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).