Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 3, 329–339
DOI: 10.2478/v10006-008-0030-y

ADDRESS SEQUENCES AND BACKGROUNDS WITH DIFFERENT HAMMING DISTANCES FOR MULTIPLE RUN MARCH TESTS

SVETLANA YARMOLIK

Belarusian State University of Computer Science and Radioelectronics
P. Brovki 6, 220013 Minsk, Belarus
e-mail:svetlana10ru@yahoo.com

It is widely known that pattern sensitive faults are the most difficult faults to detect during the RAM testing process. One of the techniques which can be used for effective detection of this kind of faults is the multi-background test technique. According to this technique, multiple-run memory test execution is done. In this case, to achieve a high fault coverage, the structure of the consecutive memory backgrounds and the address sequence are very important. This paper defines requirements which have to be taken into account in the background and address sequence selection process. A set of backgrounds which satisfied those requirements guarantee us to achieve a very high fault coverage for multi-background memory testing.

Keywords: random-access memory (RAM), memory testing, March memory test, neighbourhood pattern sensitive faults, memory address, memory background, Gray code, Hamming distance.

1. Introduction

The rapid development in semiconductor technology has led to larger and dense semiconductor memories on a single chip. As more and more memory cells are packed into a single chip, the number of failure modes increases and the need for efficient algorithms to detect faults in them becomes more critical. One of the most difficult fault diagnoses problems is the problem of the detection of *neighbourhood pattern sensitive faults* (NPSFs) (Goor, 1991; Suk and Reddy, 1980; Cheng *et al.*, 2002). The neighbourhood pattern sensitive fault model is not new, but it is still widely discussed in the literature and is becoming more and more important for memory testing. As has been shown earlier (Hayes, 1975), the attempts to detect unrestricted pattern sensitive faults in large semiconductor random-access memories (RAMs) based on classical memory tests are impractical. However, by taking into consideration new solutions and new approaches (Hayes, 1980; Franklin and Saluja, 1996; Cockburn, 1995; Yarmolik *et al.*, 1998) mostly derived on the basis of transparent memory testing, it appears to be possible to achieve a high fault coverage even for unrestricted NPSFs. In this paper we use the unrestricted neighbourhood pattern sensitive fault model and consider only a random-access memory with $N = 2^m$ bits, m being a positive integer. Furthermore, it is assumed that the RAMs are 1 bit wide, i.e., only one bit of information is read or written into the memory at a time.

Some approaches to detect NPSFs, such as the tiling method (Goor, 1991; Hayes, 1975), the two-group method (Goor, 1991; Hayes, 1980), the row-March algorithm (Franklin and Saluja, 1996), and a multi-background method (Cockburn, 1995; Yarmolik *et al.*, 1998), have been proposed. The new publications deal with reduction in the costs of memory testing (Bernardi *et al.*, 2006; Bernardi *et al.*, 2005), fault detection by output response comparison of identical circuits using half-frequency compatible sequences (Pomeranz and Reddy, 2006), transparent memory testing (Li, 2007). Traditional March algorithms (Goor, 1991) have been widely used in memory testing because of their linear time complexity, high fault coverage, and ease in built-in self-test (BIST) implementation. It is known that traditional March algorithms do not generate all neighbourhood patterns that are required for testing NPSFs. However, as has been shown in previous publications (Cockburn, 1995; Yarmolik *et al.*, 1998; Yarmolik and Yarmolik, 2006b; Yarmolik and Yarmolik, 2006a; Sokol and Yarmolik, 2006; Yarmolik and Sokol, 2006), classical March tests can be modified based on multiple address orders and multiple data backgrounds to increase the NPSFs fault coverage.

A well-known property of March tests is that for one run memory test execution there are no specific requirements for the address order as well as for memory background (Goor, 1991). For any address order and memory background, the number of detectable memory faults including the NPSF will be the same and can be calculated according to the memory test detection ability (Niggemeyer *et al.*, 2000; Niggemeyer *et al.*, 1998). In the case of multiple-run memory test execution, the consecutive memory address order and background are very important for the achievement of a high fault coverage (Yarmolik and Yarmolik, 2006b). The high efficiency of this memory testing is obtained due to the detection of an additional portion of the complex memory faults, first of all NPSFs. Any new run of the same memory test should be done with new initial conditions, namely, with a new memory background or an address order, or both the background and the address order. Let us concentrate on the efficient sequences of the address order and backgrounds for multiple memory test runs.

2. Fault models

Memory faults can be divided on the basis of the number of cells being faulty into one-cell faults (e.g., stuck-at faults, transition faults) and multiple-cell faults (e.g., coupling faults) (Goor, 1991). The latter are more difficult to detect. The general case of a fault belonging to the second group is the NPSF (Goor, 1991). A pattern-sensitive fault is a multicell coupling fault. It occurs when the content of a memory cell, or the ability to change the cell content, is influenced by a certain pattern of other cells in the memory (Goor, 1991). As has been shown in numerous publications, to consider all possible patterns of all memory cells is both impractical and unnecessary.

An NPSF is a special case of the general multi-cell coupling fault, wherein the coupling cells are the neighbourhood of the coupled cell. In general, the coupled cell is called the *base cell* and the coupling cells are called the *deleted neighbourhood cells*. The base cell and the deleted neighbourhood cells together are called the *neighbourhood cells*. The three-cell NPSF (NPSF3) five-cell NPSF (NPSF5) and nine-cell NPSF (NPSF9) are shown in Fig. 1 and have been regarded as the most often used models (Goor, 1991; Suk and Reddy, 1980; Cheng *et al.*, 2002). They will be considered in the paper.

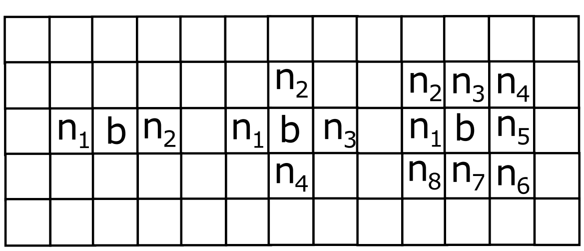

Fig. 1. Three-cell, five-cell and nine-cell NPSFs.

An NPSF, e.g., NPSF5, includes the base cell *(b)* as well as the deleted neighbourhood cells *(n1, n2, n3, n4)*. This fault model can be further categorized into three subtypes of faults as follows (Goor, 1991; Cheng *et al.*, 2002).

A *static NPSF* (SNPSF) occurs if the base cell is forced to a certain state due to the appearance of a certain pattern in the deleted neighbourhood. To detect SNPSF3 and SNPSF5, 8 SNPSF3 and 32 SNPSF5, static neighbourhood patterns must be applied and the generation of these patterns by the test algorithm must be verified.

A *passive NPSF* (PNPSF) occurs if the base cell cannot change its state from 0 to 1 or from 1 to 0 due to the appearance of a certain pattern in the deleted neighbourhood. To detect PNPSF3 and PNPSF5, 8 PNPSF3 and 32 PNPSF5, static neighbourhood patterns must be applied and the generation of these patterns within neighbourhood cells by the test algorithm must be verified.

An *active NPSF* (ANPSF) occurs if the base cell is forced to a certain state when a transition occurs in a deleted neighbourhood cell, while other deleted neighbourhood cells assume a certain pattern. To detect ANPSF3 and ANPSF5, 16 ANPSF3 and 128 ANPSF5, static neighbourhood patterns must be applied and the generation of these patterns by the test algorithm must be verified.

As has been shown in previous publications (Goor, 1991; Suk and Reddy, 1980; Cheng *et al.*, 2002; Hayes, 1975; Hayes, 1980; Franklin and Saluja, 1996), the detection of NPSF*k* depends on the abilities of the memory test to generate all 2^{k-1} possible patterns within the deleted neighbourhood cells for both state transitions in the base cell and to verify the fault-free or faulty base cell content. These conditions allow us to obtain a precise fault coverage of the March test to detect a PNPSF and an SNPSF, as well as to estimate the efficiency for the ANPSF.

We concentrate our attention on the PNPSF as the most difficult fault to be detected. First of all, it should be emphasized that due to scrambling information, as well as specific optimization techniques, there is a huge amount of such faults that should be consider. Any arbitrary k memory cells out of all N memory cells can be involved into PNPSF*k*. One out of k cells is the base cell. For the deleted neighbourhood pattern there are 2^{k-1} different patterns and there are two states for the base cell. Then the exact number of PNPSF*k* is determined according to the equation (Yarmolik and Sokol, 2006)

$$L(PNPSFk) = 2k2^{k-1}\binom{n}{k}. \tag{1}$$

It is quite important to emphasize that there is an enormous amount of such faults due to the random locations of the cells involved into the fault.

3. March test efficiency analyses

March tests are superior in terms of the test time and the simplicity of hardware implementation, and they consist of sequences of March elements. The March element includes sequences of read/write (r/w) operations, which are all applied to a given cell before proceeding to the next cell. The way of moving to the next cell is determined by the address sequence order. During the testing, March tests use address sequences called "up" and "down" sequences, denoted by $\Uparrow$ and $\Downarrow$, respectively. The notation $\Updownarrow$ means "don't care the direction of address order". The address sequences do not necessarily have to be counting sequences.

Consider the well-known March tests such as MATS+ (Goor, 1991), March C− (Goor, 1991; Suk and Reddy, 1980; Cheng *et al.*, 2002) and March PS(23N) (Yarmolik *et al.*, 1998), which can be represented for the random background $B = b_0b_1b_2 \ldots b_{N-2}b_{N-1}$ as follows:

$$
\begin{gathered}
MATS+\\
\{\Updownarrow(wb); \Uparrow(rb, w\bar{b}); \Downarrow(r\bar{b}, wb)\}\\[1em]
March\ C-\\
\{\Updownarrow(wb); \Uparrow(rb, w\bar{b}); \Uparrow(r\bar{b}, wb);\\
\Downarrow(rb, w\bar{b}); \Downarrow(r\bar{b}, wb); \Updownarrow(wb)\},\\[1em]
March\ PS(23)\\
\{\Updownarrow(wa);\\
\Uparrow(rb, w\bar{b}, r\bar{b}, wb, rb, w\bar{b}); \Uparrow(r\bar{b}, wb, rb, w\bar{b}, r\bar{b});\\
\Downarrow(r\bar{b}, wb, rb, w\bar{b}, r\bar{b}, wb); \Downarrow(rb, w\bar{b}, r\bar{b}, wb, rb)\},
\end{gathered}
\tag{2}
$$

where $b \in \{0, 1\}$ and $\bar{b}$ is an inverse value compared with b.

To investigate the memory March tests, suppose that PNPSF*k* includes memory cells with the increasing order of addresses $\alpha(0)$, $\alpha(1)$, $\alpha(2), \ldots, \alpha(k-1)$ and the base cell has the address $\alpha(i)$, where $0 \leq i \leq k-1$. Then, due to the consecutive access to the memory cells during the March test, there are four possible patterns within the deleted neighbourhood cells:

$$
\begin{array}{ll}
1 & \bar{b}_{\alpha(0)}, \bar{b}_{\alpha(1)}, \ldots, \bar{b}_{\alpha(i-1)}, b_{\alpha(i+1)}, \ldots, b_{\alpha(k-1)}\\
2 & b_{\alpha(0)} b_{\alpha(1)}, \ldots, b_{\alpha(i-1)}, \bar{b}_{\alpha(i+1)}, \ldots, \bar{b}_{\alpha(k-1)}\\
3 & \bar{b}_{\alpha(0)}, \bar{b}_{\alpha(1)}, \ldots, \bar{b}_{\alpha(i-1)}, \bar{b}_{\alpha(i+1)}, \ldots, \bar{b}_{\alpha(k-1)}\\
4 & b_{\alpha(0)} b_{\alpha(1)}, \ldots, b_{\alpha(i-1)}, b_{\alpha(i+1)}, \ldots, b_{\alpha(k-1)}.
\end{array}
\tag{3}
$$

The base cell has two possible transitions from state 0 to state 1 (↑) and from state 1 to state 0 (↓). From this it can be concluded that there are eight possible patterns within neighbouring memory cells, which can be used as the definition of eight possible PNPSF*k* fault types (Cheng *et al.*, 2001) and detected on the basis of memory March testing. For the case of all zero background $B = b_0b_1b_2 \ldots b_{N-2}b_{N-1} = 000 \ldots 00$, the eight types of PNPSF*k* are shown in Table 1.

Table 1. Types of PNPSF*k*.

Type	$\alpha(0)$	…	α(i-1)	α(i)	α(i+1)	…	α(k-1)
#1	1	…	1	↑	0	…	0
#2	1	…	1	↓	0	…	0
#3	0	…	0	↑	1	…	1
#4	0	…	0	↓	1	…	1
#5	1	…	1	↑	1	…	1
#6	1	…	1	↓	1	…	1
#7	0	…	0	↑	0	…	0
#8	0	…	0	↓	0	…	0

It should be noted that for every type of PNPSF*k* there are *k* subtypes of PNPSF*k* depending on the position of the base cell. For example, in the case of PNPSF5 for the type #1 there are five subtypes ↑0000, 1↑000, 11↑00, 111↑0 and 1111↑ of PNPSF5 detectable via March testing. The entire set of all subtypes of PNPSF3 and PNPSF5 is shown in Tables 2 and 3.

Table 2. PNPSF3 types.

Type	PNPSF*3*
#1	↑00, 1↑0, 11↑
#2	↓00, 1↓0, 11↓
#3	↑11, 0↑1, 00↑
#4	↓11, 0↓1, 00↓
#5	↑11, 1↑1, 11↑
#6	↓11, 1↓10, 11↓
#7	↑00, 0↑0, 00↑
#8	↓00, 0↓0, 00↓

Table 3. PNPSF5 types.

Type	PNPSF5
#1	↑0000, 1↑000, 11↑00,111↑0, 1111↑
#2	↓0000, 1↓000, 11↓00, 111↓0, 1111↓
#3	↑1111, 0↑111, 00↑11, 000↑1, 0000↑
#4	↓1111, 0↓111, 00↓11, 000↓1, 0000↓
#5	↑1111, 1↑111, 11↑11, 111↑1, 1111↑
#6	↓1111, 1↓111, 11↓11, 111↓1, 1111↓
#7	↑0000, 0↑000, 00↑00, 000↑0, 0000↑
#8	↓0000, 0↓000, 00↓00, 000↓0, 0000↓

A brief analysis of the PNPSF*k* shown in Tables 1–3 allows us to make the conclusion that the maximal number of PNPSF*k* that can be detected via one run March testing can be estimated as $8k-8$. All sets of PNPSFk detected by the March testing twice as representatives of two different classes include eight faults: $\uparrow 000\ldots00$, $\downarrow 000\ldots00$, $\uparrow 111\ldots11$, $\downarrow 111\ldots11$, $000\ldots00 \uparrow$, $000\ldots00 \downarrow$, $111\ldots11 \uparrow$, $111\ldots11 \downarrow$.

Then the maximal fault coverage which can be achieved through one-run March memory testing can be calculated as

$$FC_{MAX} = \frac{8k-8}{k2^k}100\% = \frac{k-1}{k2^{k-3}}100\%. \quad (4)$$

Some of the March tests allow us to get the maximal value of the fault coverage. Among those there are March PS(23N) (Yarmolik *et al.*, 1998), March 17N (Cheng *et al.*, 2001) and March 18N:

$$\begin{aligned}&\{\Updownarrow (wb);\\ &\Uparrow (rb, w\bar{b}, r\bar{b}, wb); \Downarrow (rb, w\bar{b}); \Uparrow (r\bar{b}, wb, rb, w\bar{b});\\ &\qquad \Uparrow (r\bar{b}, wb); \Uparrow (rb, w\bar{b}); \Downarrow (r\bar{b}, wb, rb)\}.\end{aligned}$$

The above tests activate and detect all detectable PNPSFk during the sequential access to memory cells (see Table 1). It should be noted that for one-run March testing it is impossible to get a high fault coverage. Only in the cases of known memory topology (Yarmolik *et al.*, 1998; Cheng *et al.*, 2001), multi-background and multi-address orders (Yarmolik and Yarmolik, 2006b; Yarmolik and Yarmolik, 2006a; Sokol and Yarmolik, 2006; Yarmolik and Sokol, 2006) is it possible to increase these values.

A sufficiently low value of fault coverage can be obtained for MATS+ type March tests, which allows us to detect only one type out of the eight types of PNPSFk. For the zero background it is only type #1 of PNPSFk (see Table 1). Then

$$FC_{MATS+} = \frac{k}{k2^k}100\% = \frac{1}{2^k}100\%. \quad (5)$$

A slightly high fault coverage can be achieved by March C$-$ type March tests due to the detection ability of four types of PNPSFk,

$$FC_{MarchC-} = \frac{4k}{k2^k}100\% = \frac{1}{2^{k-2}}100\%. \quad (6)$$

The exact values of FC_{MAX} for different k and fault coverages for the tests shown in (2) are presented in Table 4.

To summarize the above results, it can be concluded that one-run March testing has restricted abilities to detect PNPSFk. The only solution to increase the efficiency of March testing for the case of unrestricted PNPSFk (for the case of an unknown memory topology due to the data scrambling) is the application of multi-run memory testing (Yarmolik and Yarmolik, 2006b) with different address sequences or memory backgrounds.

4. Multi-run memory testing

The efficiency of multi-run memory testing strictly depends on the appropriate initial conditions for every consecutive March memory test execution. To achieve a high fault coverage, it is quite important to choose an optimal address order and memory background. As was shown in (Yarmolik and Yarmolik, 2006a) even for the same address sequence for all March test executions by applying different initial memory addresses for consecutive test runs an increasing sequence of PNPSFk coverages was obtained. Optimal seeds were selected and an algorithm for their generation was presented (Yarmolik and Yarmolik, 2006a; Yarmolik and Sokol, 2006). For the selection of the address order, an arithmetic distance was proposed as the numeric metric for the optimal address sequence selection (Yarmolik and Yarmolik, 2006b; Sokol and Yarmolik, 2006). The key idea of the proposed solutions is based on the generation of sufficiently different address sequences, which allow us to generate different neighbourhood patterns for the same memory background. Some attempts were made for restricted PNPSFk in (Cheng *et al.*, 2002; Yarmolik *et al.*, 1998; Cheng *et al.*, 2001). As has been shown for the case of known topology it is possible to achieve the 100% fault coverage for PNPSF5 due to the background selection on the basis of complete information about logical memory addresses and the location of memory cells. For unrestricted PNPSFk the optimal address order and memory background are still open issues.

To make a conclusion about the effectiveness of applying different memory address sequences and memory backgrounds to achieve a high fault coverage, let us use as the metric the so-called Hamming distance. The Hamming distance $HD[A(i), A(j)]$ between two binary vectors $A(i)$ and $A(j)$ is calculated as a weight $w[A(i) \oplus A(j)]$ of the vector $A(i) \oplus A(j)$.

To estimate the effectiveness of the sequences of binary vectors as the representative metric, let us use an average Hamming distance $AHD[A(i), A(i+1)]$ between consecutive binary patterns $A(i)$ and $A(i+1)$ (a sequence of memory addresses or backgrounds). This characteristic will be calculated as

$$AHD[A(i), A(i+1)] = \frac{1}{2^m - 1}\sum_{i=0}^{2^m-2} HD[A(i), A(i+1)]. \quad (7)$$

Suppose that a binary vector $A(i) \in \{0, 1, 2, E, 2^m - 1\}$ represents the *i*-th memory address generated according to some algorithm (a counter sequence, a Gray code, an M-sequence, etc.) and each address consists of *m* bits. Also, $A(i)$ can represent the memory background as the contents of $N = 2^m$ memory binary cells.

Table 4. PNPSF*k* fault coverage.

k	3	4	5	6	7	8	9
FC_{MAX}	66.6	37.5	20.0	10.4	5.1	2.7	1.4
$FC_{MarchPS(23N)}$	66.6	37.5	20.0	10.4	5.1	2.7	1.4
$FC_{MarchC-}$	50.0	25.0	12.5	6.25	3.1	1.5	0.7
FC_{Mats+}	12.5	6.25	3.12	1.56	0.8	0.4	0.2

The key idea of this paper is based on the Hamming distance metric. The idea which is presented in this paper is the following: For two consecutive address sequences, namely, $A1(i)$ and $A2(i)$, for two-run memory March testing, the fault coverage of PNPSF*k* will be high in the case when the average Hamming distance (7) $AHD[A2(i), A2(i+1)]$ is sufficiently higher compared with $AHD[A1(i), A1(i+1)]$. It is predicted that in this case the new set of PNPSF*k* will be detected as a result of consecutive accesses to the memory cells during the March test according to the different address sequence $A2(i)$. As a result, compared with (3), four new possible patterns within the deleted neighbourhood cells will be generated during the second memory test run.

4.1. Multi-run memory testing with different address sequences.

At the beginning, as the address sequence algorithm let us use the Gray code (Gray, 1958). An *m*-bit Gray code $G0$ lists all the binary m-bit patterns (codewords) $A_{G0}(i) = a_{m-1}a_{m-2}\dots a_1a_0$, $i \in \{0, 1, 2, \dots, 2^m - 1\}$ so that consecutive patterns differ in only one bit (Gray, 1958; Savage, 1997). In a cyclic code, the first and last patterns differ also in one bit. Moreover, the Gray codes can be viewed as *Hamiltonian paths* on the hypercube graph and cyclic codes correspond to *Hamiltonian cycles*. The transition sequence $t(A_{G0}) = (t_1, t_2, t_3, \dots, t_{N-1})$ of an m-bit Gray code enumerates the bit positions $t_l \in \{m-1, m-2, \dots, 2, 1, 0\}$, where $A_{G0}(i)$ and $A_{G0}(i+1)$ differ and $N = 2^m$. When $G0$ is cyclic, its closing transition t_N is the position where $A_{G0}(2^m - 1)$ and $A_{G0}(0)$ differ (Gilbert, 1958). For example, when $m = 3$, the Gray code $G0$ has the form $000, 001, 011, 010, 110, 111, 101, 100$, and the corresponding transition sequence $t(A_{G0}) = 0, 1, 0, 2, 0, 1, 0$. As can be seen, in this case the transition sequence consists of the sequences of index $t_l \in \{2, 1, 0\}$ for consecutive binary patterns $a_2a_1a_0$. It is obvious that the transition sequence $t(A_{G0})$ which determines any Gray code should satisfy the following statement proposed by Gilbert (1958).

Statement 1. *The transition sequence* $t(A_{G0})$=$(t_1, t_2, t_3, \dots, t_{N-1})$*, where* $t_l \in \{m-1, m-2, E, 2, 1, 0\}$ *and* $N = 2^m$*, generates an* m*-bit Gray code if and only if every contiguous subsequence* $t_{k+1}, t_{k+2}, t_{k+3}, \dots, t_{k+r}$ *for any* $r \in \{2, 3, 4, E, 2^{m-1}\}$ *consecutive* m*-bit words* $A_{G0}(k+1)$, $A_{G0}(k+2)$, $A_{G0}(k+3), \dots, A_{G0}(k+r)$ *contains some element of* $t_l \in \{m-1, m-2, \dots, 2, 1, 0\}$ *an odd number of times.*

Proof. It is easy to see that if for any $r < 2^m$ the values of the transition sequence $t_{k+1}, t_{k+2}, t_{k+3}, \dots, t_{k+r}$ do not contain any element from $m-1, m-2, \dots, 2, 1, 0$ an odd number of times, the sequence will contain at least two identical words $A_{G0}(k)$ and $A_{G0}(k+r)$. So in this case the Gray code sequence will not contain all possible *m*-bit words and the Gray code sequence will be not generated. ■

As an example, consider the *reflected Gray code* for $m = 4$. The sequences of binary m-bits codes consist of all possible combinations 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000, generated according to the transition sequence $t(A_{G0}) = (0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0)$. As has been mentioned above, the Gray code can be characterized as sequences in which two consecutive words differ only in one bit. This means that these words have the minimal value of the Hamming distance (Gray, 1958; Savage, 1997; Gilbert, 1958).

It is easy to show that the proposed characteristic (the average Hamming distance) for the original Gray code $A(i) \in \{0, 1, 2, E, 2^m - 1\}$ equals 1 ($AHD[A(i), A(i+1)] = 1$) (Savage, 1997). Let us try to modify the Gray code in such a way that the metric is changed drastically to get the high degree of differences between the original sequences with $AHD[A(i), A(i+1)] = 1$ and the new one. For this purpose, new sequences with a sufficiently high average Hamming difference should be generated or, equivalently, with characteristic $AHD[A(i), A(i+1)]$ very close to m where m is the size of the binary vector $A(i)$.

To get address sequences with a maximum average Hamming distance, let use the transition sequence of the Gray code $t(A_{G0}) = (t_1, t_2, t_3, \dots, t_{N-1})$ (Gilbert, 1958). It can be converted to a non-transition sequence $t^*(A_{G0}) = (t_1^*, t_2^*, t_3^*, \dots, t_{N-1}^*)$, where $t_l^* \in \{m-1, m-2, E, 2, 1, 0\}$ is the index of the unchangeable bit in two m-bit consecutive words. In this case, in two m-bit consecutive words $m-1$ bits are inverted. This means that the Hamming distance between two consecutive m-bit words is $m-1$. Let us define such sequences as *anti-Gray* sequences. In Table 5 the anti-Gray codes for $m = 2$, $m = 3$ and $m = 4$ and corresponding non-transition sequences are shown.

Table 5. Anti-Gray sequences.

i	$m=2$	$t^*(A_{G0})$	$m=3$	$t^*(A_{G0})$	$m=4$	$t^*(A_{G0})$
0	00		000		0000	
1	10	0	110	0	1110	0
2	11	1	011	1	0011	1
3	01	0	101	0	1101	0
4			110	2	0110	2
5			000	0	1000	0
6			101	1	0101	1
7			011	0	1011	0
8					1100	3
9					0010	0
10					1111	1
11					0001	0
12					1010	2
13					0100	0
14					1001	1
15					0111	0

For $m = 4$ and $m = 2$ all possible m-bit combinations are generated. So the anti-Gray sequence with the maximum Hamming distance is generated. For $m = 3$ only four binary patterns have been generated, namely, 000, 011, 101 and 110. So, the anti-Gray sequence for $m = 3$ does not cover all possible m-bit words. Let us define the conditions of anti-Gray sequence generation as the next statement.

Statement 2. *For even numbers of m the non-transition sequence $t^*(A_{G0}) = (t_1^*, t_2^*, t_3^*, \ldots, t_{N-1}^*)$, where $t_l^* \in \{m-1, m-2, \ldots, 2, 1, 0\}$ and $N = 2^m$, generates an anti-Gray sequence of 2^m m-bit words with the Hamming distance between two consecutive words equal to $m-1$.*

Proof. Consider two sequences of m-bit words $A_{G0}^*(k+1)$, $A_{G0}^*(k+2)$, $A_{G0}^*(k+3), \ldots, A_{G0}^*(k+r)$ with even and odd numbers of $r < 2^m$ words.

According to Gilbert (Gilbert, 1958), (Statement 1), for any even number of r ($r \in \{2, 4, 6, \ldots, 2^m\}$) the values of the transition sequence t_{k+1}, t_{k+2}, t_{k+3},..., t_{k+r} include at least one element from $\{m-1, m-2, \ldots, 2, 1, 0\}$ an odd number of times. This means that at least one bit of any m-bit classic Gray code word will be inverted an odd number of times. According to this statement, an arbitrary bit of any m-bit anti-Gray code word will be also inverted an odd number of times. It should be noted that if r is even, then for its value $r \in \{2, 4, 6, \ldots, 2m\}$ we have $A_{G0}^*(k) \neq A_{G0}^*(k+r)$.

In the case when r is odd, at least one bit of m-bit words of the sequence $A_{G0}^*(k+1)$, $A_{G0}^*(k+2)$, $A_{G0}^*(k+3), \ldots, A_{G0}^*(k+r)$ will be inverted an odd number of times. The number of all transitions for r consecutive words of the anti-Gray sequence is $r(m-1)$, where r and $m-1$ are both odd numbers, so $r(m-1)$ is also an odd number. As a result, the common number of all transitions for consecutive words of the anti-Gray sequence is an odd number. Consequently, the inequality $A_{G0}^*(k) \neq A^*G0(k+r)$ is true for any $r \in 1, 3, 5, \ldots, 2^m-1$. ■

The two consecutive words of an anti-Gray sequence have the Hamming distance $m-1$, and the average Hamming distance (7) also has the same value, $m-1$.

It is easy to show that the maximal value of the Hamming distance between two binary words a_{m-1} $a_{m-2} \ldots a_1$ a_0 and $\bar{a}_{m-1}$ $\bar{a}_{m-2} \ldots$ $\bar{a}_1$ $\bar{a}_0$, where $a_j \in \{0, 1\}$, is m. One of these two binary words is the inversion of the other one.

To get a sequence with the Hamming distance between two consecutive binary words greater than $m-1$, let us also consider the original Gray sequence. Like any numerical counting sequence consisting of all possible m-bit binary combinations, the Gray sequence $A_{G0}(i) \in \{0, 1, 2, \ldots, 2^m-1\}$, $i = 0, 1, 2, \ldots, 2^m-1$ can be divided into two subsequences $A_{G0}(i) = A1(j) + A2(j)$, $j = 0, 1, 2, \ldots, 2^{m-1}-1$. The subsequence $A1(j)$, $0, 1, 2, \ldots, 2^{m-1}-1$, consists of the first $2^m/2$ words with the most significant bit equal to 0. The subsequence $A2(j)$ $\{2^{m-1}, 2^{m-1}+1, 2^{m-1}+2, \ldots, 2^m-1\}$, $j = 0, 1, 2, \ldots, 2^{m-1}-1$, consists of another part of words with the most significant bit equal to 1. For example, a Gray sequence for $m = 3$, $A(i) \in \{000, 001, 011, 010,$

$110, 101, 101, 100\}$, can be divided into $A1(j) = \{000, 001, 011, 010\}$ and $A2(j) = \{110, 101, 101, 100\}$. As can be seen, the words from the set $A1(j)$ are inversions of those from $A2(j)$.

This fact can be generalized as the following property of any counting numerical sequence $A = a_{m-1}a_{m-2}\dots a_2a_1a_0$ which consists of all possible 2^m binary combinations $a_{m-1}a_{m-2}\dots a_2a_1a_0$, generating in arbitrary order, where $a_q \in \{0,1\}$, $q \in \{0,1,2,\dots,m-1\}$ and all combinations appear in A only once.

Property 1. *Any numerical counting sequence $A(i) \in \{0,1,2,\dots,2^m-1\}$, $i = 0,1,2,\dots,2^m-1$, is divided into two subsequences $A1(j)$ and $A2(j)$, where $j = 0,1,2,\dots,2^{m-1}-1$, on the basis of the value of $a_q \in \{0,1\}$, $q \in \{0,1,2,\dots,m-1\}$, such that $A1(j)a_q = 0$ and $A2(j)a_q = 1$, and all the binary words of the sequence $A1(j)$ are the inversions of the words of the sequence $A2(j)$.*

This property follows from the definition of any numerical counting sequences. For example, the counting sequence $A(i) \in \{000, 001, 010, 011, 100, 101, 110, 111\}$ is divided into $A1(j) \in \{000, 001, 100, 101\}$ for $a_1 = 0$ and $A2(j) \in \{010, 011, 110, 111\}$ for $a_1 = 1$.

Based on Property 1, it is easy to construct an algorithm for rearranging words within any numerical counting system $A(i)$ in such a way that a word A belonging to $A1(j)$ will be followed by the word $\bar{A}$ from $A2(j)$ as the inversion of the word A. For a given numerical counting sequence (a counting sequence, a Gray code, an M-sequence, etc.) and the size of the word m, Algorithm 1 can be used to generate a sequence with the maximal average Hamming distance between two consecutive words (Yarmolik, 2006).

Algorithm 1. *Maximal average Hamming distance sequence generation.*

Input: {m; *numerical counting sequence* $A(i), i = 0,1,2,\dots,2^{m-1}-1$; *and q*}
Begin
1. *For* $i = 0$ *do*
2. *Generate* $A(i) = a_{m-2}a_{m-3}\dots a_1a_0$;
3. *According to the value of* q *insert* 0 *into the word* $a_{m-2}a_{m-3}\dots a_1a_0$ *to get* $A_M(2i)$ $= A1 = a_{m-1}\,a_m - 2\dots a_{q+1}0a_{q-1}\dots a_1a0$;
4. *Generate* $A_M(2i+1) = A2 =$ $= a^*_{m-1}a^*_{m-2}\dots a^*_{q+1}1a^*_{q-1}\dots a^*_1a^*_0$ *as negation of A1;*
5. *Increment:* $i = i + 1$;
6. *For* $i < 2^{2-1}$ *go to* 2*, else go to the* End

End
Output: {*Sequence* $A_M(i), i = 0,1,2,\dots,2^m-1$ *with maximal average Hamming distance*}.

In Table 6 examples of sequences with maximal average Hamming distances for the Gray code sequence with $m = 2$, $m = 3$ and $m = 4$, and different values of q, which have been obtained according to Algorithm 1, are shown.

The Hamming distances between two consecutive words of this sequence obtained on the basis of the original Gray code are m and $m-1$. It follows from the statement that the Hamming distance for the Gray code equals 1. Then the Hamming distance between the inverted Gray code word and the consecutive Gray code word equals $m-1$. There are the same numbers of consecutive pairs of words within $A_M(i)$ obtained based on the Gray code with the distances m and $m-1$ Then the average value of the Hamming distance $AHD[A_M(i), A_M(i+1)]$ is $m - 0.5$.

For experimental investigation, the common memory March tests, *MATS+* and *March C− (9N)* (Goor, 1991; Suk and Reddy, 1980; Cheng *et al.,* 2002), have been chosen. For both tests the two-runs testing procedure was implemented and the fault coverage for pattern sensitive faults PNPSF3 and PNPSF5 was calculated. It should be noted that the detection of all PNPSF*k* could be achieved as a result of generating all possible 2^k binary patterns in any k out of N memory cells. In Tables 7 and 8 experimental results for *MATS+* and *March C−* are shown, respectively, for counter and Gray $(A_C - A_G)$ address sequences; counter and anti-Gray sequences $(A_C - A^*_G)$; counter and sequence A_M with the maximal average Hamming distance $(A_C - A_M)$; the Gray sequence and anti-Gray $(A_G - A^*_G)$; Gray and $A_M - (A_G - A_M)$ and $(A^*_G - A_M)$.

For both tests the estimates of fault coverages were obtained for two different sizes of the memory determined by the value of m. The presented results show that the fault coverage does not depend on memory size.

In (Yarmolik and Sokol, 2006), it was shown that the fault coverage for PNPSF*k* for multiple runs of March testing with a change in the starting address (seed) for address sequences has limitations. The value of the limit cannot be reached using only one address sequence for multi-run memory testing. At the same time the application of different address sequences is allowed to increase the fault coverage (see Tables 7 and 8). But the fault coverage largely depends on combinations of the address sequences used. For example, the application of the counter sequence A_C and the Gray sequence A_G for two runs of the March test $MATS+$ permits to detect only 17.4% of all possible PNPSF3 while the application of the counter sequence A_C and the anti-Gray sequence A^*_G allows us to detect 21.5% (see Table 7).

Experimental results show the efficiency of the proposed metric of the average Hamming distance $AHD[A(i), A(i+k)]$ and the efficiency of using different combinations of address sequences for multiple runs of March testing.

Table 6. Sequences with maximal average Hamming distances.

i	$m=2$		$m=3$			$m=4$			
	$q=1$	$q=0$	$q=2$	$q=1$	$q=0$	$q=3$	$q=2$	$q=1$	$q=0$
0	00	00	000	000	000	0000	0000	0000	0000
1	11	11	111	111	111	1111	1111	1111	1111
2	01	10	001	001	010	0001	0001	0001	0010
3	10	01	110	110	101	1110	1110	1110	1101
4			011	101	110	0011	0011	0101	0110
5			100	010	001	1100	1100	1010	1001
6			010	100	100	0010	0010	0100	0100
7			101	011	011	1101	1101	1011	1011
8						0110	1010	1100	1100
9						1001	0101	0011	0011
10						0111	1011	1101	1110
11						1000	0100	0010	0001
12						0101	1001	1001	1010
13						1010	0110	0110	0101
14						0100	1000	1000	1000
15						1011	0111	0111	0111

Table 7. Fault coverage for PNPSF3 and PNPSF5 for two-run $MATS+$ testing

Address sequences	PNPSF3		PNPSF5	
	$m=4$	$m=8$	$m=4$	$m=8$
$A_C - A_G$	17.4	17.7	4.9	5.0
$A_C - A_G^*$	21.5	21.5	5.9	5.9
$A_C - A_M$	20.7	21.0	5.7	5.8
$A_G - A_G^*$	20.6	20.3	5.6	5.7
$A_G - A_M$	18.8	19.0	5.3	5.4
$A_G^* - A_M$	18.3	18.9	5.4	5.4

Table 8. Fault coverage for PNPSF3 and PNPSF5 for two-run $MarchC-$ testing.

Address sequences	PNPSF*3*		PNPSF*5*	
	$m=4$	$m=8$	$m=4$	$m=8$
$A_C - A_G$	66.4	66.5	19.8	20.2
$A_C - A_G^*$	72.1	71.7	22.1	22.0
$A_C - A_M$	69.1	68.8	21.0	20.9
$A_G - A_G^*$	71.5	70.2	21.6	21.5
$A_G - A_M$	69.4	69.3	21.0	21.0
$A_G^* - A_M$	70.6	69.6	21.5	21.4

4.2. Multi-background March memory testing. To achieve a high fault coverage of PNPSFk for multi-run memory testing it is quite important to choose appropriate backgrounds depending on the type of memory test. Let us concentrate on three types of memory tests (2). The first one allows us to generate only one background within neighbouring cells, like $MATS+$, the second, two backgrounds (i.e., $MarchC-$), and the third, four backgrounds, as was shown in the case of $MarchPS(23N)$. Obviously, for different types of memory tests the optimal backgrounds will be different. For example, in the case of the two-run testing of an eight-bit memory (see Table 10), 11111111 is the second optimal background for the test $MATS+$ to detect PNPSF3, and for $MarchPS(23N)$, 10101010 is one of the optimal backgrounds for detecting the same $PNPSF3$. As the first background in both the cases the all-zero background was applied, i.e., $B = b_0b_1b_2b_3b_4b_5b_6b_7 = 00000000$.

To select an optimal background, let us use the proposed Hamming distances $HD(B_g, B_d)$ between two backgrounds B_g and B_d for multi-background memory testing as the metric for background selection. The selection algorithm for optimal background selection forms a basis for the following statements.

Statement 3. *In the case of m runs of the memory test which allows us to generate only one pattern (3) within neighbouring cells based on backgrounds $B_0, B_1, B_2, \dots, B_{m-1}$, an optimal set of backgrounds of this type should have the maximal Hamming distance $HD(B_g, B_d)$ between any pair (B_g, B_d), where $g, d \in \{0, 1, 2, \dots, m-1\}$.*

Table 9. Two-background memory test fault coverage.

MATS+		March PS(23N)	
Second Pattern $a_0a_1\cdots a_6a_7$	FC [%]	Second Pattern $a_0a_1\cdots a_6a_7$	FC [%]
00000001 00000010 ... 10000000	4×17.1	00000001 01111111 10000000 11111110	4×72.9
00000011 00000101 ... 11000000	4×20.5	00000111 00011111 11111000 11100000	4×80.1
00000111 00001011 ... 11100000	4×22.8	00001011 00101111 11010000 11110100	4×81.9
00001111 00010111 ... 11110000	4×24.1	00011011 00100111 11011000 11100100	4×83.9
00011111 00101111 ... 11111000	4×24.8	00101011 11010100	4×85.1
00111111 01011111 ... 11111100	4×25.0	01010101 01010110 ... 10101010	4×86.9

This statement can be used for the selection of an optimal value of the background for a *MATS+* like memory test.

Statement 4. *In the case of m runs of the memory test which allows us to generate two patterns within neighbouring cells based on backgrounds B_0, B_1, B_2, $\ldots$, B_c, an optimal set of backgrounds of this type should have the maximal Hamming distance $HD(B_g,\ B_d)$ between any background pair $(\bar{B}_g, \bar{B}_d)$, where $g, d \in \{0, 1, 2, \ldots, c-1\}$ and $B_g^{'}$ is the background B_g or its negation $\bar{B}_g$.*

This statement can be applied to background selection for the second (two patterns) and third (four patterns) types of March tests, like $MarchC-$ and $MarchPS(23N)$, respectively.

Some experiments were performed to confirm those statements. All possible combinations of three backgrounds $B0$, $B1$ and $B2$ for 8 memory cells were generated. During this process the fault coverage of the discussed tests for PNPSF3 was obtained. The Hamming distance between all pairs of the background was calculated and presented. The achieved results are shown in Tables 10 and 11. It should be noticed that the same fault coverage was obtained for many different background sets. Therefore in the tables there is only part of all obtained results.

Table 10. Correlation between the Hamming distance and the fault coverage for *MATS+*.

Background set	**Hamming distance**	**FC [%]**
$B_0 = 00000000$ $B_1 = 00000001$ $B_2 = 00000010$	$H(B_0, B_1) = 1$ $H(B_0, B_2) = 1$ $H(B_1, B_2) = 2$	3×21.87
$B_0 = 00000000$ $B_1 = 00000111$ $B_2 = 10110000$	$H(B_0, B_1) = 3$ $H(B_0, B_2) = 3$ $H(B_1, B_2) = 6$	3×33.03
$B_0 = 00000000$ $B_1 = 11001111$ $B_2 = 11110010$	$H(B_0, B_1) = 6$ $H(B_0, B_2) = 5$ $H(B_1, B_2) = 5$	3×37.05

Table 11. Correlation between the Hamming distance and the fault coverage for *March PS(23N)*

Background set	**Hamming distance**	**FC [%]**
$B_0 = 00000000$ $B_1 = 00000001$ $B_2 = 00000010$	$HD(B_0, B_1) = 1$ $HD(B_0, B_1^*) = 7$ $HD(B_0, B_2) = 1$ $HD(B_0, B_2^*) = 7$ $HD(B_1, B_2) = 2$ $HD(B_1, B_2^*) = 6$	6×79.16
$B_0 = 00000000$ $B_1 = 00000111$ $B_2 = 10110000$	$HD(B_0, B_1) = 3$ $HD(B_0, B_1^*) = 5$ $HD(B_0, B_2) = 3$ $HD(B_0, B_2^*) = 5$ $HD(B_1, B_2) = 6$ $HD(B_1, B_2^*) = 2$	6×89.28
$B_0 = 00000000$ $B_1 = 11001001$ $B_2 = 10010011$	$HD(B_0, B_1) = 4$ $HD(B_0, B_1^*) = 4$ $HD(B_0, B_2) = 4$ $HD(B_0, B_2^*) = 4$ $HD(a_1, B_2) = 4$ $HD(a_1, B_2^*) = 4$	6×96.42

In Tables 10 and 11 examples of experimental results can be seen for multi-backgrounds $MATS+$ and $MarchPS(23N)$ tests. Both tests were run three times with different backgrounds, as shown in Tables 10 and 11. It can be noticed how important background selection is. In the first case ($MATS+$ test), the fault coverage changes in the range from 21.87% to 37.05%. It can be noticed that the minimal fault coverage (21.87%) was obtained when the Hamming distance among all pairs of backgrounds was minimal ($HD(B_0, B_1) = 1$, $HD(B_0, B_2) = 1$, $HD(B_1, B_2) = 2$), and the maximal value of the fault coverage (37.05%) was obtained for the backgrounds for which the Hamming distance among all pairs of backgrounds was maximal (in this case $HD(B_0, B_1) = 6$, $HD(B_0, B_2) = 5$, $HD(B_1, B_2) = 5$). It should be underlined that to obtain the best fault coverage the value of the Hamming distance between all pairs of backgrounds should be maximized (see Statement (3) and Table 10).

The same conclusions can be drawn from the results shown in Table 11. However, it should be noticed that in Table 11 the Hamming distance was calculated not only between the original backgrounds, but also between their inversions. This is because the test $MarchPS(23N)$ used generates more than one pattern within neighbourhood cells and, according to Statement (4), the Hamming distance should be maximized between both the original backgrounds and their inversions. In reality, $MarchPS(23N)$ generates four patterns where two of them are the inverted versions of the other two patterns.

The results presented in Tables 10 and 11 prove that Statements (3) and (4) can be used during the background selection process. Those results show the high correlation between the Hamming distance between all pairs of backgrounds and the fault coverage of the multi-run testing. Therefore we can use those statements in the multi-background memory testing process.

5. Conclusions

In this paper the unrestricted neighbourhood pattern sensitive fault model was considered and a subtype of this model, the so-called passive neighbourhood pattern sensitive faults (PNPSF*k*), was chosen as a target fault model for multi-run memory testing. The efficiency of traditional March tests to detect PNPSFk was analyzed and their low ability to detect such a type of faults was shown. As a solution to increase the fault coverage, multi-run memory testing was proposed. Any new run of the same memory test should be done with new initial conditions, namely, with a new memory background or a new memory address sequence. To choose appropriate memory address sequences for consecutive memory test execution, an average Hamming distance was proposed and validated as the metric. New address sequences with maximal values of the average Hamming distance and algorithms for their generation were obtained. Experimental results showed that the high fault coverage could be reached for address sequences with various values of this metric. For comparable values of this metric, the fault coverage is lower. In the case of multi-background memory testing, an optimal background could be selected on the basis of the Hamming distance between backgrounds. If the Hamming distance between all pairs of backgrounds is high, then the fault coverage also takes a high value.

References

Bernardi E., Sancez M., Squillero G. and Sonza Reorda M. (2006). An effective technique for minimizing the cost of processor software-based diagnosis in SoCs, *Proceedings of the Conference on Design, Automation and Test in Europe*, Munich, Germany, pp. 412–417.

Bernardi P., Grosso M., Rebaudengo M. and Sonza Reorda M. (2005). Exploiting an infrastructure IP to reduce the costs of memory diagnosis in SoCs, *Proceedings of the European Test Symposium*, Tallinn, Estonia, pp. 202–207.

Cheng K.-L., Tsai M.-F. and Wu C.-W. (2001). Efficient neighborhood pattern-sensitive fault test algorithms for semiconductor memories, *Proceedings of the IEEE VLSI Test Symposium (VTS)*, Marina del Rey, CA, USA, pp. 225–237.

Cheng K.-L., Tsai M.-F. and Wu C.-W. (2002). Neighborhood pattern sensitive fault testing and diagnostics for random access memories, *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* **21**(11): 1328–1336.

Cockburn B. E. (1995). Deterministic tests for detecting scrambled pattern-sensitive faults in RAMs, *Proceeding IEEE International Workshop on Memory Technology, Design and Testing (MTDT 95),* San Jose, CA, USA, pp. 117–122.

Franklin M. and Saluja K. K. (1996). Testing reconfigured RAM's and scrambled address RAM's for pattern sensitive faults, *IEEE Transactions on Computer-Aided Design of Integrated Circuits* **15**(9): 1081–1087.

Gilbert E. N. (1958). Gray codes and paths on the n-cube, *Bell System Technical Journal* **37**: 815–826.

Goor A. J. v. d. (1991). *Testing Semiconductor Memories: Theory and Practice*, John Wiley & Sons, Chichester.

Gray F. (1958). Pulse code communication, *U.S. Patent 2,632,058*.

Hayes J. P. (1975). Detection of pattern sensitive faults in random access memories, *IEEE Transactions on Computers* **24**(2): 150–157.

Hayes J. P. (1980). Testing memories for single cell pattern sensitive fault, *IEEE Transactions on Computers* **29**(2): 249–254.

Li J.-F. (2007). Transparent-test methodologies for random access memories without/with ECC, *Transactions on Computer-Aided Design of Integrated Circuits and Systems* **26**(10): 1888–1983.

Niggemeyer D., Redeker M. and Otterstedt J. (1998). Integration of non-classical faults in standard march tests, *Proceedings of the 1998 IEEE International Workshop on Memory Technology, Design and Testing*, San Jose, CA, USA, pp. 91–96.

Niggemeyer D., Redeker M. and Rudnick E. (2000). Diagnostic testing of embedded memories based on output tracing, *Proceedings of the IEEE International Workshop on Memory Technology, Design and Testing*, San Jose, CA, USA, pp. 113–118.

Pomeranz I. and Reddy S. M. (2006). Fault detection by output response comparison of identical circuits using half-frequency compatible sequences, *Proceedings of the International Test Conference*, Santa Clara, CA, USA, pp. 202–207.

Savage C. (1997). A survey of combinatorial Gray codes, *SIAM Review* **39**(4): 605–629.

Sokol B. and Yarmolik S. V. (2006). Address sequence for March tests to detect pattern sensitive faults, *Proceedings of the 3rd IEEE International Workshop on Electronic Design Test & Applications (DELTA'06)*, Kuala Lumpur, Malaysia, pp. 354–357.

Suk D. S. and Reddy S. M. (1980). Test procedures for a class of pattern sensitive faults in semiconductor random access memories, *IEEE Transactions on Computers* **29**(6): 419–429.

Yarmolik S. V. (2006). Gray code with maximum of Hamming distance, *Proceedings of the 4th International Science-Practice Forum on Information Technologies and Cybernetics*, Dnipropetrovsk, Ukraine, p. 77.

Yarmolik S. V. and Sokol B. (2006). Optimal memory address seeds for pattern sensitive faults detection, *Proceedings of the IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS'2006)*, Prague, Czech Republic, pp. 220–221.

Yarmolik S. V. and Yarmolik V. N. (2006a). Memory pattern sensitive faults detection using multiple runs of March tests, *Informatics* **1**(9): 104–113.

Yarmolik V. N., Klimets Y. and Demidenko, S. (1998). March PS(23n) test for DRAM pattem sensitive faults, *Proceedings of the 7th IEEE Asian Test Symposium (ATS)*, Singapore, pp. 354–351.

Yarmolik V. N. and Yarmolik S. V. (2006b). Address sequences for multiple run march tests, *Automatic Control and Computer Sciences* **5**: 59–68.

Received: 6 November 2007
Revised: 10 April 2008