Memory test experiment: industrial results and data

S. Hamdioui, Ad J. van de Goor, J. Delos Reyes and M. Rodgers

Abstract: The results of 12 well-known and three fault-primitive-based memory test algorithms applied to 0.13 micron technology 512 kB single-port SRAMs are presented. Each test algorithm is used with up to 16 different stress combinations (SCs) (i.e. different address sequences, data backgrounds and voltages) resulting in 122 tests. The results show that SCs influence the fault coverage (FC) of the test algorithms, that the highest FC is obtained at a low voltage level and that the highest detected number of unique faults is obtained at a high voltage level. They also show that the tests with the most promising FC, based on the theory, also tend to have the highest FC in practice. Moreover, the test results show that some algorithms detect faults that cannot be explained with the existing fault models, indicating that the existing fault models still leave much to be explained; for example, no theoretical basis exists to model the stresses and the predicted FC for a given test.

1 Introduction

Random access memories, which are an integral part of any ULSI chip (e.g. a microprocessor), are widely considered to be one of the most critical components of current digital systems, not only because the memory share of the chip area is increasing and is expected to be about 94% in 2014 [1], but also because of technology shrinking, which makes memories more sensitive to defects.

Tests for semiconductors' memories have experienced a long development process. Before 1980, tests had very long test times for a given fault coverage (FC) (i.e. the number of detected faults divided by the number of total faults), typically of order $O(n^2)$, where n is the size of the memory. Such tests can be classified as the *ad hoc tests* because of the absence of fault models and proofs. Tests such as zero one, GALPAT and Walking 1/0 belong to this class [2].

To reduce the test time and improve the FC, test development has been focused on the possible faults that can appear in the memory. For that reason, functional fault models, which are abstract fault models, were introduced during the early 1980s. The advantage of these models was that the FC could be proven while the test time was usually of order $O(n)$, that is, linear with the size of the memory. Some important functional fault models introduced in that time period were the stuck-at fault, the address decoder fault (AF) [3], the coupling fault (CF) [4–6] and the neighbourhood pattern sensitive fault [7, 8]. These functional fault models were abstract fault models not based on real memory design and/or real defects. To reflect the faulty behaviour of real defects in real designs, defect injection and circuit simulation, as well as inductive fault analysis (IFA) [9, 10], were introduced and used. IFA is a systematic procedure to predict the faults in an integrated circuit by injecting spot defects in the simulated geometrical representation of the circuit. It allows for the establishment of the fault models based on simulated defects in real designs. The result was that new functional fault models were introduced [11], for example the state coupling fault, the data retention fault, the stuck open fault, and the un-restored write fault [12]. After the introduction of the functional fault models, march tests have become the dominant type of tests because of their advantages. Their test times are linear with the size of the memory, and the FC of the considered fault models can be mathematically proven. Many march tests have been introduced with different degrees of success [2–6, 11, 13–17].

In the late 1990s, the experimental results of applying a large number of tests to a large number of chips [18–22] indicated that many functional tests detect faults which cannot be explained with the existing fault models at that time. This led to the introduction of the framework of all possible fault models for memories based on the fault primitive (FP) concept [23, 24]. The concept of FP also allowed for the classification of the memory faults framework in different classes: static against dynamic [23, 25] (i.e. depending on the number of operations required to sensitise the fault), simple against linked [7, 26, 27] (i.e. depending on the way the FPs manifest themselves) and so on. The framework's classes have been analysed experimentally to investigate their validity. This led to the introduction of new fault models [17, 28–30]: write disturb fault, incorrect read fault, transition CF and so on.

In our previous work, we introduced three FP-based tests to target three different fault classes: March SS [31] for static faults, March RAW [25] for dynamic faults and March SL [27] for linked faults. The question is how efficient are these FP-based tests in practice when compared with the traditional tests for advanced semiconductor memories. To answer this question, a memory test experiment at Intel was performed; the results and their analysis are summarised in this paper. The paper is organised as follows: Section 2 gives an overview of the used BTs and stresses in the experiment; Section 3 describes the test

© IEE, 2006

IEE Proceedings online no. 20050104

doi:10.1049/ip-cdt:20050104

Paper first received 2nd June and in revised form 5th October 2005

S. Hamdioui and Ad J. van de Goor are with the Computer Engineering Laboratory, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands.

J. Delos Reyes and M. Rodgers are with Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052, USA

E-mail: s.hamdioui@ewi.tudelft.nl

results; Section 4 compares the fault coverage (FC) of the FP-based BTs and that of the traditional BTs; Section 5 analyses the impact of stresses on the FP-based tests; Section 6 presents an optimal set of BTs and stresses; Section 7 presents conclusions.

This paper presents the results of a test set consisting of 15 base tests (BTs) (i.e. test algorithms), each with up to 16 stress combinations (SCs) (e.g. different address sequences, data backgrounds (DBs) and voltages) resulting in 122 tests applied to 0.13 micron technology 512 kB single-port SRAMs; the testing was performed at 200 MHz using direct access mode. Note that a test consists of a BT applied using a particular SC. The design of the memory under consideration is based on 6T cells with required peripherals' circuits (sense amplifiers, precharge circuits, write drivers address decoders, etc.).

2 Used BTs and stresses

This section gives an overview of the used BTs and stresses during the memory test experiment.

2.1 Overview of used BTs

Table 1 lists the used BTs; the test length 'TL' of each BT is also included where n denotes the size of the memory cell array, C the number of columns and R the number of rows. The used march notation is explained as follows [6]: a complete march test is delimited by the '{...}' bracket pair, whereas a march element is delimited by the '(...)' bracket pair. March elements are separated by semicolons, and the operations within a march element are separated by commas. Note that all operations of a march element are performed at a certain address before proceeding to the next address, which can be done in either an increasing (⇑) or a decreasing (⇓) address order. When the address order is not relevant, the symbol (⇕) is used.

As mentioned previously, the set of used BTs in the experiment consists of three FP-based BTs and only 12 well-known traditional BTs (because of time constraints). The BTs with the most promising FC and unique fault (UF) detection were selected [3, 13, 18, 20–22, 32].

2.1.1 FP-based BTs:

The FP-based BTs consist of three march tests listed in the first block of Table 1.

- March SS [31, 32] to target all simple static memory faults. Static faults are faults sensitised by performing at most one operation; for example, the state of the cell is always stuck at one, a read operation to a certain cell causes that cell to flip. Simple faults are faults which cannot influence the behaviour of each other. That means that the behaviour of a simple fault cannot change the behaviour of another one; therefore masking cannot occur.
- March RAW [25] to target some dynamic faults. Dynamic faults are faults that can only be sensitised by performing more than one operation sequentially; for example, two successive read operations cause the cell to flip; however, if only one read operation is performed, the cell will not flip [29, 30]. March RAW is designed to target dynamic faults caused by read-after-write operations that have been observed in real designs [25].
- March SL [27] to target all simple linked faults. Linked faults are faults that influence the behaviour of each other [3, 7, 26]. That means that the behaviour of a certain fault can change the behaviour of another one such that masking can occur. Masking makes the testing of linked faults very complex when compared with testing of simple faults.

2.1.2 Traditional BTs:

A set of 12 well-known BTs has been selected for the experiment; BTs with the most promising FC and unique fault detection were selected. They are listed in the second block of Table 1. For Hammer, the notation for example, 10 ∗ w1 means that the write operation is performed ten times successively to the same cell. As the original versions of GALPAT and Walking 1/0 have a

Table 1: Description of the used BTs

Test	TL	Description
March SS [31]	$22n$	⇕ ⇑(r0,r0,w0,r0,w1); ⇑(r1,r1,w1,r1,w0); ⇓(r0,r0,w0,r0,w1); ⇓(r1,r1,w1,r1,w0); ⇕(r0)}
March RAW [25]	$26n$	{⇕(w0); ⇑(r0, w0,r0,r0,w1,r1); ⇑(r1,w1,r1,r1,w0,r0); ⇓(r0,w0,r0,r0,w1,r1); ⇓(r1,w1,r1,r1,w0,r0); ⇕(r0)}
March SL [27]	$41n$	{⇕(w0); ⇑(r0,r0,w1,w1,r1,r1,w0,w0,r0,w1); ⇑(r1,r1,w0,w0,r0,r0,w1,w1,r1,w0); ⇓(r0,r0,w1,w1,r1,r1,w0,w0,r0,w1); ⇓(r1,r1,w0,w0,r0,r0,w1,w1,r1,w0)}
SCAN [13]	$4n$	{⇑(w0); ⇑(r0); ⇑(w1); ⇑(r1)}
MATS + [5]	$5n$	{⇕(w0); ⇑(r0,w1); ⇓(r1,w0)}
MATS++ [2]	$6n$	{⇕(w0); ⇑(r0,w1); ⇓(r1,w0,r0)}
March C- [4]	$10n$	{⇕(w0); ⇑(r0,w1); ⇑(r1,w0); ⇓(r0,w1); ⇓(r1,w0); ⇕(r0)}
PMOVI [16]	$13n$	{⇓(w0); ⇑(r0,w1,r1); ⇑(r1,w0,r0); ⇓(r0,w1,r1); ⇓(r1,w0,r0)}
March SR [17]	$14n$	{⇓(w0); ⇑(r0,w1,r1,w0); ⇑(r0,r0); ⇑(w1); ⇓(r1,w0,r0,w1); ⇓(r1,r1)}
March G [6]	$23n$	{⇕(w0); ⇑(r0,w1,r1,w0,r0,w1); ⇑(r1,w0,w1); ⇓(r1,w0,w1,w0); ⇓(r0,w1,w0); ⇑(r0,w1,r1); ⇑(r1,w0,r0)}
Hammer [21]	$49n$	{⇑(w0); ⇑(r0,10 ∗ w1,r1); ⇑(r1,10 ∗ w0,r0); ⇓(r0, 10 ∗ w1,r1); ⇓(r1,10 ∗ w0,r0)}
GalColumn [2]	$6n+4nR$	{⇑(w0); $\Uparrow_b$(w1$_b$, col(r0,r1$_b$), w0$_b$); ⇑(w1); $\Uparrow_b$(w0$_b$,col(r1,r0$_b$),w1$_b$)}
GalRow [2]	$6n+4nC$	{⇑(w0); $\Uparrow_b$(w1$_b$, row(r0,r1$_b$), w0$_b$); ⇑(w1); $\Uparrow_b$(w0$_b$,row(r1,r0$_b$),w1$_b$)}
WalkColumn [3]	$8n+2nR$	{⇑(w0); $\Uparrow_b$(w1$_b$, col(r0),r1$_b$,w0$_b$); ⇑(w1); $\Uparrow_b$(w0$_b$,col(r1),r1$_b$,w0$_b$)}
WalkRow [3]	$8n+2nC$	{⇑(w0); $\Uparrow_b$(w1$_b$, row(r0),r1$_b$,w0$_b$); ⇑(w1); $\Uparrow_b$(w0$_b$,row(r1),r1$_b$,w0$_b$)}

time complexity of $O(n^2)$, two $O(n.\sqrt{n})$ versions of GALPAT and Walking 1/0 BTs have been used. The two $O(n.\sqrt{n})$ versions of GALPAT are: GalColumn and GalRow. For example, for GalColumn the read action is restricted to only the cells in the same column as the base cell, instead of galloping through the memory. For GalRow and GalColumn, the notation, for example, row(r0,r1_b) means apply an r0 (read 0) operation in an incrementing order to the cells of the row of the base cell and apply r1 (read 1) operation to the base cell after each r0 operation; a similar explanation applies to col(r0,r1_b). Similarly, for WalkRow and WalkColumn, the notation, for example, row(r0) means apply an r0 operation using an incrementing address order to the row of the base cell and skip the base cell; a similar explanation applies to col(r0).

2.2 Used stresses

Each BT has been applied using several different SCs. An algorithm SC specifies the way the test is performed, and therefore it influences the sequence and/or the type of the memory operations. The used stresses are the address directions and the DBs.

The used addressing directions consist of fx and fy.

'Fast x' (fx): 'Fast x' addressing is simply incrementing or decrementing the address in such a way that each step goes to the next row.

'Fast y' (fy): 'Fast y' addressing is simply incrementing or decrementing the address in such a way that each step goes to the next column.

A DB is defined as the pattern of ones and zeros as seen in an array of memory cells. The used DBs are:

1. Solid (s): all 0s and all 1s.
2. Checkerboard (c): 0101.../1010.../0101.../1010...
3. Column stripe (cs): 0101.../0101.../0101.../0101...
4. Row stripe (rs): 0000.../1111.../0000.../1111...

Table 2 lists the 61 tests applied at both high voltage and low voltage; a test consists of a BT (i.e. test algorithm) applied using a particular SC. The total number of tests is then the number of BTs (i.e. 15), each multiplied with the corresponding number of SCs (#SC) and with 2 {high- and low-voltage testing}, that is, total tests = $\Sigma(BT_i * (\#SC_i) * 2) = 122$. The column 'TT/SC' gives the test time, in milliseconds (ms), of each BT using a single SC for the tested chip. To calculate the test time per BT, the 'TT/SC' has to be multiplied with '#SC' and with 2 {high- and low-voltage testing}. The total test time of all tests is 160.942 ms/chip, where the four non-linear BTs consume about 43% of the total test time. In the table, the solid, the checkerboard, the column stripe and the row stripe DBs are denoted as 's', 'c', 'cs' and 'rs', respectively, the addressing directions are denoted as 'fx' and 'fy'. A '+' in the table indicates that the corresponding SC is applied, and a '−' denotes that it is not; for example, WalkRow is used with fy (fast y) and s (solid) DB. Because of time constraints, only stresses with the most promising FC for traditional BTs were selected [3, 13, 18, 20–22, 32].

3 Test results

All SCs have been implemented at two different voltage levels: high voltage (HVcc) and low voltage (LVcc). Testing a huge number of 512 kB SRAM chips resulted in:

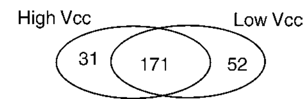

Fig. 1 *Venn diagram of the FC*

- HVcc testing: 1545 chips failed; 1343 chips failed all tests (i.e. 61), whereas 202 chips only failed some tests.
- LVcc testing: 1543 chip failed; 1320 chips failed all tests, whereas 223 chips only failed some tests.

From now on, we will concentrate only on the chips that did not fail all tests, as they are the most interesting.

Fig. 1 shows the Venn diagram of the influence of the voltage levels on detectable faults, as derived from the test results database. The total number of devices found to be faulty is: $31 + 171 + 52 = 254$. The FC at HVcc testing is 202 out of 254, whereas the FC at LVcc testing is 223 out of 254. Note that 171 faults are detected both at LVcc and HVcc. In addition, 52 faults are detected at LVcc only, whereas 31 faults are detected at high voltage only. This clearly explains the necessity of testing at both voltages in order to achieve a good FC. Low voltage testing is important for detecting faults caused by high-ohmic bridges [20, 33, 34], whereas high voltage testing is important for detecting resistive open defects [30, 35, 36].

As the database of the test results is very large, it has to be simplified for analysis purposes. Therefore we will first consider the FC of each BT. The FC of a BT is the union of the FCs of its corresponding SCs. A die belongs to the union (i.e. considered detected by a BT) if at least one SC of that BT found the die to be faulty. For example, MATS+ is implemented using fx-s (i.e. 'fast x' and solid DB) and fy-s; the fault is considered detected if at least one of the two MATS+ tests detects the fault (Table 2).

Table 3 shows the unions and intersections of the 15 BTs for HVcc and Table 4 shows the results for LVcc. A die belongs to the union of two BTs if at least one of the two BTs found the die to be faulty and belongs to the intersection of two BTs if both BTs found the die to be faulty. The first column in each table gives the BT number; the second column gives the name of the BT. The column 'FC' lists the FC of the corresponding BT; the column 'UFs' gives number of unique faults (UFs) each BT detects. UFs are faults that are only detected once with a single test; for example, at HVcc, GalRow detects nine UFs that are not detected with any other test.

The union and intersection of each pair of BTs are shown in the rest of the tables. The numbers on the diagonal in bold give the FC of the BTs, which are also listed in the column 'FC'; for example, at HVcc, March SS has FC = 184. The part above the main diagonal shows the union for each BT pair, whereas the part under the diagonal lists the intersection of each BT pair; for example, at HVcc, the union of March C- and PMOVI is 185 and their intersection is 179. On the basis of the two tables and the Venn diagram, one can conclude the following.

- *HVcc testing*

1. The total number of faulty chips detected is 202.
2. The best BTs, in terms of FC, are March SL and March G with FC = 185, March SS and March RAW with FC = 184 and March C- with FC = 183.
3. There are 12 UFs detected with four tests; these are listed in Table 5 together with their FC and the number of UFs (#UFs) each detects.

Table 2: List of the used BTs and their SCs

Number	BT	TT/SC, ms	SC	SC							
				fx				fy			
				s	c	cs	rs	s	c	cs	rs
1	GalColumn	21.217	1	+	−	−	−	−	−	−	−
2	GalRow	1.556	1	−	−	−	−	+	−	−	−
3	Hammer	2.000	1	+	−	−	−	−	−	−	−
4	March C−	0.410	6	+	−	+	+	+	−	+	+
5	March G	0.942	2	+	−	−	−	+	−	−	−
6	MATS+	0.205	2	+	−	−	−	+	−	−	−
7	MATS++	0.246	2	+	−	−	−	+	−	−	−
8	PMOVI	0.532	8	+	+	+	+	+	+	+	+
9	March RAW	1.065	8	+	+	+	+	+	+	+	+
10	Scan	0.164	4	−	+	+	+	−	+	−	−
11	March SL	1.679	8	+	+	+	+	+	+	+	+
12	March SR	0.573	8	+	+	+	+	+	+	+	+
13	March SS	0.901	8	+	+	+	+	+	+	+	+
14	WalkColumn	10.813	1	+	−	−	−	−	−	−	−
15	WalkRow	0.983	1	−	−	−	−	+	−	−	−

4. The best union pair in terms of the FC is 195 achieved with GalRow and March G and with GalRow and March SL (Table 3).

• *LVcc testing*

1. The total number of faulty chips detected is 223.
2. The best BTs, in terms of FC, are March C- with FC = 215, March SL with FC = 209 and March SS and March RAW with FC = 208.
3. There are no UFs detected at LVcc testing.
4. The best union pair in terms of the FC is 220 achieved with March C- and March RAW (Table 4).

It is important to note here that the three FP-based BTs (i.e. March SS, March SL and March RAW) score very high for both HVcc and LVcc testing.

Using Tables 3 and 4, it is possible to determine the BTs detecting supersets of faults in comparison with the other BTs in this experiment. For example, GalColumn detects a superset of WalkColumn at HVcc testing (Table 3); this is because the intersection of the two tests is 160 (which is the FC of WalkColumn), and their union is 164 (which is the FC of GalColumn). Keep in mind that in this experiment, the number of stresses used with each BT is not the same for all BTs (Table 2). Determining the BTs detecting supersets allows for deriving a reduced set of BTs that has the same FC as the initial test set (Table 1). The reduced set is given in Table 6; it consists of nine BTs for HVcc as well as for LVcc, where eight BTs are common.

4 FP-based BTs analysis

In this section, the evaluation of the FC of three FP-based BTs (denoted as FP-BTs) (i.e. March SS, March SL and March RAW) will be done and compared with the FC of the other 12 BTs. One useful way to do that is to calculate

Table 3: Union and intersection of BTs at HVcc (total FC = 202)

Number	BT Name	FC	UFs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	GalColumn	164	0	**164**	188	179	183	186	180	178	183	186	177	186	181	184	164	181
2	GalRow	176	9	152	**176**	192	192	*195*	189	188	190	193	190	*195*	192	193	188	178
3	Hammer	171	1	156	155	**171**	186	188	186	184	185	186	181	188	184	186	178	183
4	March C−	183	0	164	167	168	**183**	186	184	183	185	187	186	186	187	185	183	183
5	March G	185	0	163	166	168	182	**185**	186	186	186	188	188	188	189	187	186	186
6	MATS+	175	0	159	162	160	174	174	**175**	179	183	186	181	185	185	184	179	181
7	MATS++	177	0	163	165	164	177	176	173	**177**	183	186	181	186	185	184	177	181
8	PMOVI	181	0	162	167	167	179	180	173	175	**181**	184	185	187	187	185	183	181
9	March RAW	184	0	162	167	169	180	181	173	175	181	**184**	188	188	188	186	186	184
10	Scan	168	1	155	154	158	165	165	162	164	164	164	**168**	188	178	186	173	181
11	March SL	185	1	163	166	168	182	182	175	176	179	181	165	**185**	188	186	186	186
12	March SR	170	0	153	154	157	166	166	160	162	164	166	160	167	**170**	187	178	184
13	March SS	184	0	164	167	169	182	182	175	177	180	182	166	183	167	**184**	184	184
14	WalkColumn	160	0	160	148	153	160	159	156	160	158	158	155	159	152	160	**160**	181
15	WalkRow	168	0	151	166	156	168	167	162	164	168	168	155	167	154	168	147	**168**

Table 4: Union and intersection of BTs at LVcc (total FC = 223)

Number	BT Name	FC	UFs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	GalColumn	184	0	**184**	212	210	215	204	199	197	205	212	195	211	205	209	186	202
2	GalRow	194	0	166	**194**	212	218	213	211	211	212	214	208	215	210	214	211	196
3	Hammer	199	0	173	181	**199**	217	216	211	210	214	215	208	216	212	217	209	209
4	March C−	215	0	184	191	197	**215**	218	215	215	219	*220*	215	219	217	218	215	215
5	March G	196	0	176	177	179	193	**196**	200	200	206	212	202	213	209	210	203	205
6	MATS+	191	0	176	174	179	191	187	**191**	193	204	211	196	209	204	208	196	202
7	MATS++	191	0	178	174	180	191	187	189	**191**	204	211	194	210	203	208	194	201
8	PMOVI	201	0	180	183	186	197	191	188	188	**201**	209	204	213	209	210	204	205
9	March RAW	208	0	180	188	192	203	192	188	188	200	**208**	211	213	212	213	211	208
10	Scan	189	0	178	175	180	189	183	184	186	186	186	**189**	210	201	208	194	199
11	March SL	209	0	182	188	192	205	192	191	190	197	204	188	**209**	210	211	210	211
12	March SR	193	0	172	177	180	191	180	180	181	185	189	181	192	**193**	209	202	204
13	March SS	208	0	183	188	190	205	194	191	191	199	203	189	206	192	**208**	208	210
14	WalkColumn	179	0	177	162	169	179	172	174	176	176	176	174	178	170	179	**179**	201
15	WalkRow	182	0	164	180	172	182	173	171	172	178	182	172	180	171	180	160	**182**

the union of the FC of the FP-BTs and compare it with the union of the FC of the other 12 BTs.

4.1 Analysis at HVcc testing

Fig. 2*a* shows the Venn diagram of the FC union of the three FP-BTs when compared with the 12 traditional BTs (Table 2). The total FC is 202. Fig. 2*a* shows that 188 out of 202 faults can be detected with the FP-BTs only, whereas the other 12 BTs detect 200 out of 202 faults. There are 14 faults that are not covered with the FP-BTs; 11 of them are UFs (Table 5). Note that the total number of UFs is 12 and that March SL detects one of them.

Consider now the set of the three BTs shown in Table 5, which detect UFs at HVcc (March SL is excluded); let 'H-UF-BTs' denote this set of BTs (i.e. H-UF-BTs = {GalRow, Hammer, Scan}). The analysis of the FC of H-UF-BTs reveals that the union of their FC is 198 out of 202 faults, as is shown in Fig. 2*b*. In addition, the union of H-UF-BTs with the FP-BTs achieves 100% FC (i.e. 202 from 202). Note that 188 out of 202 faults are covered by the FP-BTs and that the latter detects four faults that are missed by H-UF-BTs. Thus, the FC achieved with the initial test set of 15 BTs can also be achieved with a short test set consisting six BTs: three FP-BTs and three H-UF-BTs.

Any fault detected with FP-BTs can (probably) be explained because these BTs target well predefined faults. However, most detected UFs (by empirical tests) cannot be explained with the well-known fault models; this means that additional faults exist that still should be modelled. The detected UFs call for a detailed analysis in order to understand the defect mechanisms behind them; a deep understanding of the defect mechanisms and their faulty behaviour will allow for modelling the faults and for introducing shorter/optimal BTs that cover such faults.

Table 5: BTs detecting UFs

BT	HVcc		LVcc	
	FC	#UFs	FC	#UFs
GalRow	176	9	—	—
Hammer	171	1	—	—
Scan	168	1	—	—
March SL	185	1	—	—
Total		12		0

4.2 Analysis at LVcc testing

Fig. 3 shows the Venn diagram of the FC of the three FP-BTs when compared with the rest of 12 BTs at LVcc testing. All faults detected by the FP-BTs are also detected by the union of the other 12 BTs; these consist of 213 faults out of 223 (i.e. 95.51%).

As has been shown in Section 3, there are no BTs detecting UFs at LVcc (Table 5). The questions are now what are the faults missed by the FP-BTs and which BTs (from the initial BT set) have to be added to the FP-BTs in order to achieve the complete FC, that is, 223/223. A detailed analysis showed that at least Hammer should be added. The next questions are then what kind of faults a Hammer detects and how they can be modelled. These issues remain still to be worked out.

On the basis of the analysis, one can derive an optimal set of BTs detecting all faults at HVcc, as well as at LVcc (Table 7). Testing at HVcc requires six BTs and at LVcc requires four BTs; four BTs are common. Inspecting the table reveals that some of the BTs are empirical tests (e.g.

Table 6: Reduced set of BTs with 100%FC

Number	For HVcc	For LVcc	Type BT
1	GalRow	GalRow	$O(n.\sqrt{n})$
2	Hammer	Hammer	$O(n)$
3	March C−	March C−	$O(n)$
4	March G	March G	$O(n)$
5	—	PMOVI	$O(n)$
6	March RAW	March RAW	$O(n)$
7	Scan	—	$O(n)$
8	March SL	March SL	$O(n)$
9	March SR	March SR	$O(n)$
10	March SS	March SS	$O(n)$

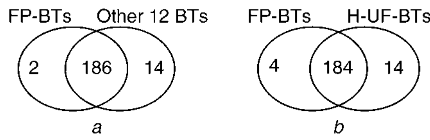

Fig. 2 *FC of the FP-BTs at HVcc*

GalRow, Hammer), not designed to target well-defined fault models. Such tests detect faults that cannot be explained with the well-known fault models; they remain still to be understood and to be modelled. This will allow for developing low-cost fault model BTs. Questions like the following still remain to be answered: (a) what are the defect mechanisms behind faults detected by GalRow and Hammer, (b) how they can be modelled at the functional level, (c) can we develop optimal BTs for such faults and so on.

5 Effect of SCs on the FP-based BTs

As is known, choosing an appropriate stress to be used with a certain BT has a large impact on the FC. Therefore it makes sense to discuss the impact of the SCs on the FC of the three FP-BTs and to find out what are the best SCs that the three FP-BTs have to be used with (from the performed experiment point of view).

A detailed analysis has been done for the impact of the SCs on the FC of the three FP-BTs. The results are summarised in Table 8. It gives the FC of each SC with the three FP-BT and also lists the minimal number of SCs to be used with each of the three FP-BTs in order to achieve the maximal FC. The minimal SCs that have to be used with each FP-BT (in order to achieve 100% FC) is given in bold. For example, at HVcc testing, March SS requires the use of only two SCs: (fy, cs) and (fy, rs). If the number is given within '()', then it means that only one of such SCs is required. For example, March SL at LVcc requires the use of (fx, c), (fx, cs), (fy, s) and one of the following SCs: (fx, rs), (fy, c) or (fy, cs). On the basis of Table 8, we can conclude that:

- Instead of using an initial set of 48 SCs for FP-BTs (i.e. 16 {the # of SCs including HVcc and LVcc} × 3 {the three FP-BTs}), one can only use 20 SCs while achieving the same FC: 7 SCs at HVcc and 13 SCs at LVcc.
- For achieving a 100% FC, the number of SCs required at HVcc is much less than that required at LVcc; for example, March SS requires only two SCs at HVcc, whereas it requires five SCs at LVcc (see last row of Table 8).
- Generally speaking, using FP-BTs with fy addressing scores better than with fx addressing.
- A special analysis done showed the following:

  – for March SS: (fy, rs) covers (fx, c), (fx, cs), (fx, rs) and (fx, s) at HVcc testing.

  – for March RAW: (fy, rs) covers (fy, s) irrespective of the voltage at which testing is done. In addition, at HVcc testing (fy, rs) also covers (fx, cs) and (fy, cs).

  – for March SL: (fx, cs) covers (fx, rs), and (fy, cs) covers (fy, c) at HVcc testing.

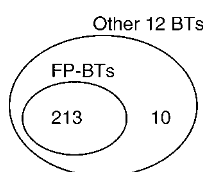

Fig. 3 *FC of FP-BTs at LVcc*

Table 7: Optimal BTs set achieving 100% FC

BT	TL	HVcc		LVcc	
		BT	FC	BT	FC
GalRow	$6n+4nR$	+	176	–	–
Hammer	$49n$	+	171	+	199
March RAW	$26n$	+	184	+	208
Scan	$4n$	+	168	–	–
March SL	$41n$	+	185	+	209
March SS	$22n$	+	184	+	208

6 Optimal test set and SCs

For the performed memory test experiment, it has been shown in Section 4 that in order to achieve the same FC as that of the initial 15 BTs (with a total of 122 SCs), only a minimal set of six BTs is required (Table 6). In order to obtain an idea about the impact of selecting appropriate SCs on the overall test time while keeping the same FC, the minimal number of SCs that have to be used with the minimal test set (i.e. six BTs) will be presented.

Table 9 gives the SCs needed to be used with each of the six BTs. The column 'TT/SC' lists the test time of each BT per SC; the column '#SC' gives the number of SCs each BT has to be used with at HVcc and LVcc; for example, March SS has to be used with two SCs at HVcc and five SCs at LVcc. An 'HL' in the table denotes that the SC is used both at HVcc and LVcc, an 'L' only at LVcc, an 'H' only at HVcc and a '—' not used. For example, Hammer is used only with (fx, s) at HVcc and LVcc. The minimal number of SCs required to achieve the FC achieved with the initial 122 SCs is only 26: 12 SCs at HVcc and 14 SCs at LVcc. Note that Scan was initially used with four SCs at HVcc and LVcc. However, the impact of the stress on the FC at HVcc showed that only three SCs are required in order to achieve the same FC. At LVcc, Scan is not required (Table 7). The required test time for the initial test set was 160.942 ms/chip; however with the optimal test set, the required test time is just 30.498 ms/chip (i.e. a reduction factor of 5.3).

This clearly indicates the importance of test optimisation and the overall test time reduction. Optimising the test set means, in addition to selecting appropriate BTs, also selecting the minimal number of SCs that has to be associated with each BT in order to achieve the maximal FC. Unfortunately so far, there is no theoretical model to correlate the FC with SCs for a given BT.

Table 8: Minimal set of SCs for FP-BTs

SC	March SS		March RAW		March SL	
	HVcc	LVcc	HVcc	LVcc	HVcc	LVcc
(fx, c)	171	**186**	175	**183**	175	**188**
(fx, cs)	170	**187**	**(178)**	196	**176**	**191**
(fx, rs)	171	**(193)**	**181**	**200**	173	**(191)**
(fx, s)	170	183	175	184	**174**	183
(fy, c)	179	188	**(175)**	**185**	179	**(187)**
(fy, cs)	**180**	**(195)**	**(178)**	**197**	182	**(196)**
(fy, rs)	**182**	**195**	**(181)**	196	**(180)**	193
(fy, s)	177	**199**	**(177)**	175	**(179)**	**199**
Total	2	5	2	4	3	4

Table 9: List of minimal SCs

Number	BT	TT/SC, ms	#SC		SC							
					fx				fy			
			HVcc	LVcc	s	c	cs	rs	s	c	cs	rs
1	GalRow	1.556	1	0	–	–	–	–	H	–	–	–
2	Hammer	2.000	1	1	HL	–	–	–	–	–	–	–
3	March RAW	1.065	2	4	–	L	–	HL	–	L	HL	–
4	Scan	0.164	3	0	–	–	H	H	–	H	–	–
5	March SL	1.679	3	4	H	L	HL	–	–	–	L	HL
6	March SS	0.901	2	5	–	L	L	–	L	–	HL	HL

7 Conclusions

This paper summarises the results of applying 15 BTs, each with up to 16 SCs, resulting in 122 tests, to advanced Intel SRAMs. The 15 BTs consist of 12 well-known tests and three FP-based tests. The following conclusion can be drawn for the memory under consideration.

- SCs have a large impact on the FC of the BTs. This is because tests detect not only the faults they are designed for but also some other unknown complex faults that are stress related. No theoretical model exists to correlate the used stress and the predicted FC.
- Testing at low voltage is more effective in terms of FC: the number of faults detected only at low voltage is much higher than the number of faults detected only at high voltage. However, testing at high voltage is more effective in detecting UFs. In order to achieve a high FC, testing needs to be done at both voltage levels.
- Testing at low voltage is shown to be very important for the detection of resistance bridges. When testing at low voltage, one can detect resistance bridges with up to five times higher resistance than that can be detected at nominal voltage [33, 34].
- Testing at high voltage is shown to be more important for the detection of resistive opens [30, 35]. The faulty behaviour of resistive opens manifests itself mainly as a dynamic fault, that is, time-related faults.
- For the memory under consideration, the sensitivity to the bridges tends to be higher than that to opens. This can be explained by the high FC achieved by LVcc testing when compared with HVcc testing.
- The three FP-based tests (i.e. March SS, March SL and March RAW) score overall much better than the traditional tests. In addition, the FP-BTs score better when used with ‘fast y’ rather than ‘fast x’ addressing. ‘Fast x’ addressing is more suitable for the detection of row address decoder faults (AFs) and faults in the peripheral circuits (i.e. sense amplifiers, precharge circuits, mux’s etc.) [37]. That ‘fast y’ scores better than ‘fast x’ in our experiment can be explained by the fact that such faults are less important for the memory under consideration than the memory cell array faults and the column AFs (which require ‘fast y’ for their detection).
- The best test, in terms of detecting UFs, is GalRow. However, it detects considerably fewer faults. GalRow is powerful in detecting dynamic faults (i.e. time-related faults) due to resistive open defects either in the column decoder or in the memory cell array single rows (for example, a resistive defect at a via shared by two adjacent cells in the same row) [37].
- To reduce the test time, the non-linear tests have to be eliminated. This requires a better understanding of the detected (unique) faults such that they can be modelled and linear optimised tests can be designed. Random testing methods may be an alternative solution for the detection of not yet modelled faults for complex memories of small sizes [38–40].
- The tests with the most promising FC, based on what could be expected from the theory, also have the highest FC in practice. However many detected faults still leave much to be explained.

It is important to note that these results are, of course, design dependent. Even if the core of memory under consideration is based on the standard 6T cells, other design implementations (i.e. layouts) and/or fabrication processes will have most likely different results. For example, one memory layout can be more sensitive to bridges between cells belonging to the same row and another layout can be more sensitive to bridges (and/or opens) between cells in the same column. Also, because the size of the experiment is limited, some effects may not have been noticed. Therefore the validity of data cannot be generalised. The following issues still remain to be resolved.

- Understand the defect mechanisms behind the detected UFs. This will allow for modelling them at the functional level and for developing optimal tests.
- As the memories grow in size and speed, the lines carrying those signals will have, in addition to a high load, a high parasitic capacitance. This increases their sensitivity for delay- and timing-related faults because of their capacitive coupling with other signals, power and ground lines and so on. Moreover, the significance of the resistive opens is considered to increase in current and future technologies [41], not only because of the copper wiring but also the presence of many, long interconnections and the growing number of metal layers and vias. As the partial resistive opens behave as delay- and time-related faults, these faults will become more important in the new memory technologies [36, 42–44].

8 References

1 Allan, A., *et al.*: ‘International technology roadmap for semiconductors (ITRS)’, *Computers*, 2002, pp. 42–53

2 Breuer, M.A., and Friedmzan, A.D.: ‘Diagnosis and reliable design of digital systems’ (Computer Science Press, Woodland Hills, CA, USA, 1976)

3 van de Goor, A.J.: ‘Testing semiconductor memories, theory and practice’ (ComTex Publishing, Gouda, The Netherlands, 1998, 2nd edn.)

4 Marinescu, M.: ‘Simple and efficient algorithms for functional RAM testing’. Proc. Int. Test Conf., 1982, pp. 236–239

5 Nair, R.: ‘An optimal algorithm for testing stuck-at faults random access memories’, *IEEE Trans. Comput.*, 1979, **C-28**, (3), pp. 258–261

6 Suk, D.S., and Reddy, S.M.: 'A march test for functional faults in semiconductors random-access memories', *IEEE Trans. Comput.*, 1980, **C-29**, (6), pp. 419–429
7 Papachristou, C.A., and Saghal, N.B.: 'An improved method for detecting functional faults in random access memories'. *IEEE Trans. Comput.*, 1985, **C-34**, (2), pp. 110–116
8 Suk, D.S., and Reddy, S.M.: 'Test procedure for a class of patterns sensitive faults in semiconductors random-access memories', *IEEE Trans. Comput.*, 1981, **C-30**, (12), pp. 982–985
9 Maly, W.: 'Modeling of lithography related yield losses for CAD of VLSI circuits', *IEEE Trans. CAD*, 1985, **4**, (3), pp. 166–177
10 Shen, J.P.: 'Inductive fault analysis of CMOS integrated circuits', *IEEE Des. Test Comput.*, 1985, **2**, (6), pp. 13–26
11 Dekker, R., Beenker, F., and Thijssen, H.: 'A realistic fault models and test algorithms for static random access memories', *IEEE Trans. Comput.*, 1990, **C9**, (6), pp. 567–572
12 Adams, R.D., and Cooley, E.S.: 'False write through and un-restored write electrical level fault models for SRAMs'. Proc. IEEE Int. Workshop on Memory Technology, Design, and Testing, 1997, pp. 27–32
13 Abadir, M.S., and Reghbati, J.K.: 'Functional testing of semiconductor random access memories', *ACM Comput. Surv.*, 1983, **15**, (3), pp. 175–198
14 Adams, R.D., and Cooley, E.S.: 'Analysis of a deceptive read destructive memory fault model and recommended testing'. Proc. IEEE North Atlantic Test Workshop, 1996
15 Adams, R.D.: 'High performance memory testing' (Kluwer Academic Publisher, 2003) ISBN: 1-4020-7255-4
16 De Jonge, J.H., and Smeulders, A.J.: 'Moving inversions test pattern is thorough, yet speedy', *Comput. Des.*, 1976, pp. 169–173
17 Hamdioui, S., and van de Goor, A.J.: 'Experimental analysis of spot defects in SRAMs: realistic fault models and tests'. Proc. Ninth Asian Test Symp., 2000, pp. 131–138
18 Goto, S.N.H., and Iwasaki, K.: 'Experiment fault analysis of 2 MB SRAM chips'. Proc. Design and Test Europe Conf., 1999, pp. 623–630
19 Kim, V.K., and Chen, T.: 'On comparing functional fault coverage and defect coverage for memory testing', *IEEE Trans. CAD*, 1999, **18**, (11), pp. 1676–1683
20 Schanstra, I., and van de Goor, A.J.: 'Industrial evaluation of stress combinations for march tests applied to SRAMs'. Proc. IEEE Int. Test Conf., 1999, pp. 983–992
21 van de Goor, A.J., and de Neef, J.: 'Industrial evaluation of DRAMs tests'. Proc. Design Automation and Test in Europe, 1999, pp. 623–630
22 van de Goor, A.J., and Paalvast, A.: 'Industrial evaluation of DRAMs SIMM tests'. Proc. Int. Test Conf., 2000, pp. 426–435
23 Al-ars, Z., van de Goor, A.J.: 'Static and dynamic behavior of memory cell array spot defects in embedded DRAMs', *IEEE Trans. Comput.*, 2003, **52**, (3), pp. 293–309
24 van de Goor, A.J., and Al-Ars, Z.: 'Functional fault models: a formal notation and taxonomy'. Proc. IEEE VLSI Test Symp., 2000, pp. 281–289
25 Hamdioui, S., Al-ars, Z., and van de Goor, A.J.: 'Testing static and dynamic faults in random access memories'. Proc. IEEE VLSI Test Symp., 2002, pp. 395–400
26 Al-ars, Z., Hamdioui, S., and van de Goor, A.J.: 'A fault primitive based analysis of linked faults in RAMs'. Proc. IEEE Int. Workshop on Memory Technology, Design, and Testing, 2003, pp. 33–28
27 Hamdioui, S., Al-ars, Z., van de Goor, A.J., and Rodgers, M.: 'Linked faults in random-access-memories: concept, fault models, test algorithms and industrial results', *IEEE Trans. CAD Integr. Circuits Syst.*, 2004, **23**, (5), pp. 737–757
28 Al-Ars, Z., and van de Goor, Ad J.: 'Impact of memory cell array bridges on the faulty behavior in embedded DRAMs'. Records of Asian Test Symp., 2000, pp. 282–289
29 Al-ars, Z., and van de Goor, Ad J.: 'Static and dynamic behavior of memory cell array opens and shorts in embedded DRAMs'. Proc. Design Automation and Test in Europe, 2001, pp. 496–503
30 Borri, S., *et al*.: 'Defect oriented dynamic fault models for embedded SRAMs'. Proc. European Test Symp., 2003, pp. 23–28
31 Hamdioui, S., van de Goor, A.J., and Rodgers, M.: 'March SS: a test for all static simple RAM faults'. Proc. IEEE International Workshop on Memory Technology, Design, and Testing, 2002, pp. 95–100
32 Hamdioui, S.: 'Testing static random access memories: Defects, fault models and test patterns' (Kluwer Academic Publishers, Boston, 2004) ISBN 1-4020-7752-1
33 Chang, J.Y-Y., and McCluskey, E.J.: 'Detecting delay faults by very-low-voltage testing'. Proc. Int. Test Conf., 1996, pp. 367–376
34 Kruseman, B., *et al*.: 'Systematic defects in deep-submicron technologies'. Proc. Int. Test Conf., 2002, pp. 290–299
35 Majhi, A.K., *et al*.: 'Memory testing under different stress conditions: an industrial evaluation'. Proc. Design, Automation and Test in Europe (DATE), 2005, pp. 438–443
36 Azimane, M., and Majhi, A.K.: 'New test methodology for resistive open defect detection in memory address decoders'. Proc. IEEE VLSI Test Symp., 2004, pp. 123–128
37 van de Goor, A.J., and Hamdioui, S.: 'Detecting faults in peripheral circuits and an evaluation of SRAM tests'. Proc. IEEE Int. Test Conf., USA, October 2004, pp. 114–134
38 Karaimi, F., Meyer, F.J., and Lombardi, F.: 'Random testing of multi-port static random access memories'. Proc. IEEE Int. Workshop on Memory Technology, Design, and Testing, 2002, pp. 101–106
39 Otterstedt, J., Niggemeyer, D., and Williams, T.W.: 'Detection of CMOS address decoder open faults with March and pseudo random memory tests'. Proc. Int. Test Conf., 1998, pp. 53–62
40 van de Goor, A.J., and Lin, M.: 'The implementation of pseudo-random memory tests on commercial memory testers'. Proc. IEEE Int. Test Conf., 1997, pp. 226–235
41 Kundu, S., *et al*.: 'Test challenges in nanometer technologies', *J. Electron. Test. Theory Appl.*, 2001, **17**, (3/4), pp. 209–218
42 Dilillo, L., *et al*.: 'March iC-: an improved version of March C- for ADOFs detection'. Proc. IEEE VLSI Test Symp., 2004, pp. 129–134
43 Klaus, M., and van de Goor, Ad.J.: 'Tests for resistive and capacitive defects in address decoders'. Proc. Asian Test Symp., 2001, pp. 31–36
44 Nigh, P., and Gattiker, A.: 'Test method evaluation experiments & Data'. Proc. IEEE Int. Test Conf., 2000, pp. 454–463