An Industrial Evaluation of DRAM Tests

Ad J. van de Goor
Delft University of Technology

This application of 40 well-known memory tests to 1,896 1-Mbyte × 4 DRAM chips, used up to 48 different stress combinations with each test. The results show the importance of selecting the right stress combination, and that the theoretically better tests—those covering more different functional faults—also have higher fault coverage.

DRAM PRODUCTION tests are currently necessary to reach a defect-per-million level that approaches the single-digit numbers. This implies that a single memory test is insufficient; rather, a set of tests is necessary. In addition, to obtain economically acceptable test times, test engineers must often manually optimize the test sets for the particular technology used. Consequently, researchers have devoted much time to designing memory tests optimized to detect a particular class of faults.[1-4] These tests might have an academic origin or are based on Spice simulation, and possibly also on inductive fault analysis. The effectiveness of these tests remains a question.

A test consists of a base test—an algorithm—applied using a particular stress combination (SC). An SC consists of a combination of values for the different stresses, for example, V_{DD} = 4.5 V and *Temp* = 70° C. In previous work, researchers have performed base tests with up to 48 SCs,[5] reporting the results of testing 1,024, 128-Kbyte × 8 SRAM chips using a few tests. These results indicated that the fault coverage depended heavily on the stresses used, such as the load used on the output pins or the power supply voltage. Earlier work applied a small set of march base tests, using many SCs, to 2-Kbyte × 8 SRAMs; this work also indicated that SCs were very important.[6]

This article presents the results of an initial test set consisting of 40 tests, applied to 1,896 Fujitsu 1-Mbyte × 4 fast-page-mode DRAM chips; I base it on work presented at the 1999 Conference on Design Automation and Test in Europe.[7] In a later experiment,[8] a colleague and I applied a different set of base tests and SCs to single inline memory modules, with results that are comparable with those of this work.

Used base tests and stresses

I describe the base tests and the stresses that compose an SC. I then give an overview of the tests built from the base tests and SCs.

Base tests

The base tests we used for this set of experiments consisted of four classes: electrical, march, base cell, and repetitive. The number that follows each class name (in parentheses) is the number of base tests in that class.

The notation for the base tests uses the following symbols:

- ⇑ denotes an increasing address order (that is, address 0, 1, 2, and so on),
- ⇓ denotes a decreasing address order, and
- ⇕ denotes that the test engineer can arbitrarily choose the address order to be ⇑ or ⇓.

I list the required test time after the name of the base test; it is expressed as some function of *n,* the number of memory words. It can also rely on the settling time, t_s = 5 ms.

Class 1: Electrical tests (11). The class of electrical base tests consists of checks for contact (1), DC parameters (7), and AC parameters (3).

1. Contact verifies the connection between the device under test (DUT) and the memory tester.
2. InpLkH, input leakage-current high, verifies $I_{I(L)-max}$.
3. InpLkL, input leakage-current low, verifies $I_{I(L)-min}$.
4. OutpLkH, output leakage-current high, verifies $I_{O(L)-max}$.
5. OutpLkL, output leakage-current low, verifies $I_{O(L)-min}$.

0740-7475/04/$20.00 © 2004 IEEE Copublished by the IEEE CS and the IEEE CASS

6. ICC1, operating current, verifies I_{CC1}.
7. ICC2, standby current, verifies I_{CC2}.
8. ICC3, refresh current, verifies I_{CC3}.
9. DataRet, data retention ($4n + 6t_s$): {⇑(*wDh*); *V*cc ← *V*cc-min; *Del*; *V*cc ← *V*cc-typ; ⇑(*rDh*)}. In the preceding test description, $Del = 1.2t_{REF}$, where t_{REF} is the refresh time. Repeat this test for $\overline{Dh}$; *Dh* is the checkerboard data background, as discussed in a following section about the stresses.
10. Volatility ($6n + 6t_s$): {⇑(*wDh*); Vcc ← Vcc-min; ⇑(*rDh*); Vcc ← Vcc-typ; ⇑(*rDh*)}. Repeat this test for $\overline{Dh}$.
11. VccRW ($8n + 6t_s$): {Vcc ← Vcc-max; ⇑(*wd*); Vcc ← Vcc-min; ⇑(*rd*); ⇑(*wd*); Vcc ← Vcc-max; ⇑(*rd*)}. Repeat this test for $\overline{d}$, which is some data background.

Class 2: March tests (18). March tests are very popular tests for detecting functional faults such as address decoder and coupling faults.[4] We performed some tests twice; the second time included an extra read operation added to each march element of that test. This applies to

- MarchC-Rb, where suffix Rb denotes the extra read operation at the beginning of each march element of MarchC-;
- PMOVI-Re, where suffix Re denotes an extra read at the end of each march element of PMOVI; and
- MarchU-Rm, with an extra read in the middle of each march element of MarchU.

D denotes the delay time for data retention faults.[4] Given these descriptions, the tests in this class are as follows:

12. Scan ($4n$): {⇕(*w*0); ⇕(*r*0); ⇕(*w*1); ⇕(*r*1)}
13. MATS+ ($5n$): {⇕(*w*0); ⇑(*r*0, *w*1); ⇓(*r*1, *w*0)}
14. MATS++ ($6n$): {⇕(*w*0); ⇑(*r*0, *w*1); ⇓(*r*1, *w*0, *r*0)}
15. MarchA ($15n$): {⇕(*w*0); ⇑(*r*0, *w*1, *w*0, *w*1); ⇑(*r*1, *w*0, *w*1); ⇓(*r*1, *w*0, *w*1, *w*0); ⇓(*r*0, *w*1, *w*0)}
16. MarchB ($17n$): {⇕(*w*0); ⇑(*r*0, *w*1, *r*1, *w*0, *r*0, *w*1); ⇑(*r*1, *w*0, *w*1); ⇓(*r*1, *w*0, *w*1, *w*0); ⇓ (*r*0, *w*1, *w*0)}
17. MarchC- ($10n$): {⇕ (*w*0); ⇑(*r*0, *w*1); ⇑(*r*1, *w*0); ⇓(*r*0, *w*1); ⇓ (*r*1, *w*0); ⇕(*r*0)}
18. MarchC-Rb ($15n$): {⇕(*w*0); ⇑(*r*0, *r*0, *w*1); ⇑(*r*1, *r*1, *w*0); ⇓(*r*0, *r*0, *w*1); ⇓ (*r*1, *r*1, *w*0); ⇕(*r*0, *r*0)}
19. PMOVI ($13n$): {⇓(*w*0); ⇑(*r*0, *w*1, *r*1); ⇑(*r*1, *w*0, *r*0); ⇓(*r*0, *w*1, *r*1); ⇓(*r*1, *w*0, *r*0)}
20. PMOVI-Re ($17n$): {⇓(*w*0); ⇑(*r*0, *w*1, *r*1, *r*1); ⇑(*r*1, *w*0, *r*0, *r*0); ⇓(*r*0, *w*1, *r*1, *r*1); ⇓(*r*1, *w*0, *r*0, *r*0)}
21. MarchG ($23n + 2D$): {⇕(*w*0); ⇑(*r*0, *w*1, *r*1, *w*0, *r*0, *w*1); ⇑(*r*1, *w*0, *w*1); ⇓(*r*1, *w*0, *w*1, *w*0); ⇓(*r*0, *w*1, *w*0); *D*; ⇕(*r*0, *w*1, *r*1); *D*; ⇕(*r*1, *w*0, *r*0)}
22. MarchU ($13n$): {⇕(*w*0); ⇑(*r*0, *w*1, *r*1, *w*0); ⇑(*r*0, *w*1); ⇓(*r*1, *w*0, *r*0, *w*1); ⇓(*r*1, *w*0)}
23. MarchUD ($13n + 2D$): {⇕(*w*0); ⇑(*r*0, *w*1, *r*1, *w*0); *D*; ⇑(*r*0, *w*1); *D*; ⇓(*r*1, *w*0, *r*0, *w*1); ⇓(*r*1, *w*0)}
24. MarchU-Rm ($15n$): {⇕(*w*0); ⇑(*r*0, *w*1, *r*1, *r*1, *w*0); ⇑(*r*0, *w*1); ⇓(*r*1, *w*0, *r*0, *r*0, *w*1); ⇓(*r*1, *w*0)}
25. MarchLR ($14n$): {⇕(*w*0); ⇓(*r*0, *w*1); ⇑(*r*1, *w*0, *r*0, *w*1); ⇑(*r*1, *w*0); ⇑(*r*0, *w*1, *r*1, *w*0); ⇓(*r*0)}
26. MarchLA ($22n$): {⇕(*w*0); ⇑(*r*0, *w*1, *w*0, *w*1, *r*1); ⇑(*r*1, *w*0, *w*1, *w*0, *r*0); ⇓(*r*0, *w*1, *w*0, *w*1, *r*1); ⇓(*r*1, *w*0, *w*1, *w*0, *r*0); ⇓(*r*0)}
27. MarchY ($8n$): {⇕(*w*0); ⇑(*r*0, *w*1, *r*1); ⇓(*r*1, *w*0, *r*0); ⇕(*r*0)}
28. XMOVI ($17n \times \log_2 n$): Repeat PMOVI for X-address increment = 2^i ($0 \le i \le 9$).
29. YMOVI ($17n \times \log_2 n$): Repeat PMOVI for Y-address increment = 2^i ($0 \le i \le 9$).

Class 3: Base-cell tests (6). We designed this class of tests to detect the influence of a disturbance of the base cell on other cells, or vice versa.[4] They use the following notation:

- The term *row* denotes the address incrementing along a row of the base cell, skipping the base cell.
- The term *col* denotes the address incrementing along a column of the base cell, skipping the base cell.
- The symbol ◇ indicates addressing of the north (N), east (E), south (S), or west (W) neighbors of the base cell.
- The notation $w1_b$ indicates a *w*1 in the base cell.
- The notation $r0_b$ indicates an *r*0 from the base cell.

30. Butterfly ($14n$): {⇑(*w*0); ⇑($w1_b$, ◇(*r*0), $w0_b$); ⇑(*w*1); ⇑($w0_b$, ◇(*r*1), $w1_b$)}
31. GalCol ($2n + 4n\sqrt{n}$): {⇑(*w*0); ⇑($w1_b$, *col*(*r*0, $r1_b$), $w0_b$); ⇑(*w*1); ⇑($w0_b$, *col*(*r*1, $r0_b$), $w1_b$)}
32. GalRow ($2n + 4n\sqrt{n}$): {⇑(*w*0); ⇑($w1_b$, *row*(*r*0, $r1_b$), $w0_b$); ⇑(*w*1); ⇑($w0_b$, *row*(*r*1, $r0_b$), $w1_b$)}
33. WalkCol ($6n + 2n\sqrt{n}$): {⇑(*w*0); ⇑($w1_b$, *col*(*r*0), $r1_b$, $w0_b$); ⇑(*w*1); ⇑($w0_b$, *col*(*r*1), $r0_b$, $w1_b$)}
34. WalkRow ($6n + 2n\sqrt{n}$): {⇑(*w*0); ⇑($w1_b$, *row*(*r*0), $r1_b$, $w0_b$); ⇑(*w*1); ⇑($w0_b$, *row*(*r*1), $r0_b$, $w1_b$)}
35. SlidDiag ($4n\sqrt{n}$): For each diagonal element, denoted by d, perform the test for each diagonal, denoted by $⇑_d$: {⇑(*w*0); $⇑_d$($w1_d$, *r*, $w0_d$); ⇑(*w*1); $⇑_d$($w0_d$, *r*, $w1_d$)}

Class 4: Repetitive tests (3). Repetitive tests perform multiple read or write operations to a single cell; denoted by rx^y or rw^y, these tests repeat the rx or wx operation y times. Repeating the tests makes partial fault effects become full fault effects. The $\nearrow$ denotes an address increment along the main diagonal.

36. HammerR ($40n$): {$\Uparrow(w0)$; $\Uparrow(r0, w1, r1^{16}, w0)$; $\Uparrow(w1)$; $\Uparrow(r1, w0, r0^{16}, w1)$}
37. Hammer ($4n + 2002\sqrt{n}$): {$\Uparrow(w0)$; $\nearrow(w1_b^{1000}, row(r0), r1_b, col(r0), r1_b, w0_b)$; $\Uparrow(w1)$; $\nearrow(w0_b^{1000}, row(r1), r0_b, col(r1), r0_b, w1_b)$}
38. HammerW ($4n + 2\sqrt{n}$): {$\Uparrow(w0)$; $\nearrow(w1_b^{16}, col(r0), w0_b)$; $\Uparrow(w1)$; $\nearrow(w0_b^{16}, col(r1), w1_b)$}

Stresses

A stress can be a refinement of a certain operation of a base test (such as the address order or the data to be written), or it can be an external condition applied to the DUT with the intent of making faults easier to detect. So the latter group includes timing, voltage, temperature, and load stresses. The work discussed here uses only typical values for these loads, equivalent to two transistor-transistor-logic loads and 100 pF. Note that all tests have scrambling enabled; that is, they permit the rearrangement of addresses and data bits so that logically adjacent addresses and data bits are also topologically adjacent.[9]

Address stress. The base tests used four types of address stresses:

- *Ax,* Fast *X,* increments row address first; the notation ${}_x\Uparrow$ or ${}_x\Downarrow$ denotes this stress.
- *Ay,* Fast *Y,* increments column address first; the notation ${}_y\Uparrow$ or ${}_y\Downarrow$ denotes this stress.
- *Ac,* address complement, is an alternating sequence of an address and its complement, for example, the address sequence 000, 111, 001, 110, 010, 101, 011, 100.
- *Ai*, increment 2^i, works with the MOVI tests.[4,10]

Data background stress. The base tests used four types of data background stresses:

- *Ds* indicates a solid data background of all zeros or all ones.
- *Dh* indicates a checkerboard data background, that is, the alternating sequences 01010 … and 10101 … .
- *Dr* indicates a row stripe data background; that is, a row of all zeros followed by a row of all ones, or vice versa.
- *Dc* indicates a column stripe data background, of either 0101 … or 1010 … .

Timing stress. The timing stresses used here are the maximum or minimum of a range:

- *S*–, the MinTime stress, uses minimum time t_{RCD}, where subscript RCD stands for RAS to CAS delay.
- Similarly, *S*+, the MaxTime stress, uses maximum time t_{RCD}.
- *Sl* denotes the long cycle stress, which uses $t_{RAS-max}$ (for DRAMs, typically 10 ms) and the minimum t_{RCD}.

Voltage stress. The two stresses use a minimum V_{cc} of 4.5 V (denoted *V*–) and a maximum of 5.5 V (denoted *V*+).

Temperature stress. The tests used two levels of temperature stresses, one at room temperature, 25° C, denoted as RT. The second stress is at a high temperature, 70° C, denoted as HT.

Overview of used tests

Table 1 lists the tests, which together form the initial test set (ITS); for this overview, only the first five columns are of interest. The first two columns list the number and name of the base test. (This is the same number as used in the earlier descriptions of the base tests.) Group is the group that a base test belongs to. The fourth column shows the execution time required for the corresponding test. The fifth column lists the number of SCs that we used to perform the corresponding base test, both for the room temperature (RT) and the high-temperature (HT) stresses. The total execution time for a particular base test is the product of the two corresponding column entries, that is, time × (the number of SCs). For example, for MATS+, (0.58 s) × 48 = 27.84s.

The XMOVI and YMOVI tests are identical to the PMOVI test; however, we repeat them for a number of times equal to the number of *x* or *y* address bits. With each repetition, the address incrementing or decrementing takes place with a different value of 2^i. For example, for a 3-bit *x* address and $i = 1$, the increment is $2^1 = 2$ and the resulting address sequence is 000; 010; 100; 110; 001; 011; 101; 111.[4,10]

Furthermore, base tests 39 and 40 (ScanL and

Table 1. Fault coverage for unions and intersections of base tests and SCs.

					Fault coverage											
					Union		Intersection		V–		V+		Ax		Ay	
No.	Base test	Group	Time(s)	SC	RT	HT	RT	HT	RT	HT	RT	HT	RT	HT	RT	HT
1	Contact	0	0.02	1	80	55	80	55	80	55	0	0	80	55	0	0
2	InpLkH	1	0.02	1	61	32	61	32	61	32	0	0	61	32	0	0
3	InpLkL	1	0.02	1	46	28	46	28	46	28	0	0	46	28	0	0
4	OutpLkH	1	0.02	1	4	0	4	0	4	0	0	0	4	0	0	0
5	OutpLkL	1	0.02	1	6	7	6	7	6	7	0	0	6	7	0	0
6	ICC1	2	0.04	1	6	0	6	0	6	0	0	0	6	0	0	0
7	ICC2	2	0.04	1	19	16	19	16	19	16	0	0	19	16	0	0
8	ICC3	2	0.04	1	6	0	6	0	6	0	0	0	6	0	0	0
9	DataRet	3	0.49	4	75	36	54	26	54	35	68	34	75	36	0	0
10	Volatility	3	0.72	4	72	38	53	29	70	38	71	36	72	38	0	0
11	VccRW	3	0.95	4	69	36	54	29	67	36	68	36	69	36	0	0
12	Scan	4	0.46	48	144	118	30	22	124	89	128	113	75	39	120	114
13	MATS+	5	0.58	48	211	152	39	23	197	120	182	138	108	49	184	147
14	MATS++	5	0.69	48	215	140	39	23	203	119	183	124	111	49	189	137
15	MarchA	5	1.73	48	222	157	39	23	206	125	193	143	119	48	202	155
16	MarchB	5	1.96	48	232	157	40	24	214	123	196	146	121	50	210	154
17	MarchC-	5	1.15	48	234	163	39	23	215	126	200	151	119	51	213	159
18	MarchC-Rb	5	1.73	32	213	158	41	23	195	128	185	143	123	50	205	155
19	PMOVI	5	1.50	48	201	144	40	23	185	120	178	137	105	47	170	142
20	PMOVI-Re	5	1.96	32	208	261	42	23	187	213	189	246	107	48	192	259
21	MarchG	5	2.69	48	230	159	40	23	208	131	206	140	124	48	205	157
22	MarchU	5	1.50	48	234	165	42	23	219	132	201	147	133	48	210	163
23	MarchUD	5	1.53	48	243	165	43	23	224	133	213	146	140	49	221	164
24	MarchU-Rm	5	1.73	32	217	160	42	23	200	100	197	156	133	49	204	1,158
25	MarchLR	5	1.61	48	235	173	42	24	217	138	209	160	130	50	216	171
26	MarchLA	5	2.54	48	241	158	41	24	216	126	210	147	125	51	220	154
27	MarchY	5	0.92	48	267	168	40	24	250	136	212	140	116	48	240	167
28	XMOVI	6	14.99	16	256	291	74	69	226	241	237	288	256	291	0	0
29	YMOVI	6	14.99	16	213	217	87	54	195	178	195	212	0	0	213	217
30	Butterfly	7	1.61	16	103	36	43	24	101	35	85	36	103	36	0	0
31	GalCol	7	472.68	1	53	27	53	27	0	0	53	27	53	27	0	0
32	GalRow	7	472.68	1	96	45	96	45	0	0	96	45	96	45	0	0
33	WalkCol	7	236.92	1	55	28	55	28	0	0	55	28	55	28	0	0
34	WalkRow	7	236.92	1	100	47	100	47	0	0	100	47	100	47	0	0
35	SlidDiag	7	472.45	1	95	32	95	32	0	0	95	32	95	32	0	0
36	HammerR	8	4.61	16	115	39	38	23	111	37	99	38	115	39	0	0
37	Hammer	8	0.69	16	100	37	41	24	94	36	89	37	100	37	0	0
38	HammerW	8	4.15	16	139	75	43	24	129	52	124	74	139	75	0	0
39	Scan-L	9	42.07	8	313	33	180	25	304	31	283	30	313	33	0	0
40	MarchC-L	9	105.17	8	340	45	241	27	331	33	309	42	340	45	0	0
	Totals				731	475			678	414	617	424	645	364	378	405

MarchC-L) are identical to base tests 12 and 17, except for the use of the long cycle with t_{RAS} = 10 ms (as described in the section on stresses). We have not performed the base tests with all $4 \times 4 \times 3 \times 2 = 96$ SCs

because, for certain tests (such as Contact), using a large number of SCs does not make sense. For other base tests (such as GalCol) the test time would become excessive.

To save test time, we use the *Ai* address stress only for base tests XMOVI and YMOVI, and the *Sl* timing stress only for the base tests Scan and MarchC-. The result is that the total number of applied tests is not 40 × 48, but the sum of all entries of the column SCs, which is 857. The total test time for all tests is Σ (time × number of SCs), which is 4,814 seconds per DUT.

Test results

We applied the tests listed earlier, using two values for the temperature stress: room temperature or 25°C, denoted RT; and a high temperature of 70°C, denoted HT. Of the 1,896 chips (DUTs) tested at RT, 731 failed at least one RT test. Those DUTs passing the RT tests entered the HT tests, except for 25 lost during RT testing because of a jam in the handler. Rather than 1,896 – 731 – 25 = 1,140, all 1,896 DUTs should also have been tested with the HT stress.

We used the Advantest T3332 tester. The total test time to perform all RT tests was 4,814 seconds, or 1 hour and 20 minutes per DUT. Considering the fact that the T3332 tester can test 32 DUTs in parallel, this results in a total test time of: 4,814 × 1,896/(32 × 3,600) = 79.2 hours for the RT tests, and 4,814 × 1,140/(32 × 3,600) = 47.6 hours for the HT tests.

The application of the base tests with the various SCs to the 1,896 DUTs at RT and to the 1,140 DUTs at HT resulted in a large database, which we had to simplify for analysis purposes. Therefore, I have introduced the notions of union and intersection.

I define a fault as a DUT that fails a test. *FC* is the number of different faults detected by a single test, or by some number of tests. The *union* represents the total number of faulty DUTs (faults) detected by a number of tests; the *intersection* represents the set of common faults detected by a number of tests. In the interest of presenting results accurately, I have used both tables and graphs.

Overall test results

In Table 1, the columns to the right of the vertical line show test results per base test. The union and intersection of the *FC*s for the corresponding base tests applied SC times, are listed. The table also gives results for performing a particular base test with an identified stress (for example, *Ay*, denoting the union of the *FC*s of the corresponding base tests using Fast Y addressing).

Because of space limitations, I do not list all 15 stresses, but only the four most interesting ones, *V*– through *Ay*. For example, the *FC* of MarchC- (test 17), for all 48 SCs, is 234 for RT and 163 for HT. The *FC* of all MarchC- tests with stress *V*– = 4.5 V is 215 for RT and 126 for HT. The table totals each column at the bottom. For example, it shows that the total of RT union is 731, which is the total *FC* of all RT tests. Note that this number is greater than the total of union at HT, 475. Similarly, *V*– at RT is 678, which is greater than V+ at RT, 617. *V*– at HT = 414 is less than *V*+ at HT = 424.

From Table 1, we draw the following conclusions:

- The three best RT base tests are (in the Union RT column) MarchC-L (*FC* = 340), ScanL (*FC* = 313), and MarchY (*FC* = 267). The three best HT base tests are XMOVI (*FC* = 291), PMOVI-Re (*FC* = 261), and YMOVI (*FC* = 217).
- The effect of adding extra read operations to the march elements (indicated by the base test suffixes Rb, Rm, or Re) varies. For MarchC-Rb, the *FC* decreased from 234 to 213 for RT, and from 163 to 158 for HT. For MarchU-Rm, the *FC* decreased from 234 to 217 for RT, and from 165 to 160 for HT. For PMOVI-Re, the *FC* increased from 201 to 208 for RT, and from 144 to 261 for HT. It appears that extra read operations only contributed to the *FC* when added to the end of the march elements.
- The effect of delays in the base tests (*Del* = t_{REF} = 16.4 ms) is as follows: For MarchUD, the *FC* increased from 234 to 243 for RT, and remained at 165 for HT. For MarchG (which is identical to MarchB except for the added delays), the *FC* decreased from 232 to 230 for RT, and increased from 157 to 159 for HT. Our preliminary conclusion is that adding delays has a mixed impact on the *FC*.
- The -L tests (tests 39 and 40) applied at HT have a relatively low *FC*, probably because the RT tests had already detected and eliminated DUTs that exhibited leakage.

Figure 1 shows *FC* values per base test for Union RT (the leftmost bar of each pair) and Intersection RT (the rightmost bar of each pair). The figure shows the high *FC* of the -L tests (Scan-L, test 39; and MarchC-L, test 40). The large differences between the unions and intersections per base test indicate the importance of the SC. For example, for MarchY (test 27) the *FC* varies between 250 (for *V*–) and 112 (for *Ac*, not shown in Table 1).

Actually, the *FC* varies even more drastically, between 181 (for *AyDsS+V–Tt*) and 45 (for *AcDcS–V+Tt*, as well as for *AcDcS+V–Tt*), when taking individual SCs into account. (Figure 1 does not record these individual values). The high *FC*s of the moving inversion tests—PMOVI-Re (test 20), XMOVI (test 28), and YMOVI (test 29)—are interesting.

From Figure 1, as well as from the columns Union RT and Intersection RT of Table 1, the large difference between the *FC* of the union and the intersection of each base test is apparent; this means that the SC has a large impact on the *FC*, and that the *FC*s of a given base test, performed with different SCs, have little in common.

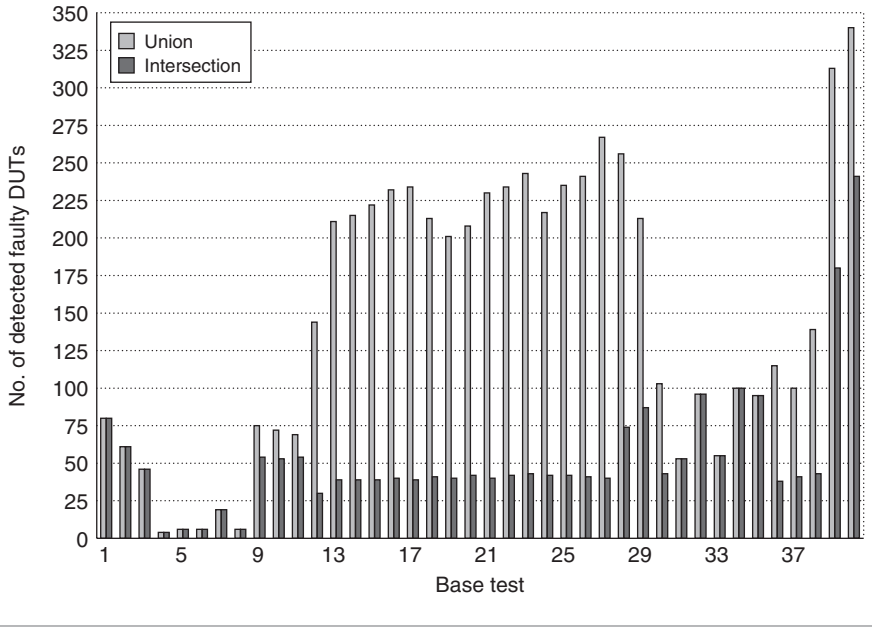

Figure 1. Unions and Intersections per base test.

Room temperature (25° C); total *FC* = 731

	0	1	2	3	4	5	6	7	8	9	Group
	80	58	8	25	26	34	29	25	25	30	0. Contact
		67	8	8	10	19	15	10	11	15	1. Leakage
			19	14	14	15	15	16	15	16	2. ICC
				78	70	74	70	69	69	57	3. Retention
					144	141	125	97	109	75	4. Scan
						372	240	135	142	108	5. March
							282	132	145	102	6. X/Y MOVI
								161	120	85	7. Gal/Walk
									157	88	8. Hammer
										342	9. Long
0. Contact	**55**										
1. Leakage	35	**4**									
2. ICC	6	7	**16**								
3. Retention	22	8	9	**40**							
4. Scan	24	11	9	38	**118**						
5. March	29	15	11	39	117	**383**					
6. X/Y MOVI	24	11	10	37	101	263	**316**				
7. Gal/Walk	22	8	8	36	38	52	47	**55**			
8. Hammer	23	9	8	37	44	69	64	45	**78**		
9. Long	21	6	7	27	25	41	37	29	28	**46**	

High temperature (70° C); total *FC* = 475

Figure 2. Intersections of the unions for test groups.

Analysis of test and stress groups

Figure 2 shows the intersections of the unions for the test groups; the Group column in Table 1 shows the group of each base test. The bold diagonal entries of Figure 2 show the total *FC* of each group; the upper nondiagonal entries show the intersections for RT, and the lower nondiagonal entries are for HT. The largest contributions to the total *FC* of 731 for RT are group 5 (march tests), *FC* = 372; group 9 (-L tests), *FC* = 342; and group 6 (XMOVI and YMOVI tests), *FC* = 282. From the column for group 9, you can see that the -L tests for RT detect 342 faults. Of those, few are detectable by the tests of any other group. You might expect this result because of the unique timing used with the -L tests.

From Figure 2, you can also see that the march tests (group 5) have a total *FC* of 372, and the Scan test (Group 4) has an *FC* of 144; the intersection of groups 4 and 5 is 141, which means that group 5 almost completely covers group 4.

At HT, the largest contributions to the total *FC* of 475 are group 5 (march tests), *FC* = 383; and group 6 (XMOVI and YMOVI tests), *FC* = 316. From Figure 2, you can see that the march tests (group 5) have a total *FC* = 383, and the Scan tests (group 4) have an *FC* = 118. The intersection of groups 4 and 5 is 117, which means that the march tests almost completely cover the Scan test.

Figure 3 shows the intersections of the total *FC* for

Room temperature (25° C); total *FC* = 731

	V–	*V+*	*S–*	*S+*	*Ds*	*Dh*	*Dr*	*Dc*	*Ax*	*Ay*	*Ac*	Stress
	678	564	436	609	634	502	479	443	599	360	139	*V–*
		617	394	601	559	487	467	461	563	335	131	V+
			470	394	418	308	274	250	400	353	139	*S –*
V–	**414**			**655**	592	493	486	466	585	354	131	*S+*
V+	363	**424**			**652**	478	467	440	587	346	139	*Ds*
S–	398	416	**452**			**519**	427	423	476	284	114	*Dh*
S+	374	410	404	**427**			**496**	421	481	260	134	*Dr*
Ds	295	280	309	285	**323**			**475**	471	231	121	*Dc*
Dh	298	301	303	299	207	**308**			**645**	292	139	*Ax*
Dr	298	322	319	320	214	254	**331**			**378**	135	*Ay*
Dc	246	258	257	253	185	228	216	**260**			**140**	*Ac*
Ax	333	327	360	324	271	273	270	239	**364**			
Ay	356	388	384	392	279	281	315	234	294	**405**		
Ac	40	51	51	46	51	41	40	40	51	51	**51**	

High temperature (70° C); total *FC* = 475

Figure 3. Intersections of the unions for stresses.

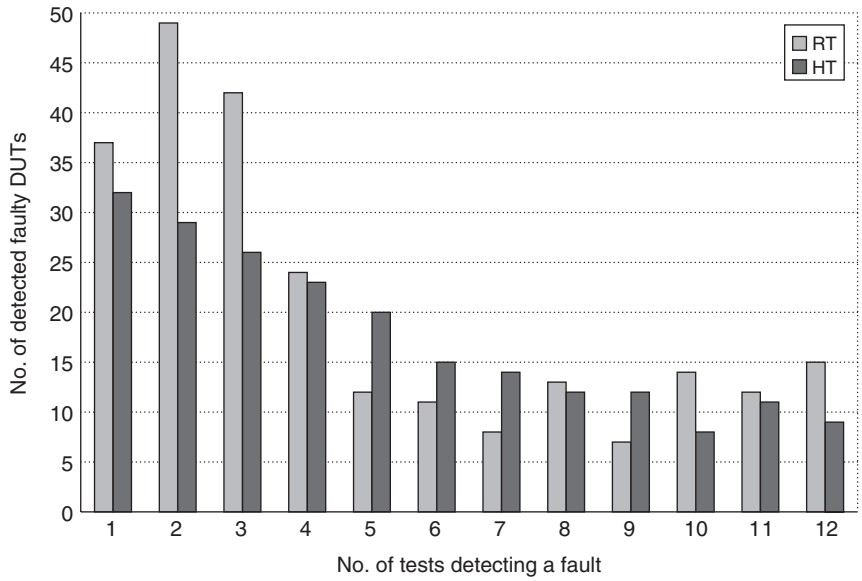

Figure 4. Number of tests detecting faulty DUTs.

each of the stresses used. The bold diagonal entries show the *FC* for each stress; for example, RT tests at *V*– have an *FC* of 678; at *V*+, the *FC* is 617. Of the 617 faulty DUTs that the RT tests detect with stress *V*+, 564 are also detected by *V*– (see the intersection of row *V*– and column *V*+); *S*+ timing has a higher *FC* than *S*– (655 versus 470); and *Ds* is the most effective data background stress. *Ac* is the worst address stress (*FC* = 140); it is almost completely covered by *Ax*. This suggests that random-like addressing was ineffective!

The HT diagonal entries show the *FC* for each stress; for example, *V*– has an *FC* = 414, while for *V*+, the *FC* = 424. Of the 424 faulty DUTs that *V*+ detects, 363 are also detected by *V*– (see row *V*+, column *V*–); *S*– timing has a higher *FC* than *S*+ (452 versus 427); and *Dr* is the most effective data background stress. *Ac* is the worst address stress (*FC* = 51); it is completely covered by both *Ax* and *Ay*. This again suggests that random-like addressing was ineffective!

Single and pair faults

The *x*-axis of Figure 4 shows the number of tests that detected a faulty DUT. Although some faults were detectable by several tests, we focused on the less easily detectable ones. Thus, the plot only covers those faulty DUTs detectable by 12 or fewer tests, out of a total 857 tests.

The *y*-axis shows the number of faulty DUTs detected by the given number of tests. For example, under RT testing, 37 DUTs are only detectable by a single test; and for HT, 32 DUTs fall into this category, which we call *single faults*. Two tests detected 49 DUTs for RT and 29 for HT, and so on. The DUTs detected by only a single test are interesting to analyze, because obtaining an *FC* of 100% will require the use of these tests.

Tables 2 and 3 show the tests that detect the single faults, together with the SC for which that test detected the faulty DUT. Count represents the number of single faults detected by the corresponding test. For RT (Table 2), the 37 single faults require 20 tests to detect; these

Table 2. Tests detecting single faults at RT.

No.	Base test	SC	Count
1	Contact	*AxDsS–V–*	1
2	InpLkH	*AxDsS–V–*	2
7	ICC2	*AxDsS–V–*	1
12	Scan	*AxDrS+V–*	1
20	PMOVI-Re	*AyDrS–V+*	1
21	MarchG	*AyDhS–V–*	2
27	MarchY	*AyDsS+V–*	16
27	MarchY	*AyDhS–V–*	1
28	XMOVI	*AxDsS–V–*	1
28	XMOVI	*AxDsS–V+*	1
28	XMOVI	*AxDcS–V+*	1
34	WalkRow	*AxDcS+V+*	1
35	SlidDiag	*AxDcS+V+*	1
38	HammerW	*AxDrS–V+*	1
39	ScanL	*AxDhS+V–*	1
39	ScanL	*AxDrS+V–*	1
40	MarchC-L	*AxDhS+V–*	1
40	MarchC-L	*AxDrS+V–*	1
40	MarchC-L	*AxDrS+V+*	1
40	MarchC-L	*AxDcS+V+*	1
	Totals	20	37

Table 3. Tests detecting single faults at HT.

No.	Base test	SC	Count
1	Contact	*AxDsS–V–*	2
2	InpLkH	*AxDsS–V–*	1
3	InpLkL	*AxDsS–V–*	1
7	ICC2	*AxDsS–V–*	5
20	PMOVI-Re	*AyDsS+V+*	3
20	PMOVI-Re	*AyDhS+V+*	1
20	PMOVI-Re	*AyDrS+V–*	5
20	PMOVI-Re	*AyDrS+V+*	2
22	MarchU	*AyDhS+V–*	1
27	MarchY	*AyDsS+V–*	6
28	XMOVI	*AxDcS–V+*	1
29	YMOVI	*AyDhS–V+*	1
29	YMOVI	*AyDrS–V+*	3
	Totals	13	32

Table 4. Tests detecting paired faults at RT.

No.	Base test	SC	Count
1	Contact*	*AxDsS–V–*	11
2	InpLkH*	*AxDsS–V–*	12
4	OuptLkH	*AxDsS–V–*	1
9	DataRet	*AxDsS+V–*	1
17	MarchC-	*AyDhS–V–*	1
23	MarchUD	*AyDsS–V+*	1
23	MarchUD	*AyDhS–V–*	1
23	MarchUD	*AyDhS–V+*	1
25	MarchLR	*AyDhS–V–*	1
27	MarchY	*AyDsS–V+*	4
27	MarchY*	*AyDsS+V–*	5
27	MarchY*	*AyDhS–V–*	1
28	XMOVI	*AxDsS–V+*	2
28	XMOVI	*AxDsS+V+*	2
28	XMOVI	*AxDhS–V+*	4
28	XMOVI	*AxDhS+V+*	4
28	XMOVI	*AxDcS–V+*	1
28	XMOVI	*AxDcS+V+*	1
29	YMOVI	*AyDsS–V–*	1
29	YMOVI	*AyDsS+V–*	1
29	YMOVI	*AyDrS–V–*	1
29	YMOVI	*AyDrS+V–*	1
30	Butterfly	*AxDsS–V–*	1
30	Butterfly	*AxDsS+V–*	1
32	GalRow	*AxDcS+V+*	12
34	WalkRow*	*AxDcS+V+*	12
35	SlidDiag*	*AxDcS+V+*	1
39	ScanL*	*AxDhS+V–*	1
39	ScanL*	*AxDrS+V–*	1
39	ScanL	*AxDrS+V+*	1
40	MarchC-L	*AxDsS+V–*	3
40	MarchC-L*	*AxDhS+V–*	1
40	MarchC-L	*AxDhS+V+*	1
40	MarchC-L*	*AxDrS+V–*	3
40	MarchC-L*	*AxDrS+V+*	1
40	MarchC-L	*AxDcS+V–*	1
	Totals	36	98

*These tests detected both single and pair faults.

tests have a total test time of 1,270 s. For HT (Table 3), the 32 single faults require 13 tests for detection; their total test time was 55 s.

The significant difference in test time indicates the effectiveness of HT testing for DRAMs, because HT stress maximizes the leakage phenomena. For RT testing, MarchY is the only march test in the table; this indicates that the march tests cover similar faults. For HT testing, the MOVI tests are dominant, especially PMOVI-Re, showing the importance of the extra read operation.

Tables 4 and 5 show the tests that detected pair faults. Because two tests now detect each fault, the

Table 5. Tests detecting paired faults at HT.

No.	Base test	SC	Count
1	Contact*	*AxDsS–V–*	8
2	InpLkH*	*AxDsS–V–*	8
17	MarchC-	*AxDrS+V+*	1
17	MarchC-	*AxDcS–V+*	1
17	MarchC-	*AyDrS–V+*	1
20	PMOVI-Re	*AyDsS–V+*	6
20	PMOVI-Re*	*AyDsS+V+*	7
20	PMOVI-Re	*AyDrS+V+*	1
20	PMOVI-Re	*AyDcS–V+*	2
20	PMOVI-Re	*AyDcS+V+*	2
25	MarchLR	*AyDrS–V+*	1
27	MarchY	*AyDsS–V+*	1
27	MarchY*	*AyDsS+V–*	2
28	XMOVI	*AxDrS–V+*	2
28	XMOVI	*AxDrS+V+*	1
28	XMOVI*	*AxDcS–V+*	3
28	XMOVI	*AxDcS+V+*	2
29	YMOVI*	*AyDrS–V+*	2
29	YMOVI	*AyDrS+V+*	2
38	HammerW	*AxDrS–V+*	2
38	HammerW	*AxDrS+V+*	2
40	MarchC-L	*AxDrS+V–*	1
	Totals	22	58

* These tests detected both single and pair faults.

Total = 98 for the 49 RT pair faults, and 58 for the 29 HT pair faults. For RT, detecting the 49 pair faults requires 36 tests and a total test time of 2,104 s. For HT, the 29 pair faults require 22 tests and a total test time of 220 s, again indicating the effectiveness of HT testing.

Of the list of 20 RT tests in Table 2, 11 tests, indicated with an asterisk, also appear in Table 4. Of the list of 13 HT tests in Table 3, six tests also appear in Table 5. Of the 20 RT tests, the contribution of the march tests to the *FC* is very low. (Only MarchY is a pure march test.) The nonlinear tests (XMOVI, YMOVI, GalRow, WalkRow, and SlidDiag) detect a total of 43 pair faults at RT and 12 at HT, and the long tests (those with the -L suffix) detect a total of 13 pair faults at RT and 1 at HT. The SCs for the single and pair faults show a large variation, indicating that, for a high *FC*, the ITS should include many different SCs.

Test time optimization

Figure 5 shows the RT *FC* as a function of the test time for the optimization algorithms RemSin, RemHdt, SingGr, and Greedy. The Greedy algorithm establishes a test set by adding the test with the highest *FC*, until it reaches the targeted *FC*. The SingGr algorithm first selects all tests that detect single faults, and then applies Greedy until it reaches the targeted *FC*. RemSin reduces the ITS by systematically removing the test that detects single faults and has the longest test time; when the targeted *FC* is still higher, it applies the same algorithm for pair faults, triple faults, and so on. Figure 5 clearly shows that RemHdt has the best performance. For *FC* less than 100%, RemHdt first removes those single faults that require the longest test time to detect (the *hardest* to detect, hence the "Hdt" in the algorithm's name), then RemHdt removes the tests for the hardest-to-detect pair faults, and so on. A graph like this is useful in making an economical trade-off between the *FC* and the test time (test cost).

Comparing RT and HT results

A comparison of the RT (25° C) and HT (70° C) tests results in the following observations:

- For RT, *Ax* and *S+* (for HT, *Ay* and *S–*) are the most effective addressing and timing stresses. The most effective data background and voltage stresses differ between RT and HT: *Ds* for RT versus *Dr* for HT; and *V–* for RT versus *V+* for HT.
- Long cycle tests (group 9) are not important in the HT tests; they detect 342 faults at RT, including six of the 37 single faults. At HT, they detect only 46 faults, of which none are single faults.
- Only extra read operations added to the end of march elements (like in PMOVI-Re) are effective, and even more so at HT.
- The differences between the unions and intersections of base tests (see Table 1) are larger for HT tests, which means that the appropriate SC for HT is even more important.
- HT tests are more effective, because a major failure phenomenon for DRAMs is leakage, which increases exponentially with temperature. The test results support this by, first, showing that the detection of single faults at RT requires 20 tests, with a total test time of 1,270 s. HT requires only 13 tests with a total test time of 55s.

  Second, the detection of pair faults at RT requires 36 tests and 2,104 s, versus 22 tests and 220 s for HT, an order-of-magnitude reduction in test time! Third, detecting all 731 RT faults requires 42 tests and a total test time of 1,335 s; the detection of all 475 HT faults requires 30 tests and only 159 s!

Theoretical versus actual results

Table 6 shows the *FC* of some of the base tests, in order of increasing fault detection capabilities, based on theoretical expectations.[4] The values for the columns Union and Intersection come from the columns RT Union and RT Intersection of Table 1. From Table 6, it is possible to see that the industrial results correspond to the theoretically expected results, with two exceptions: MarchY,[4] for unclear reasons, does better; and PMOVI does worse. The column Maximum *FC* shows the SC with the highest *FC* for that particular base test. Here, the industrial results are also in line with theoretical expectations, except again for MarchY and PMOVI. Note that the maximum *FC* is almost consistently obtained with an SC of *AyDs* (Fast Y addressing and a solid data background). The Minimum *FC* column shows the SC with the lowest *FC* for that particular base test; *FC* is almost identical for all base tests and occurs mostly with an SC of *AcDcS–V+*.

General observations

The most effective RT base tests are MarchY and those tests that use the long timing, such as Scan-L and MarchC-L. Long timing is likely important to detecting cell leakage. For HT, the most effective tests are XMOVI, PMOVI-Re, and YMOVI. These tests all belong to the MOVI class, indicating delay faults in the X and Y address decoder paths.

The *FC* for a given base test depends to a large extent on the SC; hence, the determination of the most effective SC is very important. This is especially true for the tests that detect single and pair faults.

The *Ac* (address complement) stress consistently scores worst; this indicates that faults are most likely between neighbor cells in the same row or column.

From the *FC* at the group level (Table 2), you can conclude that many of the groups (such as the groups with the MOVI and the -L tests) cover faults of a specific class.

HT tests are more efficient for detecting single and pair faults. This category requires significantly fewer tests, which also results in a dramatic reduction in test time.

The SCs with the highest *FC* for RT tests are *AyDsS–V+* or *AyDsS+V–*. This finding applies to all tests, as Table 6 shows. For the HT tests, *AyDrS–V+* produces the highest *FC* for all tests. The lowest *FC* for RT tests is with *AcDcS–V+*; for HT tests, it is *AcDhS+V–*. This result indicates that *Ac* and *Dh* (the checkerboard data background) consistently produce the lowest *FC*.

Variations on existing base tests that add read operations to the end of the march elements increase the *FC*.

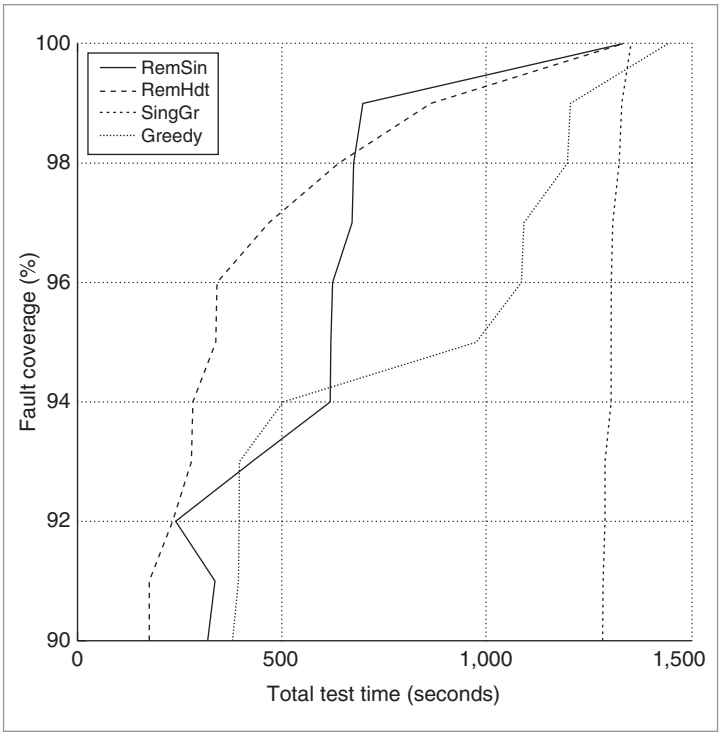

Figure 5. Optimizations of the RT test set.

Table 6. Fault coverage for selected base tests at RT.

	FC		*FC*		*FC*	
Base test	**Union**	**Intersection**	**maximum**	**SC**	**minimum**	**SC**
Scan	144	30	67	*AcDsS–V–*	38	*AcDhS–V+*
MATS+	211	39	140	*AyDsS–V+*	44	*AyDhS+V–*
MATS++	215	39	137	*AyDsS–V+*	43	*AcDcS–V+*
MarchY	267	40	181	*AyDsS+V–*	45	*AcDcS–V+*
MarchC-	234	39	155	*AyDsS–V+*	45	*AcDhS–V+*
MarchU	234	42	154	*AyDsS–V+*	48	*AcDhS–V+*
PMOVI	201	40	138	*AyDsS–V+*	46	*AcDcS–V+*
MarchA	222	39	143	*AyDsS+V–*	44	*AcDcS–V+*
MarchB	232	40	144	*AyDsS+V–*	44	*AcDcS–V+*
MarchLR	235	42	155	*AyDsS–V+*	48	*AcDcS–V+*
MarchLA	241	41	157	*AyDsS–V+*	47	*AcDcS–V+*

IT IS IMPORTANT to note that these results are for a particular type of DRAM chip. Other chips will likely have different results because of differences in design and fabrication, so would require further research. Another avenue of investigation could be in reducing the test time to an economically acceptable level (of about 120 s); this would mean eliminating the nonlinear tests. This, in turn, would require a better understanding of the detected faults to design linear tests optimized for the specific faults.

Tests with the most promising *FC,* based on theory, also tend to have the highest *FC* in practice. However, there is still much that remains unexplained. For example, no theoretical bases exist to model stresses and predict the *FC* of a given SC; this also remains a research topic. ■

Acknowledgments

This article would not have been possible without the contribution of J. de Neef. As part of his MSEE thesis work, de Neef implemented the tests on the tester, applied the tests, and collected the results. He was the coauthor on a previous publication.[7]

■ References

1. H.-D. Oberle, M. Maue, and P. Muhmenthaler, "Enhanced Fault Modeling for DRAM Test and Analysis," *Proc. 9th IEEE VLSI Test Symp.* (VTS 91), IEEE CS Press, 1991, pp. 149-154.
2. H.-D. Oberle and P. Muhmenthaler, "Test Pattern Development and Evaluation for DRAMs with Fault Simulator RAMSIM," *Proc. Int'l Test Conf.* (ITC 91), IEEE CS Press, 1991, pp. 548-555.
3. A.J. van de Goor and G.N. Gaydadjiev, "March LR: A Memory Test for Realistic Linked Faults," *Proc. 14th IEEE VLSI Test Symp.* (VTS 96), IEEE CS Press, 1996, pp. 272-280.
4. A.J. van de Goor, *Testing Semiconductor Memories, Theory and Practice*, ComTex Publishing, 1998.
5. H. Goto, S. Nakamura, and K. Iwasaki, "Experimental Fault Analysis of 1Mb SRAM Chips," *Proc. 15th IEEE VLSI Test Symp.* (VTS 97), IEEE CS Press, 1997, pp. 31-36.
6. I. Schanstra and A.J. van de Goor, "An Industrial Evaluation of Stress Combinations for March Tests applied to SRAMs," *Proc. IEEE Int'l Test Conf.* (ITC 99), IEEE CS Press, 1999, pp. 983-992.
7. A.J. van de Goor and J. de Neef, "Industrial Evaluation of DRAM Tests," *Proc. Design Automation and Test in Europe* (DATE 99), IEEE CS Press, pp. 623-630.
8. A.J. van de Goor and A. Paalvast, "Industrial Evaluation of DRAM SIMM Tests," *Proc. IEEE Int'l Test Conf.* (ITC 00), IEEE CS Press, 1999, pp. 426-435.
9. A.J. van de Goor and I. Schanstra, "Address and Data Scrambling: Causes and Impact on Memory Tests," *Proc. 1st IEEE Int'l Workshop on Electronic Design, Test and Application* (DELTA 02), IEEE Press, 2002, pp. 128-136.
10. M. Klaus and A.J. van de Goor, "Tests for Resistive and Capacitive Defects in Address Decoders," *Proc. 10th Asian Test Symp.* (ATS 01), IEEE CS Press, 2001, pp. 31-36.

Ad J. van de Goor is a professor of computer architecture at the Delft University of Technology in The Netherlands, a position from which he partly retired in 2000. His research interests include computer architecture and testing. Van de Goor has an MSEE from the Technical University of Delft, and an MSEE and a PhD in electrical engineering from Carnegie Mellon University, Pittsburgh, Penn. He has written two textbooks and more than 175 papers, and is a Fellow of the IEEE.

■ Direct questions and comments to Ad J. van de Goor, Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Section Computer Engineering, Mekelweg 4, 2628 CD Delft, The Netherlands; a.j.vdgoor@ewi.tudelft.nl.