SRAM Delay Fault Modeling and Test Algorithm Development

Rei-Fu Huang, Yan-Ting Lai, Yung-Fa Chou, and Cheng-Wen Wu
Laboratory for Reliable Computing (LaRC)
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 30013

Abstract — With the advent of deep-submicron VLSI technologies, the working speed of SRAM circuits has grown to a level that at-speed testing of SRAM has become an important issue. In this paper, we present delay fault models for SRAM, i.e., the faults that affect the access time of the SRAM circuit. We also develop the test algorithm that detects these faults. The proposed SRAM delay-fault test algorithm has a complexity of $3N + 2k$ Read/Write operations, where N is the number of words and k is the word count in a row.

I. Introduction

Memory cores have been widely used in system-on-chip (SOC) designs. An SOC that contains hundreds of SRAM cores is not uncommon today. According to ITRS [1], embedded memories represent a major portion of the area of a typical system-on-chip (SOC), and the percentage is likely to go beyond 90% in less than ten years. In addition, memories usually have higher circuit density than logic blocks, so their testing and diagnosis is becoming more and more important. With the advent of deep-submicron (DSM) VLSI technologies, the working speed of SRAM circuits has grown to a level that at-speed testing of SRAM has been considered an important issue. The advent of DSM technologies also brought forth the requirement of new fault models, especially the delay fault models [2].

Failure analysis is an important step for establishing good fault models for subsequent diagnostics and debugging of the semiconductor memory products. There have been many research works on RAM fault models in the past (see [3] for a summary of conventional RAM fault models). In [4–6], the authors report the results of defect analysis on DRAM and SRAM circuits, and propose some new fault models. They also show the analysis of the faulty memory behavior under different temperatures. In [7, 8], the authors simulate the SRAM circuits using different test algorithms based on the notion of weighted critical area (WCA). They also show the defect coverage and fault coverage of the test algorithms. By simulating the faulty circuit with different test algorithms, some researchers [9, 10] were able to show the correlation between the defect coverage and fault coverage for each test algorithm. There are also some research results showing defects that cause circuit delay. In [11], the authors provide a screening technique for delay defects caused by high resistance interconnect. In [12], the authors present resistive open and bridge fault models incorporating both functional and delay effects of spot defects. In another work, the maximal aggressor fault model is presented as a high level representation of physical defects that lead to crosstalk errors on SOC interconnects [13]. However, all these previous studies on delay effect of defects are for general DSM circuits.

In this paper, we focus on the delay effect of defects in SRAM circuits. The analysis of the delay effect is based on the SRAM defects that are modeled as word-line delay fault and bit-line delay fault, according to the fault behavior caused by the defects. We also propose a test algorithm for the delay faults. The proposed SRAM delay-fault test algorithm has a complexity of $3N + 2k$ Read/Write operations, where N is the number of words and k is the word count in a row. Experimental results show that the test algorithm can effectively detect delay faults caused by defects on SRAM circuits.

II. Delay Fault Models

The delay in an SRAM circuit can be caused by different defects, such as (resistive) opens, shorts, and various process variation defects. In this paper, we do not address specific defects. We consider only the delay behavior caused by the defects. We model and classify the fault behavior as *word-line delay fault* (WDF) and *bit-line delay fault* (BDF). The details of the two fault models are given below.

A. Word-Line Delay Fault (WDF)

The SRAM circuit is composed of a cell array and the peripheral circuit. The latter includes the address decoder, pre-charge circuit, equalization circuit, sense amplifier, etc. Each of these peripheral components can be implemented in multiple ways. The word-line delay described here is assumed to be caused by defective address decoder and/or word-lines. The address decoder is a two-level logic circuit that has several implementation styles, including multi-level decoding for large memories and dynamic circuit implementation for reducing the decoding time and circuit complexity. Figure 1 shows a simple block diagram of a RAM, including the address decoder design that has both the X-decoder and Y-decoder for row and column selection, respectively. In the figure, we show a WDF caused by a defect in the X-decoder. If the defect, e.g., results in an extra impedance in the word-line or reduces the driving power of the word-line, the access time of the memory cells in this word-line increases.

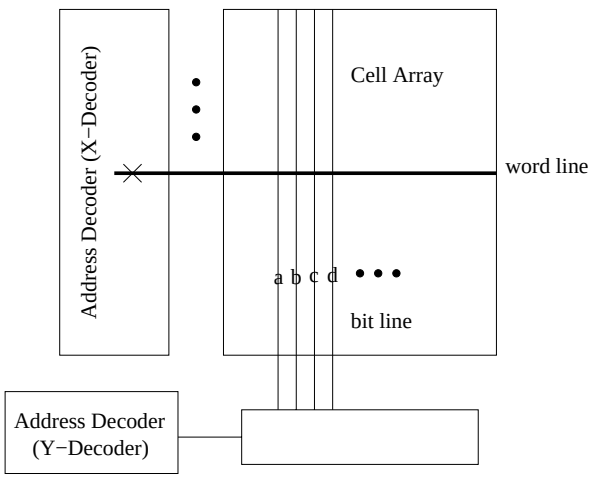

Fig. 1. A WDF example.

The fault effect (sensitivity of access timing) depends on the actual design of the address decoder. If the address decoder is implemented by a dynamic circuit, the delay is more likely to be affected. The conventional address decoder faults [3], are static faults, which may not cover defects that cause only delay problems, i.e., when the memory fails only at high speed under certain test patterns and address/access sequences. The behavior of the WDF is that the read-out data is the previous one instead of the one expected from the memory word. Because the sense amplifier is a latch-based design, it catches the previous value if the output data has a delay. Consider Fig. 1 again, where the faulty word-line is caused by a defect in the X-decoder. If a known pattern has been stored in the word and we read its neighboring cells sequentially (e.g., a, b, c, d), then the output data of the bit-lines b, c, and d will be those stored in a, b, and c.

B. Bit-Line Delay Fault (BDF)

The bit-lines are usually the longest (most loaded) lines in the memory cell array. In common SRAM circuits we see 128 or more cells driven by each bit-line. When the defect causes an extra capacitor (in parallel) or resistor (in series) on the bit-line [12–14], the delay fault may occur that affects the memory access time.

Figure 2 shows the timing diagram of the Write operation, which can be divided into three phases—the write phase, equalization phase, and pre-charge phase. The Read operation can be expressed in a similar way. An extra bit-line load capacitor will increase the equalization time, which in turn affects the pre-charge performance. If the fault occurs, then the bit-line and $\overline{\text{bit-line}}$ cannot be pre-charged to V_{DD}, and the subsequent Read operation may catch a faulty data value. Both the Write and Read operations may exhibit the fault behavior. When the memory executes the Write operation, the voltage difference of the bit-line and $\overline{\text{bit-line}}$ is larger than that of the Read operation, because the voltage forking of the Write operation is from the power supply instead of the inverter pair in the memory cell (as in the Read operation). The equalization time of the Read operation is thus less than that of the Write operation, so the fault can be detected more easily using the Write operation.

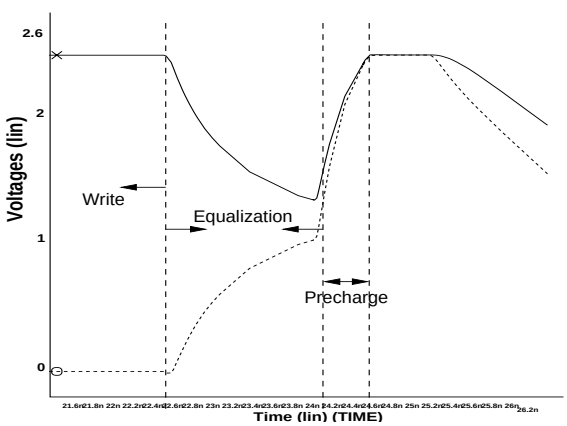

Fig. 2. Timing diagram of the Write operation.

The extra resistance in the metal lines or other parts of the circuit caused by defects has been discussed in previous works [12, 14]. In what follows we discuss some aspects of the excessive capacitance on the bit-line.

- Process control to fix the capacitance of the metal line is hard, and a 10% error is easily seen. We take a bit-line with 512 memory cells for example. The capacitance of the bit-line includes that of the metal line and that of the memory cells (seen from the bit-line). A typical value of either the capacitance of the metal line or that of the cells is about 1 pF, so the capacitance of the bit-line is about 2 pF. A process defect can easily increase the capacitance by 10%, i.e., 200 fF, which leads to a delay fault that affects the access time.

- The large capacitance on the bit-line may slow down the discharge process, which reduces the voltage difference between bit-line and bit-line.

- The memory cell has a leakage current, and it increases as we go deeper into the DSM technology. If the leakage current is too large, the behavior is like a large capacitor on bit-line.

- If the layout is not done properly, there can be an observable parasitic capacitor between two neighboring lines, e.g., the power line (V_{DD} or GND) is usually laid in parallel with bit-line or bit-line.

- For the pre-charge circuit the dynamic load has a higher performance than the static load, so the dynamic pre-charge circuit is more commonly seen in modern SRAMs. However, an SRAM using the dynamic pre-charge circuit suffers from slow access when leakage occurs. When the word-line is high, the cell current discharges the bit-line. If the pre-charge circuit leaks, the discharge process slows down or can never be done. The behavior is similar to a large capacitance on the bit-line.

- The cell current is defined as the current from the bit-line to the memory cell. If the cell current is small, the discharge process of the bit-line is slow. The behavior is similar to that caused by a large capacitor on bit-line.

The timing specifications of a memory core usually consist of four types—clock, output time, setup time, and hold time. The clock specification includes the clock period, rise time, and fall time. The output time specifies the duration from data request to data output. The setup time and hold time are for address setup and hold times, input data setup and hold times, write enable setup and hold times, etc. These are important timing specifications for the memory, so during production they should be tested. The delay fault models proposed in this paper can be used for detecting faults leading to excessive access time (i.e., the output time).

III. Test Algorithm Development

A. Testing WDF

Although we had stressed the Read operation, the WDF may also affect the Write operation—if there is a WDF, the cell on the faulty word-line may not be written properly. However, during the Write cycle, the WDF causes a different scenario from Read cycle. In the Write cycle, bit-line or bit-line usually discharges the cell with a more powerful current sink than the Read operation. We assume that the Write operation is completed correctly, though the voltage level of the word-line may be lower than expected. In such case, it is sufficient to test this fault by $(wa); (ra, wb)$, where $(w\,a)$ means writing the data word a to all cells in certain address sequence, and b is the complement of a. The $(ra\ ,w\ b)$ element is to go through the entire address space by a certain sequence, where at each address we read the expected word a and then write back its complement immediately before proceeding to the next address. When we perform ra at the next address and there is a WDF, the sense amplifier catches the previous value b instead of a.

Figure 3 shows an example for this test, using a row of four cells and assuming bit-oriented operations. After we perform $(w0)$, all the cells are reset to 0. We then execute $(r0, w1)$ from left to right. The first cell returns a correct output value, but the outputs of other cells in the same row are incorrect, because the state of the sense amplifier remains at 1 if the WDF occurs.

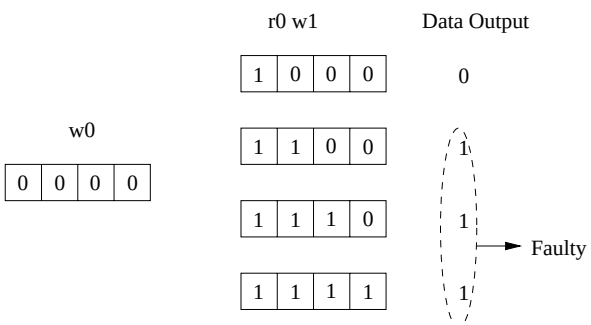

Fig. 3. Testing WDF.

B. Testing BDF

The BDF is caused by a defect leading to an excessive capacitance of bit-line or bit-line, increasing the access time of the memory. The fault behavior of BDF is that the equalization time is too long to have the next operation executed properly, so the subsequent reading can return

an incorrect value. The cell with the slowest equalization process is the farthest cell from the bit-line equalizer, and the cell that reads the next value the fastest is the nearest cell from the sense amplifier. Figure 4 shows the placement of the memory peripheral circuits. In Fig. 4(a), the equalizer and sense-amplifier are at opposite ends of the bit-line. In this case, Cell0 is the cell with the slowest equalization process, and it also reads the next value the fastest. By this observation, we can test the BDF using a three-step procedure: 1) write a to Cell0; 2) write $\bar{a}$ to Cell1 to complement the bit-line and $\overline{\text{bit-line}}$; and 3) read a from Cell0. The slow equalization and fast reading under the BDF will result in a failed Read operation in the third step, so the fault can be detected. Figure 4(b) shows another bit-line design, where the equalizer and sense-amplifier are at the same end of the bit-line. In this case, Cell2 is the cell with the slowest equalization and reading process, while Cell3 is the one with the fastest equalization and reading process. By this similar observation, we can test the BDF using another three-step procedure: 1) write a to Cell3; 2) write $\bar{a}$ to Cell2; and 3) read a from Cell3. In summary, we can test the BDF (for either case) by developing a test algorithm that contains $(wa, *w\bar{a}, ra)$, where $*w$ means writing to another address in the same column that has the slowest equalization process. Note that this test algorithm requires changing the background when accessing the same bit-line. If we include both procedures, then the test length is $3 \times 2 \times k = 6k$, where k is the word count in a row.

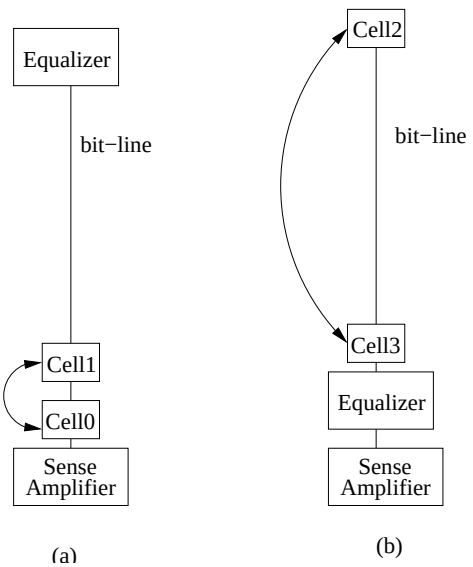

Fig. 4. Placement of peripheral circuits: (a) equalizer and sense-amplifier at opposite ends; and (b) equalizer and sense-amplifier at the same end.

C. Testing Both WDF and BDF

Using the March notation [3, 15], the WDF test algorithm can be written as $\{\Updownarrow (wa); \Updownarrow (ra, w\bar{a})\}$. Assuming that the scramble data of the memory is known, the BDF test algorithm can be written as $(w\bar{a}, *wa, r\bar{a})$. We can combine these two tests into the final test algorithm for both WDF and BDF:

$$\Updownarrow (wa); \Updownarrow (ra, w\bar{a}, *wa, r\bar{a})$$

Note that we may need to repeat the test with a complemented data background if the BDF is pattern dependent. Note also that the last two operations is the test algorithm are applied only to certain rows, as discussed above. The complexity of the proposed SRAM delay fault test algorithm thus is only $3N + 2k$ Read/Write operations, where N is the number of words and k is the word count in a row. The test algorithm also can be embedded in a test for detecting conventional RAM faults (see, e.g., [3, 15]).

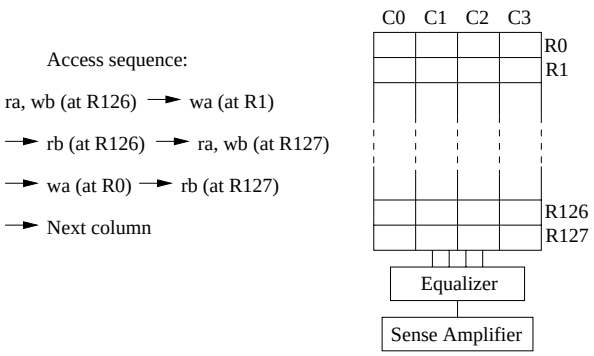

Fig. 5. Experimental SRAM circuit.

IV. Experimental Result

We have simulated this test algorithm for a real SRAM circuit as shown in Fig. 5, where the equalizers and sense amplifiers are all at the lower end of the memory cell array. We have added proper loading on the bit-line, C1, to simulate the BDF. The address jumps for detecting the BDF are also shown in Fig. 5. When the bit-line loading is about 400 fF, the fault behavior is as shown in Fig. 6. If the loading on the bit-line increases, the BDF detection probability also increases, i.e., a loading larger than 400 fF may allow us to detect the fault using more address jumps in addition to those shown in Fig. 5.

We now evaluate the fault coverage of two test algorithms, i.e., the March C− test algorithm [3] and our integrated test algorithm for BDF and WDF. The proposed

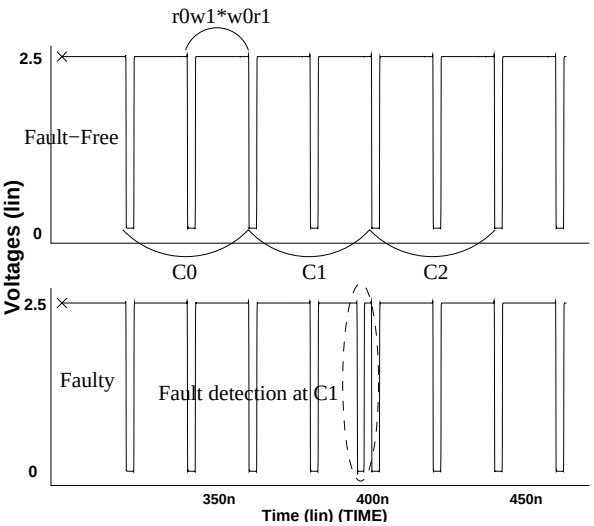

Fig. 6. Fault behavior simulation result.

integrated test algorithm is:

$$\Updownarrow(w0), \Uparrow(r0, w1), \Uparrow(r1, w0), \Downarrow(r0, w1, *w0, r1),$$
$$\Downarrow(r1, w0, *w1, r0), \Updownarrow(r0)$$

The fault coverage is calculated by our fault simulator—RAMSES [15]. Table I shows the fault coverage figures for the target fautls—the stuck-at fault (SAF), transition fault (TF), address decoder fault (AF), and coupling fault (CF). We can see that March C− does not detect BDF, but our integerated test can. The complexity of the proposed integrated test is $12N+4k$, which is $4k$ higher than March C−.

TABLE I
FAULT COVERAGE COMPARISON.

	SAF	TF	AF	CF	BDF	WDF
March C-	100	100	100	100	0	100
Integrated	100	100	100	100	100	100

V. CONCLUSION

We have proposed the BDF and WDF models for SRAM, which will be more and more important when we go deeper into the DSM technology. We also have developed a test algorithm for detecting these faults. The complexity of the proposed SRAM delay fault test algorithm is only $3N + 2k$ Read/Write operations, where N is the number of words and k is the word count in a row. The test algorithm can easily be embedded in a test for detecting conventional RAM faults.

REFERENCES

[1] Semiconductor Industry Association, "International technology roadmap for semiconductors (ITRS), 2001 edition", Dec. 2001.

[2] A. Krstic and K.-T. Cheng, *Delay Fault Testing for VLSI Circuits*, Kluwer Academic Publishers, Boston, 1999.

[3] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*, ComTex Publishing, Gouda, The Netherlands, 1998.

[4] S. Hamdioui and A. J. van de Goor, "An experimental analysis of spot defects in SRAMs: realistic fault models and tests", in *Proc. Ninth IEEE Asian Test Symp. (ATS)*, Taipei, Dec. 2000, pp. 131–138.

[5] Z. Al-Ars, A. J. van de Goor, J. Braun, and D. Richter, "Simulation based analysis of temperature effect on the faulty behavior of embedded DRAMs", in *Proc. Int. Test Conf. (ITC)*, Baltmore, Oct. 2001, pp. 783–792.

[6] Z. Al-Ars and A. J. van de Goor, "Approximating infinite dynamic behavior for DRAMs cell defects", in *Proc. IEEE VLSI Test Symp. (VTS)*, Monterey, California, Apr. 2002, pp. 401–406.

[7] A. Jee, J. E. Colburn, V. S. Irrinki, and M. Puri, "Optimizing memory tests by analyzing defect coverage", in *Proc. IEEE Int. Workshop on Memory Technology, Design and Testing (MTDT)*, San Jose, Aug. 2000, pp. 20–25.

[8] A. Jee, "Defect-oriented analysis of memory BIST tests", in *Proc. IEEE Int. Workshop on Memory Technology, Design and Testing (MTDT)*, Isle of Bendor, France, July 2002, pp. 7–11.

[9] V.-K. Kim and T. Chen, "Assessing SRAM test coverage for sub-micron CMOS technologies", in *Proc. IEEE VLSI Test Symp. (VTS)*, Monterey, Apr. 1997, pp. 24–30.

[10] V.-K. Kim and T. Chen, "On comparing functional fault coverage and defect coverage for memory testing", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 11, pp. 1676–1683, Nov. 1999.

[11] C.-W. Tseng, E. J. McCluskey, X. Shao, and D. M. Wu, "Cold delay defect screening", in *Proc. IEEE VLSI Test Symp. (VTS)*, Montreal, Apr. 2000, pp. 183–188.

[12] Z. Li, X. Lu, W. Qiu, W. Shi, and D. M. H. Walker, "A circuit level fault model for resistive opens and bridges", in *Proc. IEEE VLSI Test Symp. (VTS)*, Napa Valley, Apr. 2003, pp. 379–384.

[13] M. Cuviello, S. Dey, X. Bai, and Y. Zhao, "Fault modeling and simulation for crosstalk in system-on-chip interconnects", in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, 1999, pp. 297–303.

[14] H. Pilo, R. D. Adams, R. E. Busch, E. A. Nelson, and G. E. Rudgers, "Bitline contacts in high density SRAMs: Design for testability and stressability", in *Proc. Int. Test Conf. (ITC)*, Baltmore, Oct. 2001, pp. 776–782.

[15] C.-F. Wu, C.-T. Huang, K.-L. Cheng, and C.-W. Wu, "Fault simulation and test algorithm generation for random access memories", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 4, pp. 480–490, Apr. 2002.