Consequences of RAM bitline twisting for test coverage

Ivo Schanstra
Infineon Technologies AG
Balanstrasse 73, D-81541 Munich, Germany
Ivo.Schanstra@Infineon.com

Ad. J. van de Goor
Delft University of Technology
Department of Information Technology and Systems
Section Computer Engineering
Mekelweg 4, 2628 CD Delft, The Netherlands
A.J.vandeGoor@ITS.TUDelft.nl

Abstract: *In order to reduce coupling effects between bitlines in static or dynamic RAMs, bitline twisting can be used in the design. For testing, however, this has consequences for the to-be-used data backgrounds. A generic twisting scheme is introduced and the involved fault models are identified.*

1 Formal twisting notation

Twisting can be defined as the local reordering of parallel-running interconnect lines. It can be used for the bitline and/or wordline schemes of memories, or for busses in general [1-7].

Figure 1 shows a generic twisting scheme for a large number of interconnect lines that run from left to right. It is called a *two-dimensional* twisting scheme, because all lines run in a plane[1].

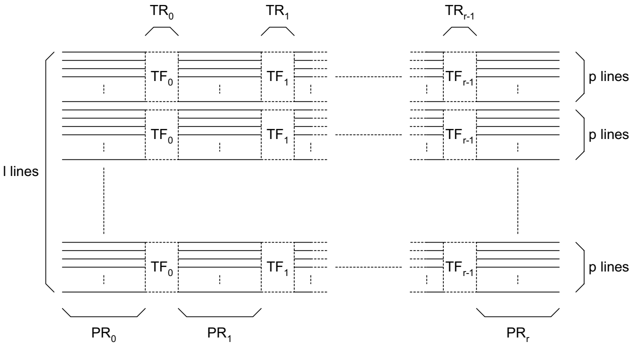

Figure 1: Basic two-dimensional twisting scheme

As can be seen from this figure, the reordering of the interconnect lines only takes place in so-called *Twisting Regions* (TRs); outside the TRs, in the so-called *Parallel Regions* (PRs), the interconnect lines simply run in parallel. There can be an arbitrary number of TRs in a twisting scheme; the number of TRs is indicated by r. The r TRs are indicated by TR$_0$, TR$_1$, ..., TR$_{r-1}$. Given r TRs, there are $r+1$ PRs; indicated by PR$_0$, PR$_1$, ..., PR$_r$.

For a large number of interconnect lines l, as shown in Figure 1, a twisting scheme consisting of $p \ll l$ lines is simply repeated. The number p is called the *period* of the twisting scheme and is defined as the number of lines in the smallest repeating pattern.

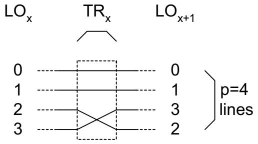

Figure 2: Detail of a twisting region

Exactly how each group of p lines is reordered in each of the r TRs is defined by the r *twisting functions* (TFs) TF$_0$, TF$_1$, ..., TF$_{r-1}$. Figure 2 shows an example section of a TR$_x$, $0 \leq x \leq r-1$ and $p = 4$. Also indicated in this figure is the *Line Order* (LO). The lines in PR$_0$ are numbered from top to bottom starting at 0; LO$_x$ is simply the order of these lines in PR$_x$, taking all twisting in TR$_0$ $\ldots$TR$_{x-1}$ into account. In the example of Figure 2, LO$_x$ is assumed identical to LO$_0$.

The LOs of Figure 2 can be written in vector form as follows:

$$\mathrm{LO}_x = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix} \qquad \text{and} \qquad \mathrm{LO}_{x+1} = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 2 \end{pmatrix}$$

The TF$_x$ can now be written in matrix form, such that $\mathrm{LO}_{x+1} = \mathrm{TF}_x \mathrm{LO}_x$:

$$\mathrm{TF}_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathrm{LO}_{x+1} = \mathrm{TF}_x \mathrm{LO}_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 2 \end{pmatrix}$$

[1] except in the twisting regions, where the third dimension must be used to implement the twisting

2 Relevant fault models

Bitline twisting influences the following aspects of memory operation:

- SRAM cell contents: an SRAM cell has two nodes storing opposite values; twisting influences which node stores the externally written value and which the opposite. This way, for example, twisting can transform a stuck-at-zero fault into a stuck-at-one fault or vice versa;

- DRAM cell contents: a DRAM cell only has one single node, which for some cells can be connected to a BL, and for the other cells to a $\overline{BL}$ of a sense amplifier. In some DRAM architectures, for a given external logic value, the value stored in the cells connected to the $\overline{BL}$ of a sense amplifier is the opposite of the value stored in cells connected to the BL. In combination with folding, for a cell in a given row and column of the memory cell array, twisting influences the value stored in that cell.

- DRAM cell location: In a DRAM, similar to address scrambling, twisting influences the location (column) in the memory cell array of the cell that stores a bit for a given address and data I/O; hence it also influences which cells (based upon logical address and I/0) are in eachothers neighborhood. For an SRAM, this location is not influenced by any of the twisting schemes shown in this paper; regardless of twisting, all SRAM cells connected to the same bitlines are in the same column of the memory cell array.

As a result of this, if a test is not symmetrical with respect to logic values[2], or if a test takes the (relative) location of cells into account, twisting influences test coverage. For the various fault types this has the following consequences.

A. Single-cell faults

All single-cell fault models are cell location independent, and are symmetric with respect to the '0' and '1' state, and '0' to '1' and '1' to '0' transitions. Common tests for single-cell faults are also symmetric with respect to logic value. Bitline twisting does not have to be taken into account.

B. 2-cell coupling faults

All 2-cell static CF models are symmetric with respect to the '0' and '1' state, and '0' to '1' and '1' to '0' transitions in the a-cell (aggressor cell) or the v-cell (victim cell). Here, tests are not always symmetric with respect to logic value. For example, March C-[3][8] will detect some dynamic faults. The static CFds $<$0w1;0/1/-$>_{a<v}$, for which the a-cell has a lower address than the v-cell, is detected by the single march element ⇑(r0,w1). This test will also detect the dynamic CFds $<$0w1r1;0/1/-$>_{a<v}$. In case of bitline twisting, such that the value of the v-cell becomes inverted, the effective fault primitive becomes $<$0w1;1/0/-$>_{a<v}$. March C- detects this using the two march elements ⇓(r0,w1);⇓(r1,w0), which is fine to detect the static fault, but not to detect the dynamic fault. Here, the fault coverage varies due to bitline twisting if it is not compensated for.

Some 2-cell coupling faults cannot be detected by functional (voltage) testing. Consider a resistive bridge between two nodes of two different SRAM cells. If the resistance value is low enough, a CFst (state coupling fault) will occur that can be detected by voltage testing. If, however, the resistive value is high enough, a weak CFst is present that can only be detected by measuring the leakage current after initializing the memory with a certain contents. Since it is not feasible to do an Iddq measurement for every possible memory content, only those cases are considered where the coupled cells are adjacent. Thus, location and therefore bitline twisting become relevant.

C. k-cell coupling faults

For k-cell coupling faults, tests cannot afford the luxury to assume just any location of the involved cells; otherwise, testtime would explode. Three cases are normally considered: all cells along a bitline, all cells along a wordline, or all cells in a physical neighborhood (NPSF). The last case is influenced by bitline twisting.

References

[1] Michael Redeker, Bruce F. Cockburn, and Duncan G. Elliott. An Investigation into Crosstalk Noise in DRAM Structures. In *Records of IEEE International Workshop on Memory Technology, Design and Testing*, pages 123–129, Isle of Bendor, France, July 2002.

[2] W. Tan, Lee Keat Peng, and Giam Siang Tian. Poly-3 bitline crack. In *Proc. 5th International Symposium on the Physical and Failure Analysis of Integrated Circuits*, pages 206–211, Singapore, December 1995.

[3] H. Hidaka, K. Fujishima, Y. Matsuda, M. Asakura, and T. Yoshihara. Twisted bit-line architectures for multi-megabit DRAMs. *IEEE Journal of Solid-State Circuits*, 24(1):21–27, February 1989.

[4] Dong-Sun Min and Dietrich W. Langer. Twisted bit-line technique for multi-gigabit DRAMs. *Electronic Letters*, 33(16):1380–1382, July 1997.

[5] Y. Oowaki, K. Tsuchida, Y. Watanabe, D. Takashima, M. Ohta, H. Nakano, S. Watanabe, A. Nitayama, F. Horiguchi, K. Ohuchi, and F. Masuoka. A 33-ns 64-Mb DRAM. *IEEE Journal of Solid-State Circuits*, 26(11):1498–1505, November 1991.

[6] Dong-Sun Min and Dietrich W. Langer. Multiple twisted dataline techniques for multigigabit DRAMs. *IEEE Journal of Solid-State Circuits*, 34(6):856–865, June 1999.

[7] A.J. van de Goor and H.I. Schanstra. Address and Data Scrambling: Causes and Impact on Memory Tests. In *Proc. 1st International Workshop on Design, Test and Applications (DELTA2002)*, pages 128–136, Christchurch, New Zealand, January 2002.

[8] A.J. van de Goor. *Testing Semiconductor Memories, Theory and Practice*. ComTex Publishing, Gouda, The Netherlands, 1998, http://ce.et.tudelft.nl/∼vdgoor/.

[2] e.g., if it detects stuck-at-zero then it also detects stuck-at-one, etc.

[3] {⇕(w0); ⇑(r0,w1);⇑(r1,w0);⇓(r0,w1);⇓(r1,w0);⇕(r0)}