ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An UGC Approved Journal)

Website: www.ijareeie.com

Vol. 6, Issue 8, August 2017

Detection of Faults in SRAM Using TOMT Algorithm

D.Srija[1], P.Ramana Reddy[2], KVBV Rayudu[3]

PG Scholar (DSCE), Dept. of ECE, JNTUACEA, Anantapur, Andhrapradesh, India[1]

Professor, Dept. of ECE, JNTUACEA, Anantapur, Andhrapradesh, India[2]

Sc-F, R&QA, RCI-DRDO, Hyderabad, Telangana, India[3]

ABSTRACT: The Transparent Online Memory Test (TOMT) introduced here is developed for online testing of parity or Hamming Code protection. Careful interleaving of a word-oriented and a bit-oriented test facilitates fault coverage and test duration comparable to the widely used March C-algorithm. TOMT actively exercises all bit cells in memory within one test period. Hence it not only detects the soft errors but also the functional faults and prevents fault accumulation.

KEYWORDS: built-in self-test, coupling fault, march test, memory testing, online testing, transparent testing.

I. INTRODUCTION

As CMOS technology has been increasing gradually, shrinking feature size is the key to the semiconductor industry. Smaller and denser physical structures facilitate the integration of more functionality on one chip, while lower power supply voltage helps keep power dissipation within manageable bounds. Both these trends contribute to a drop of the critical charge. As a result, chips become increasingly susceptible to external disturbances. The effect of cosmic rays is often used to exemplify this. Cosmic rays cause single event upsets (SEU) in integrated circuits. These SEU are referred as soft errors, as they can only change the value of a storage cell but do not cause any permanent defects.

With the physical structure and large chip area they occupy, memories are particularly sensitive to SEU and therefore considered the most unreliable parts in system. But at the same time, memories are little easy to protect from SEU by parity or Hamming code. To prevent accumulation of both soft and hard faults we complement the data integrity check performed by parity and Hamming codes by an active test. Such a test must be performed while the system is still providing its service.

Several solutions for online testing have been proposed in literature, but all of these have drawbacks in practical application. There are some approaches invert and re-invert the contents of memory blocks consecutively, employing signatures to check data integrity. The main drawback of these approaches is that faults can only be found after testing the whole block when the final signature is known. Thus, a Transparent Online Memory Test (TOMT) is proposed which performs active fault detection over the whole memory. These properties render TOMT an effective means to prevent accumulation of both soft and hard faults. The ongoing test does not prevent the processor from normal memory usage—it is completely transparent in the temporal domain. TOMT employs a word-level algorithm which acts in concert with embedded split transparent march tests on the bit-level. These transparent march tests do not generate signatures over the memory. Instead we use check bits (parity or Hamming code) on word-level for fault detection avoiding the problem of signature aliasing and sparing the time necessary for signature prediction.

II. LITERATURE SURVEY

In-flight and ground testing of single event upset sensitivity in static RAMs

This paper presents the results from in-flight measurements of single event upsets (SEU) in static random access memories (SRAM) caused by the atmospheric radiation environment at aircraft altitudes. The memory devices

DOI:10.15662/IJAREEIE.2017.0608057

were carried on commercial airlines at high altitude and mainly high latitudes. The SEUs were monitored by a Component Upset Test Equipment (CUTE), designed for this experiment. The in-flight results are compared to ground based testing with neutrons from three different sources.K. Johansson Ericsson Saab Avionics AB, Linkoping, Sweden P. Dyreklev O. Granbom are the authors of this paper.

The single-event-effect behavior of commercial-off-the-shelf memory devices—a decade in low-earth orbit.

This paper presents the results of a 10-year study on radiation effects in commercial-off-the-shelf (COTS) memory devices operating within the on-board data handling systems of five low-Earth orbiting microsatellites, designed and built at the University of Surrey (UoS). The ionising particle environment inside three of these spacecraft has been investigated concurrently using radiation monitoring payloads developed by UoS and the Defense Evaluation and Research Agency (DERA). Through the use of these monitoring instruments, and an allied programmed of ground-based testing of the memory devices, the industry-standard computer models of the radiation environment have been verified, and the memory device behavior characterized with respect to single-event (SEEs) due to galactic cosmic-rays, geomagnetically trapped particles, and solar particles. C.I. Underwood Centre for Satellite Eng. Res., Surrey Univ., Guildford, UK.

Transparent BIST for RAMs.

I present the theoretical aspects of a technique called transparent BIST for RAMs. This technique applies to any RAM test algorithm and transforms it into a transparent one. The interest of the transparent test algorithms is that testing preserves the contents of the RAM. The transparent test algorithm is then used to implement a transparent BIST. This kind of BIST is very suitable for periodic testing of RAMs. The theoretical analysis shows that this transparent BIST technique does not decrease the fault coverage for modeled faults, it behaves better for unmodeled ones and does not increase the aliasing with respect to the initial test algorithm. Furthermore, transparent BIST involves only slightly higher area overhead with respect to standard BIST. Thus, transparent BIST becomes more attractive than standard BIST since it can be used for both fabrication testing and periodic testing. M. Nicolaidis TIMA/INPG, Reliable Integrated Syst. Group, Grenoble, France.

A highly-efficient transparent online memory test.

The transparent online memory test (TOMT) proposed in this paper has been specifically developed for online testing of word-oriented memories with parity or Hamming code protection. Using a rotated address sequence the algorithm passes four times through the whole address space and performs embedded march tests for every word. The careful interleaving of word-oriented and bit-oriented test allows one to attain a fault coverage and a test duration comparable to the widely used March C- algorithm. The proposed memory test detects all stuck-at faults, all transition faults, all address decoder faults (even stuck-open address decoder faults), all single coupling faults (CFs, even write and read disturb CFs) and a reasonable percentage of linked CFs. Nevertheless, the algorithm is suitable for online use and can be implemented in hardware with moderate effort. K. Thaller Vienna Univ. of Technol., Austria.

III. CLASSIFICATION OF FAULTS

A fault is a logic error that is caused by a defect and can be modeled by a fault model. Many defects may exist in a circuit, but only some of these defects result in a fault. There are many possibilities that call trigger faults in a circuit.

Classical fault models are not sufficient to represent all important failure modes in a RAM; Functional Fault models should be employed. Memory Fault models can be classified under the categories shown below.

- Functional Faults
- Data Faults

FUNCTIONAL FAULTS: A functional fault is said to have occurred if the memory fails to provide intended function due to a hardware defect. Considering from a functional model of a proper system behavior, there are different types of fault models can be derived. Those fault models are:

STUCK-AT-FAULTS: A stuck-at fault occurs when the value of a cell or line is always 0 (a stuck-at-0 fault) or always 1 (a stuck-at-1 fault).

TRANSITION FAULTS: A transition fault occurs when a cell fails to transit from 0 to 1 or 1 to 0 in specified time period.

COUPLING FAULTS: Coupling Fault implies deviation from normal behavior of a cell because of coupling with others.
There are three types of coupling faults. They are:

- State Coupling Faults.
- Inversion Coupling Faults.
- Idempotent Coupling Faults.

State Coupling Faults (CF_{st}): Coupled (victim) cell is forced to 0 or 1 if coupling (aggressor) cell is in given state.
Inversion Coupling Faults (CF_{inv}): Transition in coupling cell complements (inverts) coupled cell.
Idempotent Coupling Faults (CF_{id}): Coupled cell is forced to 0 or 1 if coupling cell transits from 0 to 1 or 1 to 0.

ADDRESS DECODER FAULTS: Row and column decoder blocks comprise address decoder. These blocks are composed of logic gates.

From the context of memory testing, four types of faults are considered in address decoder (for both reading and writing).

- No cell is accessed for a certain address.
- No address can access a certain cell.
- With a particular address, multiple cells are accessed simultaneously.
- A particular cell can be accessed with multiple addresses.

DATA RETENTION FAULTS: Cell fails to retain its logic value after some specified time due to, e.g., leakage, resistor opens, or feedback path opens.

DATA FAULTS: Adverse environmental conditions such as cosmic irradiation, electromagnetic interference, power supply fluctuations etc. cause data bits to flip without impairing the function of the memory. The data faults induced by these nonpermanent faults (transient faults) are often referred to as soft errors.

IV. THE BASIC TOMT ALGORITHM

The Transparent Online Memory Test algorithm is proposed to work on memories with even data width or even number of information bits and with parity or Hamming code protection. It consists of two parts:

- A **word-level algorithm** determines the sequence of memory words to be tested. By activating the available memory protection scheme periodically and utilizing the respective error detection capabilities, it is capable of unveiling errors in the stored data.
- Within the word under test, a **bit-level algorithm** determines the sequence of bits to be modified and the values need to be assigned. Rather than stored data integrity check, it tests whether the bit cell actively and brought to both logic states and be read properly. It performs stimulation and check by itself. And also, it triggers faults whose impact is detected by the memory protection scheme used within the word-level algorithm.

WORD-LEVEL ALGORITHM: The word-level algorithm browses through memory in forward and backward address order as shown in Figure1. The algorithm generates the current test address from a counter value. The word under test is fetched from memory, stored in the backup register temporarily and checked for parity or Hamming code violation. Thereafter, the word's bit-level test is initiated at the end of which the whole word including check bits has been inverted. As soon as all words have been tested at the end of Run1, the whole memory contents have been inverted. Run2 uses the same address order as Run1. When Run2 is finished the original state of the memory has been restored. Now, the remaining runs of the test cycle, Run3 and Run4, are done in the same manner but in backward address order. In summary, these four runs cover all permutations of address order (forward/backward) and data background (inverted/non-inverted). This is a prerequisite for the detection of address decoder faults and furthermore yields a significant coverage improvement for inter-word coupling faults.

To fit to the inverted data, the check bits' interpretation has to be altered upon every inversion. Without such an adaption a succeeding read access to the previously bit-level-tested word would generate an error. Parity interpretation switches from even to odd and vice versa for an even number of information bits. Hamming code interpretation changes depending on the number of information bits over which a check bit has been generated. For an even number it toggles as for parity from even to odd and vice versa after every inversion. For an odd number it always stays the same.

BIT-LEVEL ALGORITHM: Due to their efficiency, the well-known march algorithms are used on the bit level. A March algorithm consists of a series of March elements which in turn are built up of read and write accesses, abbreviated by *r* and *w*.

We use the March C- test algorithm as a starting point for our basic TOMT algorithm:
{↕(w0);↑(r0,w1);↑(r1,w0);↓(r0,w1);↓(r1,w0);↨(r0)}

Obviously this test is nontransparent. To make it transparent we skip the initialization step and perform the bit-flipping on top of the existing data contents. This means that instead of writing 0 or 1 to a bit cell, we either use the initial value *x* of a bit prior to a word's march test or its inverse *x'*. Therefore, *wx* and *wx'* indicate write accesses to the current bit setting it to its initial and its inverse value. The read access *rx* and *rx'* expects its initial value and its inverse to be read, respectively.

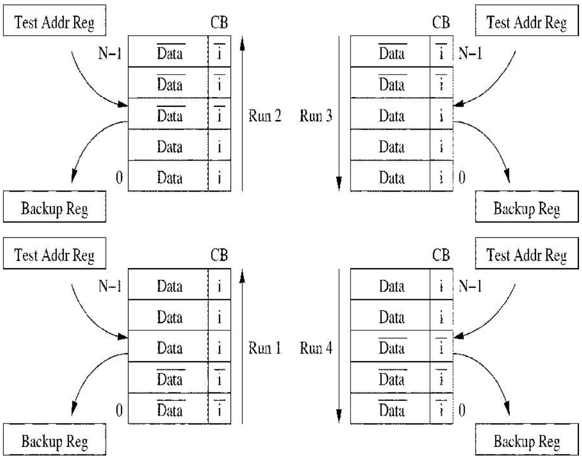


FIGURE1: Illustration of TOMT test cycle

Because the fetch operation performed by the word-level test at the beginning of each word's march test already gets the initial contents of the word under test, we are able to skip the initial *r0* in the second line. This allows to merge the final march element ↕{*r0*} by reordering the read/write pairs.

The TOMT algorithm can be implemented in a hardware module as shown in figure2. This TOMT unit is inserted into the data path between memory and processor. The implementation of the TOMT unit is performed using the hardware description language Verilog.

In test mode (i.e., while the processor does not issue a memory access) the Access Multiplexer (5) and the Write Data Multiplexer (6) give the Test Control Unit (4) full access to control, address, and data signals of the memory. Thus, by having full control over the memory, the Test Control Unit executes the TOMT algorithm.

Upon a read issued by the processor, the Test Control Unit suspends operation and the Access Multiplexer directly passes address and control signals from the processor to the memory thus prioritizing processor access over test. In parallel to the ongoing memory read access, the Test Control Unit compares the issued read address with its current test address, determines the appropriate data representation (inverted or non-inverted) and switches the Read Data Multiplexer (9) accordingly, i.e., either using or bypassing the Read Data Inverter (8). As soon as valid memory data are available the data path to the processor is already prepared. While the processor reads the data the Error Check Unit (7) checks the error code and issues an error indication if necessary.

A two-stage pipeline (1,3) speeds up processor write accesses by splitting them into two phases. During the first phase the Test Control Unit determines the appropriate data representation, again by an address comparison.

In parallel, the Check Bit Generator (2) calculates the required check bits according to the chosen memory protection scheme. The actual write access is performed during the second phase. Care should be taken to resolve the pipeline orderly for all sequences of write and read accesses. Comparing with other approaches, TOMT is capable of servicing every memory access of the processor instantaneously, i.e., within the same clock cycle.

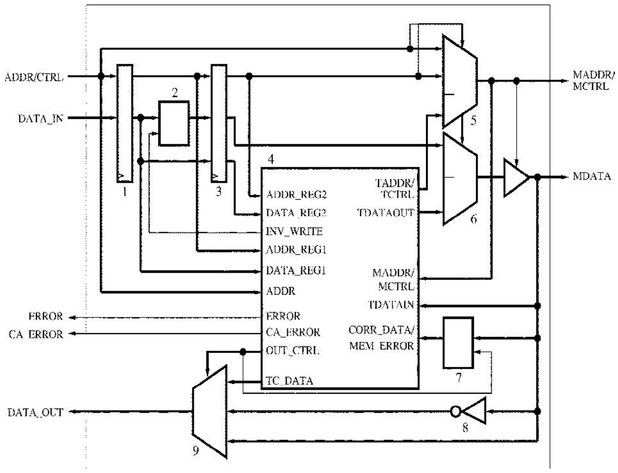


FIGURE 2: TOMT UNIT(1 first pipeline register, 2 check bit generator, 3 second pipeline register, 4 test control unit, 5 access multiplexer, 6 write data multiplexer, 7 error check unit,8 read data inverter,9 read data multiplexer)

V. RESULTS

The implementation of the TOMT unit has done in the hardware description language Verilog. The RTL schematic for the TOMT unit is shown in the figure3.

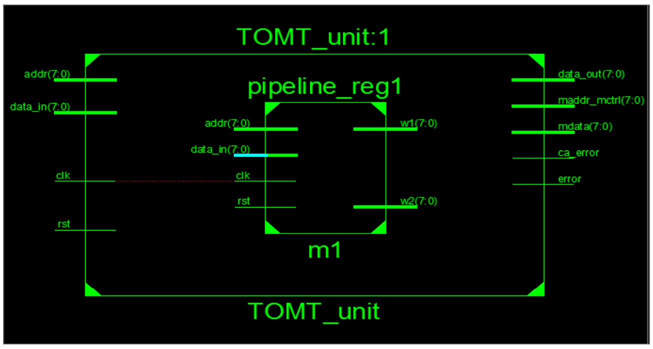
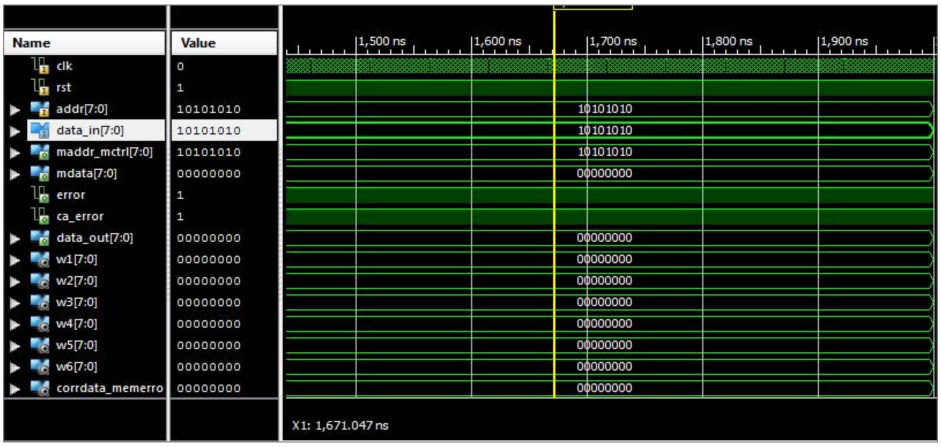


FIGURE3: RTL schematic of TOMT unit

The output of the TOMT unit is shown in the figure4. The figure4 shows the input and output are unequal and there is an error detected in the figure.



Name	Value
clk	0
rst	1
addr[7:0]	10101010
data_in[7:0]	10101010
maddr_mctrl[7:0]	10101010
mdata[7:0]	00000000
error	1
ca_error	1
data_out[7:0]	00000000
w1[7:0]	00000000
w2[7:0]	00000000
w3[7:0]	00000000
w4[7:0]	00000000
w5[7:0]	00000000
w6[7:0]	00000000
corrdata_memerro	00000000

FIGURE 4: Output of TOMT Unit

VI. CONCLUSION AND FUTURE SCOPE

CONCLUSION: An implementation of TOMT Algorithm for supporting online test in memory was presented. Thus online memory test approach proposed here is based on word-level algorithm which interacts with bit-level March tests. Various module level test of the algorithm is checked and found working. There were some integration problems that need to be addressed for the complete working of TOMT Algorithm.

FUTURE SCOPE: Future work will be directed toward hardware implementation in an FPGA for evaluation purposes. Due to its fully transparent operation and its moderate access time penalty TOMT can easily be supplemented to existing system designs.

ACKNOWLEDGEMENT

Authors thank Shri. BHVSN Murthy, the Director, RCI, DRDO and Shri. MVYS Ravi Kumar, Director, R&QA for their constant support and encouragement and also for providing facilities to carry out this work.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An UGC Approved Journal)

Website: www.ijareeie.com

Vol. 6, Issue 8, August 2017

REFERENCES

[1] K. Johansson, P. Dyreklev, B. Granbom, M. C. Calvet, S. Fourtine, and O. Feuillatre, “In-flight and ground testing of single event upset sensitivity in static RAMs,” *IEEE Trans. Nucl. Sci.*, vol. 45, pp. 1628–1632, 1998.

[2] C. I. Underwood, “The single-event-effect behavior of commercial- off-the-shelf memory devices—a decade in low-earth orbit,” *IEEE Trans. Nucl. Sci.*, vol. 45, pp. 1450–1457, 1998.

[3] M. Nicolaidis, “Transparent BIST for RAMs,” in *Proc. IEEE International Test Conference*, 1992, pp. 598–607.

[4] S. Hellebrand, H.-J. Wunderlich, A. Ivaniuk, Y. Klimets, and V. N. Yarmolik, “Error detecting refreshment for embedded DRAMs,” in *Proc. VLSI Test Symp.*, 1999, pp. 384–390.

[5] D. C. Huang, W. B. Jone, and S. R. Das, “An efficient parallel transparent BIST method for multiple embedded memory buffers,” in *Proc. Int. Conf. VLSI Design*, 2001, pp. 379–384.

[6] K. Thaller, “A highly-efficient transparent online memory test,” in *Proc. IEEE Int. Test Conf.*, 2001, pp. 230–239.

[7] M. Sachdev, “Open defects in CMOS RAM address decoders,” *IEEE Design & Test of Computers*, vol. 14, no. 2, pp. 26–33, 1997.

[8] B. Nadeau-Dostie, A. Silburt, and V. K. Agarwal, “Serial interfacing for embedded memory testing,” *IEEE Design & Test of Computers*, vol. 7, no. 2, pp. 52–63, 1990.

[9] K. Thaller, “A Transparent Online Memory Test,” Doctoral,Vienna University of Technology, 2001.

[10] S. Hamdioui and A. J. v. d. Goor, “An experimental analysis of spot defects in SRAMs: Realistic fault models and tests,” in *Proc. Asian Test Symp.*, 2000, pp. 131–138.

[11] A. J. v. d. Goor, *Testing Semiconductor Memories, Theorie and Practice*. New York: John Wiley & Sons, 1991.