Late Breaking Results: A data compaction strategy for extensive test flows of memories embedded in automotive SoCs

P. Bernardi, B. Borio, G. Insinga, B. Mendicino
Dipartimento di Automatica e Informatica
Politecnico di Torino, Italy

M. Battilana, M. Coppetta, N. Mautone,
P. Scaramuzza, F. Tengler, R. Ullmann
INFINEON Technologies, Germany and Italy

***Abstract*—Embedded memories are an essential component of modern System-on-Chips (SoCs). As memory requirements are constantly increasing, embedded memories occupy a significant percentage of the die area and are one of the main contributors to the yield of the devices. Manufacturers must conduct thorough testing to assess the reliability of their products, particularly in safety-critical environments like automotive applications. A typical automotive-grade memory test flow comprises several tests under varying conditions, including temperature and operating frequency. The SoC under test executes these tests, and they typically generate an extensive amount of diagnostic data that needs to be exported to the external world in time-consuming communications with the external testers. The computationally easiest way to encode the diagnostic data is through a list-based method, in which each single fault is logged individually, but is not particularly efficient with huge number of faults.**

This paper presents an optimized on-chip fault encoding algorithm that combines an efficient fault shape encoding method with only the encoding of the differences between each test. Experimental results collected in a simulation environment, where 10,000 devices were modeled to realistically represent fault evolution between consecutive memory tests, demonstrate an average reduction of 66.99% in diagnostic memory space requirements compared to previous State-Of-The-Art (SOTA) solutions.

***Index Terms*—Automotive SoC, reliability, memory diagnosis, encoding**

I. Introduction

Embedded Memories (eMemories) occupy a significant percentage of the die area in modern Automotive SoCs and significantly impact device yield. To assess the reliability of their devices, Automotive SoC manufacturers need to deeply test their devices in a series of tests that compose the test flows. These tests are executed in a variety of conditions including temperature, operating frequency and memory timing settings [1]. These tests, executed by the CPU or the Memory Built-In-Self-Tests (MBISTs) [2] [3] of the device itself, generate huge quantities of diagnostic data that need to be sent to the external tester. To export the data, time-consuming communication with external testers is required every time a small diagnostic buffer is filled inside the device. The communication itself may be one of the highest contributors to the total test time of the SoCs. This is especially true for less optimized diagnostic data encoding schemes such as the one in [4] that simply encode each fault in the memory in a list-based fashion. A possibility would be to resort to compression method at the cost of precise diagnostic data [5]. To maintain a lossless representation, however, a more efficient method encodes the fault shapes in a series of colored segments [6]. Using such advanced methods, it is possible to permanently store all the diagnostic life of the device directly on a small integrated diagnostic memory. However, even this efficient encoding may not be enough to efficiently store all the diagnostic history of the device on the chip and in a limited space, particularly when the SoC undergoes several memory tests.

This paper presents an optimized on-chip fault encoding algorithm that further expands the capabilities of the [6] algorithm by encoding just the differences between the first test and the subsequent ones. This approach leverages the properties of the stuck-at bit cell faults to be stuck at a certain value, thus resulting in useless repetition of the same fault information along several tests. Experimental tests on data coming from 10,000 realistically modeled devices shows an average reduction of 66.99% on the diagnostic memory requirements with respect to the method presented in [6].

II. Background

The method described in [6] takes advantage of the regular grid structure of the eMemories. Instead of encoding each single failed cell in a list structure as in [4], the method in [6] encodes the faults in colored segments called slices. Figure 1 shows the four possible colors of this encoding strategy:

- Black Slice: Contains a single failed cell.
- Blue Slice: Contains two faults that are more than 2 cells away from each other.
- Red Slice: A series of faulty bit cells interleaved with working cells.
- Orange Slice: A series of adjacent faults

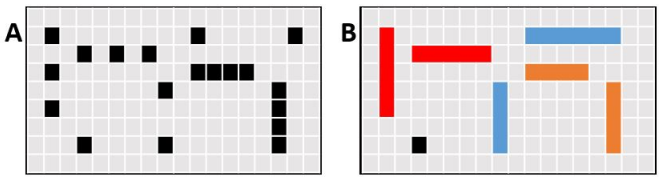

Fig. 1: A) Fault bitmap where each black square is a failing memory bit B) Encoded slice representation

III. Proposed Approach

The proposed approach starts from the method disclosed in [6], expanding its capabilities to report just the differences between a base test and the subsequent tests. In this way, the absence of information serves as information itself, and only new (or disappearing) faults are encoded. The concept is graphically explained in Fig.2. In the example, a certain number of faults were discovered and encoded in the first test. Then, a second test is executed, resulting in a similar fault shape configuration. As can be seen from Fig. 2, the proposed algorithm just encodes the differences between the first test and the following one, reducing the amount of diagnostic data required while still maintaining the lossless nature of the method presented in [6].

To achieve its goal, the algorithm efficiently stores the order of fault arrival in colored segments called delta slices, as can

978-3-9826741-0-0/DATE25/© 2025 EDAA

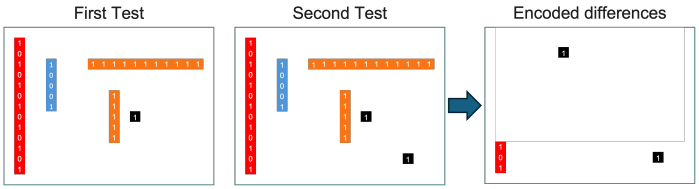

Fig. 2: Encoded differences after two memory tests.

be seen in Fig. 3. These slices are temporary and only used internally by the algorithm to save faults' order of arrival. In the example, the memory is tested from left to right and from top to bottom. The resulting Delta Slices are placed in an ordered structure with the first slices containing the first fault found.

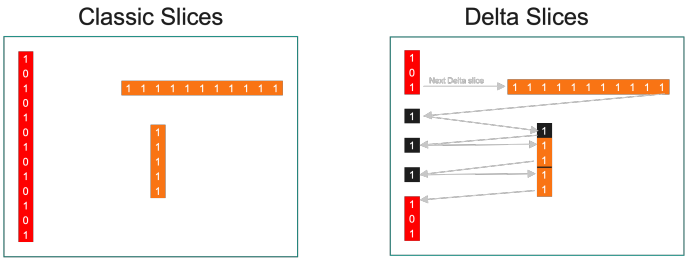

Fig. 3: Classic slices created from [6] and the temporary Delta ones that represents the order of faults arrival.

From the second memory test onward, the algorithm computes the Next Expected Fault (NEF) and check it against the received faults. For example, at the beginning of the second memory test, the algorithm picks the first fault received during the first test and takes it as the first fault of the first Delta Slice. Every time the algorithm receives a fault to analyze, it follows Algorithm 1.

Algorithm 1 Fault Handling Algorithm

```
1: if incomingFault before NEF then
2:     encodeDifference(incomingFault)
3: else if incomingFault after NEF then
4:     while incomingFault after NEF do
5:         encodeDifference(NEF)
6:         NEF ← selectNextNEF()
7:     end while
8: end if
```

IV. EXPERIMENTAL RESULTS

To validate the proposed approach, a simulator was developed in Rust to perform both [6] and the proposed algorithm. The simulator was fed with data coming from 10,000 devices modeled to realistically represent fault evolution between memories undergoing a typical Automotive eMemories test flow composed of four memory tests. A simplified example of how this devices are modeled is shown in Fig. 4. An algorithm generates realistic fault shapes for the first memory step and then expand them and/or adds other faults for the subsequent tests.

The experimental results on these 10,000 devices are summarized in Table I.

As Table I shows, the proposed approach on average saves 66.99% of diagnostic memory space when compared with the method in [6], meaning that devices under test would be able to store more amount of data in case of limited on-chip diagnostic memory space.

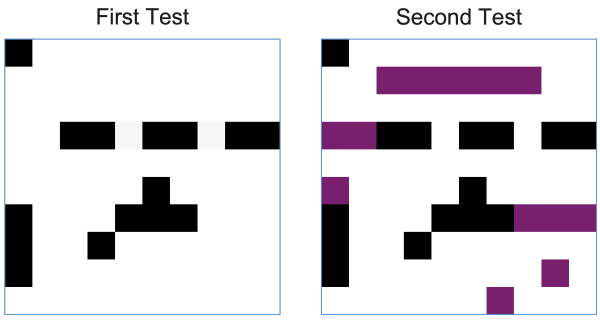

Fig. 4: In the modeled devices at each subsequent test faults are expanded and/or new ones are created (purple squares).

TABLE I: Aggregated data for 10,000 devices tested with the methods in [4], [6] and the proposed approach.

For 10000 devices	Approach in [4]	Approach in [6]	Proposed approach
Memory Usage (MB)	1865.9	72	23.8
Memory Saving wrt to [6] (%)	-2591.5%	//	66.99%

V. CONCLUSIONS

This paper presented an optimized difference-based diagnostic data compaction algorithm for memories embedded in Automotive SoCs. The algorithm efficiently represents fault patterns by encoding only the differences between successive memory test steps, taking advantage of the persistent nature of stuck-at faults in eMemories. The differences are encoded using an efficient color-coded segment approach called slices, which builds upon existing SOTA method. Experimental results collected in a simulation environment, where 10,000 devices were modeled to realistically represent fault evolution between consecutive memory tests, demonstrate that the proposed approach achieves superior compression efficiency compared to SOTA approaches, showing an average reduction of 66.99% in diagnostic memory requirements to loslessly represent all the faults in the memory. This significant improvement enables more efficient use of limited on-chip diagnostic memory space, making it particularly valuable for automotive applications where extensive memory testing is required.

REFERENCES

[1] P. Bernardi et al., "Recent Trends and Perspectives on Defect-Oriented Testing," 2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS), 2022, pp. 1-10, doi: 10.1109/IOLTS56730.2022.9897647.
[2] S. Abhas, M. K. Gurram e A. Abhijit, "Controller Architecture for Memory BIST Algorithms" in IEEE International Students' Conference on Electrical, ELectronics and Computer Science (SCEECS), 2020.
[3] I. Schanstra et al. "Semiconductor Manufacturing Process Monitoring using Built-In Self-Test for Embedded Memories" in Proceedings International Test Conference 1998.
[4] A. L. Landzberg e R. Van Nostrand, Microelectronics Manufacturing Diagnostics Handbook, New York, USA, 1993.
[5] J. Chen, J. Khare, K. Walker, S. Shaik, J. Rajsky e W. Maly, "Test response compression and bitmap encoding for embedded memories in manufacturing process monitoring" in Proceedings International Test Conference 2001.
[6] P. Bernardi et al., "Optimized diagnostic strategy for embedded memories of Automotive Systems-on-Chip," 2022 IEEE European Test Symposium (ETS), 2022, pp. 1-6, doi: 10.1109/ETS54262.2022.9810445.