Hierarchical Memory Diagnosis

Medeiros, G. C.; Fieback, M.; Gebregiorgis, A.; Taouil, M.; Poehls, L. B.; Hamdioui, S.

DOI
10.1109/ETS54262.2022.9810467

Publication date
2022

Document Version
Final published version

Published in
Proceedings - 2022 IEEE European Test Symposium, ETS 2022

Citation (APA)
Medeiros, G. C., Fieback, M., Gebregiorgis, A., Taouil, M., Poehls, L. B., & Hamdioui, S. (2022). Hierarchical Memory Diagnosis. In *Proceedings - 2022 IEEE European Test Symposium, ETS 2022* (Proceedings of the European Test Workshop; Vol. 2022-May). IEEE. https://doi.org/10.1109/ETS54262.2022.9810467

2022 IEEE European Test Symposium (ETS) | 978-1-6654-6706-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ETS54262.2022.9810467

Hierarchical Memory Diagnosis

G. C. Medeiros[1], M. Fieback[1], A. Gebregiorgis[1], M. Taouil[1], L. B. Poehls[2], S. Hamdioui[1]
[1]TU Delft, The Netherlands [2]IDS, RWTH Aachen, Germany
Email: {G.CardosoMedeiros, S.Hamdioui}@tudelft.nl

***Abstract*—High-quality memory diagnosis methodologies are critical enablers for scaled memory devices as they reduce time to market and provide valuable information regarding test escapes and customer returns. This paper presents an efficient *Hierarchical Memory Diagnosis* (HMD) approach that accurately diagnoses faults in the entire memory. Faults are diagnosed hierarchically; first, their location, then their nature (i.e., static or dynamic), and finally, their functional fault model. The HMD approach leads to a more accurate diagnostic, enabling the precise identification of yield loss causes.**

***Index Terms*—Memory, Diagnosis, Fault, Test, Algorithm**

I. Introduction

The quick identification of yield loss' root causes is critical for ramping up yield learning and shortening time to market. Memory diagnosis is a powerful tool for identifying these failure roots. However, they must cover a vast set of faults. For example, they must cover dynamic faults, i.e., faults sensitized by more than one operation, which are more common in scaled memories [1]. Furthermore, it must cover faults from outside the memory array that require special test algorithms [2], such as decoder and peripheral faults. Finally, they ideally also cover *hard-to-detect* (HTD) faults, i.e., faults whose detection is not guaranteed by simply writing and reading the memory, and require additional dedicated testing circuits [3, 4]. Any of these faults are prone to become test escapes if not detected, thus leading to *no-trouble-found* (NTF) devices [5].

Memory diagnosis approaches use various methods to identify and diagnose memory faults, e.g., fault pattern identification [6–8], fault signatures [1, 9, 10]. However, they hardly make any distinction between memory array faults and faults in other parts of the memory, such as decoders and peripherals. Furthermore, dynamic faults are only partially covered. Moreover, the diagnosis approaches relying on signatures are not easily extensible as they demand redefining the whole scheme for any modification done to the targeted faults. Thus, there is a need for a memory diagnosis methodology that covers different faults in any part of the memory chip.

This paper proposes a systematic memory diagnosis approach, namely *Hierarchical Memory Diagnosis* (HMD), to speed up the diagnosis of embedded memories, improve yield learning, and ease the characterization of customer returns and NTFs. HMD is a comprehensive approach that covers static and dynamic faults in the *entire* memory in a hierarchical manner: first, the location, then, the nature (i.e., static or dynamic), and finally, the fault model. It is *platform-independent* as it does not require a specific implementation, and *easily extensible* as it does not require recompiling signatures to diagnose new faults.

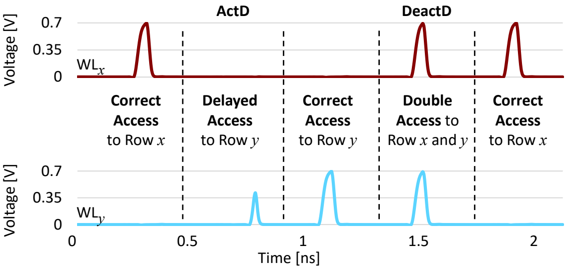

Figure 1. An ActD and DeactD Row Decoder Fault.

II. Hierarchical Memory Diagnosis

The HMD approach comprises four steps, as described next. We illustrate the proposed approach by using the diagnosis of dynamic row decoder faults as an example.

*1) **Fault Space Definition**:* the HMD approach assumes five significant memory blocks: row and column decoder, write and read path, and memory array. Each block contains its own fault space, which includes static (i.e., faults sensitized by at most one operation) and dynamic (i.e., faults sensitized by more than one operation) faults. Due to space limitation, we do not discuss the diagnosis of HTD faults. The fault space of dynamic row decoder faults specifically includes Activation (ActD) and Deactivation Delay (DeactD) [11]. ActD hinders WLs' activation, i.e., the WL is delayed and not fully activated, while DeactD hinders the WLs' deactivation, leading to simultaneously activating two WLs; these faults are observed only in *specific* address transitions [11]. Fig. 1 illustrates how WLs are affected by these faults due to a partial open defect in a pre-decoder. The faulty behavior comes from the pre-decoders failing to decode the address in due time; a timing signal deactivates WLs between operations. In the first operation, WL$_x$ is correctly generated. However, when switching from addresses A_x to A_y in the second cycle, WL$_y$ suffers from ActD. Although the subsequent access of WL$_y$ in the third cycle passes correctly, two WLs are activated in the fourth cycle when a transition from A_y back to A_x is performed: WL$_x$, which is correctly accessed, and WL$_y$, which suffers from DeactD; hence, the simultaneous access of two addresses.

*2) **Level 1 – Fault Location**:* level 1 must sensitize and detect *all* faults in the targeted fault space to prevent test escapes. To do so, we developed Alg. 1, which makes use of march notation [12]: $\Uparrow$, $\Downarrow$, and $\Updownarrow$ denote increasing, decreasing, and irrelevant address access order, respectively. $w0$, $w1$, $r0$, $r1$ represent the operations write '0', write '1', read '0', and

This work is supported by European Union's Horizon 2020 Research and Innovation through RESCUE project under grant 722325 and by the Federal Ministry of Education and Research (BMBF, Germany) within the NEUROTEC I and II projects 16ES1134 and 16ME0399

978-1-6654-6706-3/22/$31.00 ©2022 IEEE

Algorithm 1 March HMD-LVL1

```
M1: $\{\Updownarrow (w1);$
M2: $\Uparrow (w0, r0, r0, r0, r0, r0, r0, r0, r0, w0, w1, w1);$
M3: $\Uparrow (r1, r1, r1, r1, r1, r1, r1, r1, w1, w0, w0);$
M4: $\Downarrow (r0, r0, r0, r0, r0, r0, r0, r0, w0, w1, w1);$
M5: $\Downarrow (r1, r1, r1, r1, r1, r1, r1, r1, w1, w0, w0);$
M6: $\Downarrow (w1, r1, r1, r1, r1, r1, r1, r1, r1, w0);$
M7: $\Uparrow (r0, r0, r0, r0, r0, r0, r0, r0)\}$
```

Table I
FC AND TC SPACE FOR DYNAMIC ROW DECODER FAULTS

FC	Faults	TC
FC1	Only ActD	$w0_y, w0_x, \underline{w1}_y, r1_y$
FC2	Only DeactD	$w0_y, r0_y, \underline{w1}_x, r0_y, r0_y$
FC3	Both ActD and DeactD	Detected by TC1 and TC2

read '1', respectively. March HMD-LVL1 is applied using special stress combinations to detect dynamic row decoder faults [2]: hamming-distance-based addressing method with fast-row. ActD is sensitized and detected by M2 if $A_x < A_y$ or by M6 if $A_x > A_y$. DeactD is sensitized and detected by M6 and M7, respectively, if $A_x < A_y$, or by M2 and M3 if $A_x > A_y$. Faults in each memory block (e.g., column decoder, write path, memory array) will generate different and unique bitmap patterns, enabling accurate faulty block identification. For dynamic row decoder faults, single or multiple rows will fail. Moreover, knowing the set of failing addresses and addresses transitions (e.g., from A_x to A_y) also enables the precise fault identification in levels 2 and 3.

*3) **Level 2 – Fault Nature**:* diagnosis level 2 applies customized diagnostic algorithms to sensitize *static faults* in level 1's failing addresses; if no static faults are sensitized, it is assumed the faulty block suffers from *dynamic faults*. We develop algorithms targeting only static faults within one block. For the row decoder, the specific algorithm is $\{\Updownarrow (w0); \Uparrow (r0, w1, r1)\}$; it is applied to the failing row identified in level 1 using linear addressing mode and fast-column [2]. No faults are triggered as there were no specific address transitions [11]; therefore, the faulty block suffers from dynamic faults.

*4) **Level 3 – Fault Model**:* diagnosis level 3 diagnoses fault models using *Fault Classes* (FC) and *Test Classes* (TC) [10]. An FC contains faults with the same sensitizing and detecting conditions that are *externally* indistinguishable. A TC is an algorithm designed to detect a particular FC; their pass/fail information is used to generate a unique signature. Table I shows the FCs for dynamic row decoder faults and their TCs; FC3 does not require a TC as TC1 and TC2 cover it. The TCs are described by sequences of operations and specific address transitions (i.e., from A_x to A_y and vice-versa); operations that sensitize the targeted fault are underlined.

Fig. 2 shows the simulation of TC1, which targets ActD; it shows WL$_x$ and WL$_y$ (faulty row) and the contents of cells in these rows. ActD is sensitized by $w1_y$, i.e., the transition write operation fails. DeactD is detected by applying TC2. It is sensitized in $w1_x$ (3rd operation); WL$_y$ is still enabled when accessing A_x. Two read operations are necessary to detect DeactD as the first one may be affected by an ActD, e.g., the read operation does not access the cell and returns the same value from the last read operation. With the pass/fail results of both TCs, the FC x TC signature dictionary for dynamic row decoder faults can be generated, as shown by Table II.

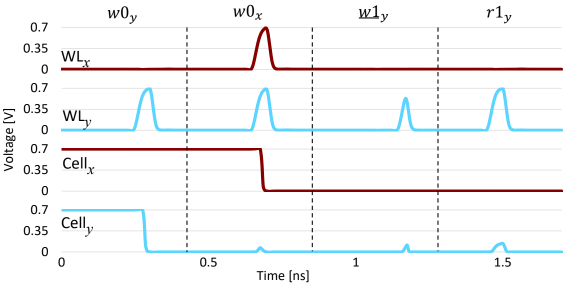

Figure 2. Applying TC1 on a memory suffering an ActD and DeactD fault.

Table II
FC x TC DICTIONARY FOR DYNAMIC ROW DECODER FAULTS.

FC	TC1	TC2	Signature
FC1	1	0	10
FC2	0	1	01
FC3	1	1	11

III. Discussion & Conclusion

We have presented a new diagnosis methodology named *Hierarchical Memory Diagnosis* (HMD). The approach was validated through defect injection and circuit simulations. Based on the obtained results, we conclude the following:

Added Value: the HMD can be applied during characterization to improve the design and manufacturing process, thus boosting manufacturing yield and speeding time to market. It can also be applied during the analysis of customer returns to help understand test escapes and NTF devices.

Key Differentiators: HMD surpasses existing methodologies by covering *all types of faults* in *all parts* of the memory. HMD is *easily extensible* as new capabilities can be integrated with existing ones without the need to ensure unique signatures. Furthermore, HMD is platform-independent; it can be applied to all sorts of memories. Moreover, it does not require dedicated diagnosing circuits, i.e., design-for-diagnosis [1].

Limitations: the HMD approach cannot indicate the aggressor cell's location of coupling faults, only the address relation between aggressor and victim.

References

[1] G. Harutyunyan *et al.*, "Memory Physical Aware Multi-Level Fault Diagnosis Flow," *IEEE Tran. on Emerging Topics in Computing*, 2018.

[2] A. J. van de Goor *et al.*, "Detecting faults in the peripheral circuits and an evaluation of SRAM tests," in *Int. Conf. on Test*. IEEE, oct 2004.

[3] J. Kinseher *et al.*, "Improving SRAM test quality by leveraging self-timed circuits," in *Des., Test, and Automation in Eur.* IEEE, mar 2016.

[4] G. C. Medeiros *et al.*, "Detecting random read faults to reduce test escapes in FinFET SRAMs," in *2021 IEEE Eur. Test Symp. (ETS)*, May 2021.

[5] G. C. Medeiros *et al.*, "Hard-to-Detect Fault Analysis in FinFET SRAMs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*, 2021.

[6] C.-W. Wang *et al.*, "Fault pattern oriented defect diagnosis for memories," in *International Test Conference*, 2003.

[7] K.-L. Cheng *et al.*, "FAME: a fault-pattern based memory failure analysis framework," in *Int. Conf. on Computer Aided Design (ICCAD)*, 2003.

[8] M. d. Carvalho *et al.*, "Optimized embedded memory diagnosis," in *14th IEEE DDECS*, 2011.

[9] A. Ney *et al.*, "A signature-based approach for diagnosis of dynamic faults in srams," in *2008 3rd International Conference on Design and Technology of Integrated Systems in Nanoscale Era*, 2008.

[10] Z. Al-Ars *et al.*, "Fault diagnosis using test primitives in random access memories," in *2009 Asian Test Symposium*, 2009.

[11] S. Hamdioui *et al.*, "An investigation on capacitive coupling in RAM address decoders," in *Int. Design and Test Workshop*. IEEE, dec 2007.

[12] A. van de Goor, *Testing Semiconductor Memories: Theory and Practice*, From Chips to Boards. J. Wiley & Sons, 1991.