2024 29th IEEE European Test Symposium (ETS)
- Industry Paper -

Combining Built-In Redundancy Analysis with ECC for Memory Testing

Luc Romain
Siemens Digital Industries Software
Ottawa, Canada
luc.romain@siemens.com

Paul-Patrick Nordmann
Siemens Digital Industries Software
Hamburg, Germany
paul-patrick.nordmann@siemens.com

Benoit Nadeau-Dostie
Siemens Digital Industries Software
Ottawa, Canada
benoit.nadeau-dostie@siemens.com

Lori Schramm
Siemens Digital Industries Software
Atlanta, GA, USA
lori.schramm@siemens.com

Martin Keim
Siemens Digital Industries Software
Orlando, FL, USA
martin.keim@siemens.com

2024 IEEE European Test Symposium (ETS) | 979-8-3503-4932-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ETS61313.2024.10567190

***Abstract*—Error Correction Codes (ECC) in current designs typically serve two purposes. For emerging non-volatile memory types (NVM), such as embedded magnetoresistive random access memory (eMRAM), ECC is necessary to counter the probabilistic behavior of the NVM, rendering the combined NVM/ECC a deterministic memory again. The second and much more prominent usage of ECC today is to protect the system against transient faults in the memory, here typically for SRAMs. On the other hand, new defect types for such emerging memories and new technology nodes may exceed reasonable costs of conventional row and column repair. To improve yield, users explore ECC as an option to augment the repair capability of the memory. This paper brings such an augmentation into a standard memory test and repair flow. It allows a user-defined, post silicon trade-off of using all or parts of the corrective power of the ECC for yield improvements and/or for system protection. Experimental results underline the very low area cost of this augmentation.**

***Index Terms*—Error Correction Codes (ECC), Built-In Redundancy Analysis (BIRA), Built-In Self Repair (BISR)**

I. Introduction

Since NVM technology is still under development, ECC logic and redundant elements are commonly used to improve device reliability [1], [2]. The test and repair solution should offer the flexibility to adjust the number of defects to be corrected using ECC and to collaborate with the Built-In Redundancy Analysis (BIRA) circuitry during memory manufacturing test. Using ECC correction and redundant elements during manufacturing is proposed in [3]. However, most of the techniques rely on failure data collection and off-chip post-processing to determine the final redundant element allocations. This paper describes an autonomous on-chip solution consisting of an augmented BIRA circuitry capable of calculating an optimal repair solution using both redundant elements and ECC, given user-defined requirements. An overview of the standard repair test flow is shown in Section II. The augmented BIRA circuit, referred to as ECC-aware BIRA, is presented in Section III. The built-in solution uses the memory's ECC correction capability and redundant elements to improve manufacturing yield, especially for NVMs. It can also be used with ECC-equipped SRAMs during in-system memory testing. Furthermore, we show a test sequence with fault injection where the ECC-aware BIRA hardware automatically allocates redundant elements in an optimal way. Simulation results are provided to demonstrate the functionality of the proposed solution. Finally, we analyze the area and performance of the solution using a logic synthesis tool in SectionIV.

II. Standard Memory Repair Flow

The memory test hardware involved in testing memories with redundant elements is shown in Figure 1. The memory interface is connected to the memory, on the right, and the memory test controller is on the left. A BIRA module is located inside the memory interface. The elements in red illustrate the new ECC-aware components, which will be introduced in Section III-B. These elements do not participate in the standard memory test flow. The comparator registers

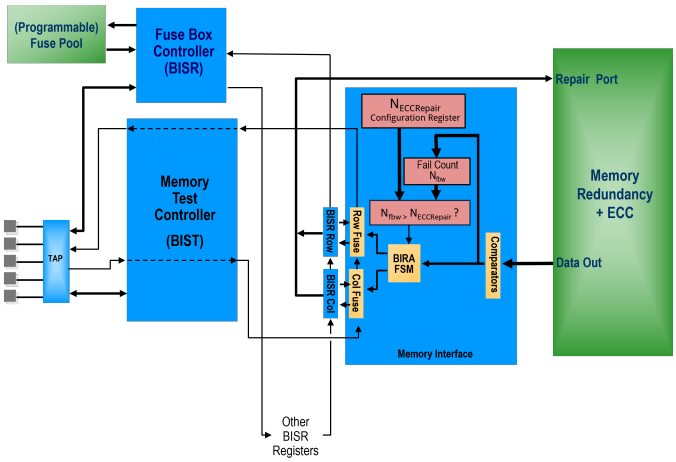

Figure 1: Built-in memory test architecture overview

capture the defect locations during the memory test execution. The BIRA logic uses the values of these registers to calculate a