A mechanism of row redundancy sharing for reparable memory systems

Publisher: IEEE

Karen Amirkhanyan ; Samvel Shoukourian ; Valery Vardanian All Authors

Abstract: In this paper, we proposed a method for implementation of a mechanism for “redundancy sharing” allowing to repair a fault/defect in a memory instance from a memory system... **View more**

Metadata

Abstract:
In this paper, we proposed a method for implementation of a mechanism for “redundancy sharing” allowing to repair a fault/defect in a memory instance from a memory system with hundreds/thousands of memory instances with an available shared redundant element of another memory instance in the same group of the Memory System that has been preliminarily grouped with respect to its main parameters, e.g. clock, power and position of instances. The calculations showed that the hardware is saved to a great extent with a negligible impact on memory's functional performance. Not only excluded redundancies contribute in the area saving but also reduction of the redundancy registers due to reduction of redundancies, as well as reduction of the control logic and the number of necessary fuses.

Published in: 2017 Computer Science and Information Technologies (CSIT)

Date of Conference: 25-29 Sept. 2017

Date Added to IEEE *Xplore*: 12 March 2018

ISBN Information:

INSPEC Accession Number: 17631387

DOI: 10.1109/CSITechnol.2017.8312140

Publisher: IEEE

Conference Location: Yerevan, Armenia

Contents

I. Introduction

Built-in Self-Repair (BISR) is widely used for improving memory-core yield. Although, the portion of a BISR circuit with respect to the area of the corresponding memory instance area is usually small, but due to the number of hundreds (or even possibly, thousands) of memory instances in a Memory System (MS), when each memory instance with redundancy had its own BISR circuit, the total area of the BISR circuits in an MS increased to a great extent. To decrease the overall area overhead of BISR circuits in an MS, many researchers proposed some notions of "shared BISR", grouping, reusing, multiple shared buses, etc. (see [1]–[12]). They reduced the area overhead by performing test and repair of memory instances serially that increased the time for test and repair. To reduce area overhead and test & repair time, "shared BISR" was used performing test & repair of multiple memory instances in parallel. Hence, grouping is another approach to be applied with respect to the typically great number of memory instances in an MS to make use of the property of structural identity of memory instances from the same group.

Authors
Figures
References
Citations
Keywords
Metrics