Dynamic Sparing and Error Correction Techniques for Fault Tolerance in Nanoscale Memory Structures

Casey M. Jeffery, Ahmet Basagalar, and Renato J. O. Figueiredo

University of Florida, Gainesville, FL, 32611, United States

Abstract — **This paper describes a memory organization that supports hierarchical, dynamic fault tolerance mechanisms applicable to heterogeneous CMOS/molecular systems. It has been projected that, in such systems, programmable mapping circuitry becomes necessary when the interface between microscale and nanoscale address lines is formed in a non-deterministic manner. This mapping circuitry may double as a means of implementing more advanced forms of reconfiguration and error correction codes useful in detecting and recovering from runtime faults. Computer simulation is used to estimate the effectiveness of various configurations. Results show that dynamic remapping of spare memory rows allows for improvements in reliability over standard error correction codes.**

Index Terms — **Nanotechnology, memory architecture, memory fault tolerance, ULSI, simulation.**

I. Introduction

Continued advances in the fabrication of molecular structures has led to speculation that nanoelectronic devices may be incorporated with CMOS technology in the realm of ultra-dense logic and memory systems. Many proposed molecular systems involve self-assembly techniques which, although ideally suited for large, regular memory structures, are predicted to result in an increased hard error rate due to physical defects. This will be compounded by an increased sensitivity of the ever-smaller bit storage units to soft errors, e.g. resulting from noise and radiation effects. Current memory devices rely on fault tolerance schemes such as modified Hamming error correction codes (ECC) and static spare swapping to cope with these errors. It is conceivable, however, that such solutions may not be sufficient in systems that are larger and built using less reliable components and/or manufacturing techniques.

One approach to the design of heterogeneous systems with nanoscale pitch and microscale interfaces is the use of demultiplexor circuits. In this method the interface contacts may be formed in parallel using a nondeterministic process, at the expense of an approximate five-fold increase in the number of address lines and the need to map out the actual address of each location in memory [3]. The circuitry necessary to perform this mapping process, as well as test and repair processes, can be integrated into the memory using CMOS technology and will be collectively referred to as a processing node (PN).

The goal of this paper is to explore methods of providing an increased level of fault tolerance by continuing to make use of the PNs autonomously, throughout the lifetime of the memory to detect and correct errors. The proposed fault tolerance methods include the novel approach of combining ECC, reconfiguration, and graceful degradation all in a hierarchical, distributed manner. The techniques target ultra-dense, potentially nonvolatile memories that can, in the future, provide storage capacities typical of today's storage subsystems, while supporting random-access memory interfaces.

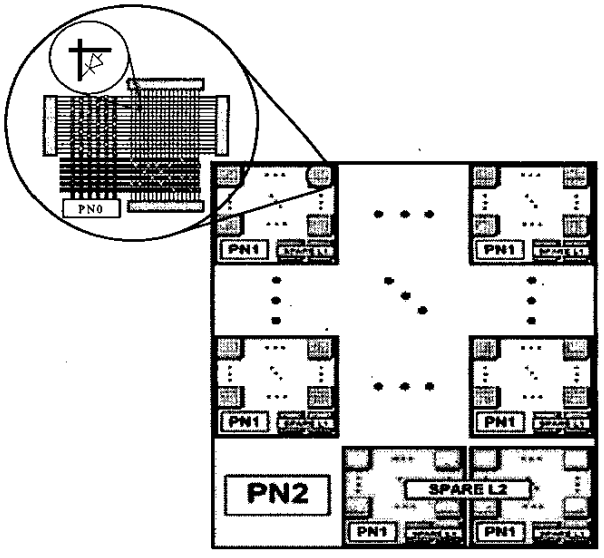

Fig. 1. Hierarchical memory architecture with spare devices and processing nodes at each level.

II. Overview of the Proposed Memory Architecture

Fabrication and performance considerations motivate the partition of ultra-large memory systems across smaller sub-arrays. The architecture used in this paper is based on

a hierarchical, divided-word-line (DWL) structure that logically connects the sub-arrays, as in Fig. 1. The hierarchical design follows principles similar to those presented in [2]. The lowest level (level-0) consists of relatively small, roughly symmetrical array of molecular storage cells with a single PN. The second level (level-1) is then an array of level-0 devices addressed by a PN, and similarly, the third level (level-2) is an array of level-1 devices with a PN. These three levels prove sufficient to present the ideas in the paper; nonetheless, the architecture can be expanded to include more than three levels.

A. Fault Tolerance Techniques at Level-0

The level-0 array is word addressable and has two possible forms of fault tolerance. The first is the use of Hamming ECCs with the capability of single-error-correction (SEC) or double-error-correction (DEC) that can be implemented on each word at the cost of parity bits, additional latency, and an increase in PN complexity. The other possibility is the use of redundant rows and/or columns that can be mapped in to repair faulty devices. The spare remapping leverages the circuitry already built into the PN. To minimize latency, no multi-cell sparing optimizations are attempted and spare use is initiated at the occurrence of each error, or after n errors in the case that an n-bit ECC is implemented.

The redundancy measures are initiated entirely on a local basis. There is no sharing of information or spares between level-0 PNs. The only external signal generated by the level-0 PN is one that warns the level-1 PN that spare availability is running low or that the error correction capability has been exceeded.

B. Fault Tolerance Techniques at Level-1

The second level of the memory consists of an array of level-0 devices. Similar to level-0, both ECC and redundancy are possible, but the implementations are slightly different. The form of ECC used at this level is similar in operation to fourth-level RAID used in hard disks [6]. For N level-0 arrays in a level-1 row, N-1 are used for data and the remaining one is used for storing parity bits based on the data arrays. This method allows for an uncorrectable error at the word level to be corrected by reading a word from each of the level-0 arrays and calculating the parity code. This form of redundancy has the cost of an additional column of level-0 arrays for each level-1 device, as well as the circuit complexity and latency associated with the parity calculation.

There is also the possibility of reserving a number of level-0 arrays as spare devices. In this case an entire array is swapped in response to uncorrectable errors. The decision to swap in a spare device is made based on the signals sent from the level-0 PNs.

C. Fault Tolerance Techniques at Level-2

The purpose of this level is to break the level-1 devices into groups across which the parity redundancy can be employed. The only form of fault tolerance currently being investigated at this level is the possibility of sparing entire level-1 arrays. The responsibility of level-2 PNs is thus to copy data into spare arrays and perform address remapping in response to signals received from level-1 PNs.

III. Experimental Setup and Results

Various configurations of the architecture described above have been simulated for memories with gigabit capacity. The purpose of these simulations, much as the analysis done in [8], is to find and exploit the greatest degree of synergism possible. It is not the place of this paper to relate these configurations' tradeoffs in absolute speed, size, or complexity, rather the conclusions made are meant to be technology independent. The implementation details must be taken into consideration when applying these techniques to a specific technology.

A. Simulator Structure

A simulator was written to efficiently model the performance of very large memories. It consists of three main parts:

1) Generation of hard errors for the memory lifetime
2) Simulation of PN responses to each of the errors
3) Recording of the memory state at specified intervals

The generation of errors is done using the algorithm defined in [1], which is based on order statistics. The soft error rate (SER) used is estimated based on an extrapolation of the rate given in [5] to the target memory size and estimated cycle times. The hard error rate (HER) is assumed to be roughly two orders of magnitude less than the SER. The bit errors for the entire lifetime of the memory are generated based on these values and randomly assigned to locations in the memory.

The responses of the PNs to each failure are then simulated. If the failure results in fewer than the number of correctable bits in a word, it is chained together with other errors in the same level-0 device in much the same way as [4]. If the failure results in the number of correctable bits or more being in error for a word, then a spare row or column is switched in and all errors removed from the system are disabled in the failure list. If a failure does cause an uncorrectable error at the lowest level or if the spares have been exhausted, the level-1 PN is

signaled. If the error is then correctable at level-1 with ECC information, the word error is chained together with other word errors in the same level-1 row. If a word error is not correctable at level-1, then a spare level-0 array is swapped in. In the case that an error is not correctable within level-1 or if the spares have been exhausted, then the level-2 PN is signaled to request a replacement at the highest level. Any of these fault tolerance techniques can be disabled.

The status of the entire memory, including the number of hard uncorrectable errors, soft uncorrectable errors, and spare usage is recorded at each polling interval. This provides a survey of the performance of the memory over an entire lifetime.

B. Simulation Results

The set of experiments shown in Fig. 2 looks at multi-level memory organizations with 1Gb usable size composed of 1Mb modules. The results are the average of 5,000 trials and show the number of hard uncorrectable errors (UEs) in the system after 100,000 hours (~11.4 years) of system life. It is observed again that SEC with spares has better performance than DEC initially, but this is not true later in the memory life as the spares are exhausted. It is also notable that the SEC implementations using RAID have a significant performance advantage for a given amount of overhead. The only configuration to protect the memory from all errors is DEC with spares, which requires a significantly higher overhead.

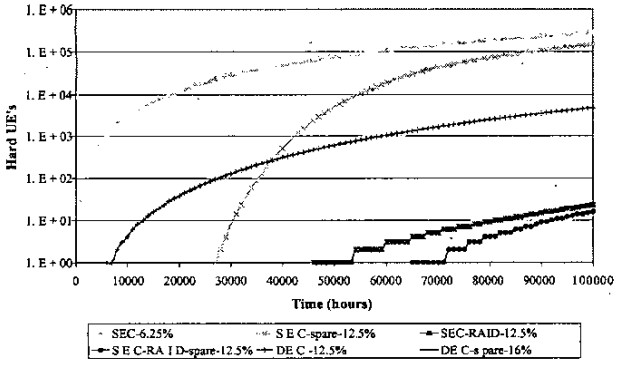

Fig. 2. Simulation results of hard UE's in 1Gb memory with 128-bit word size.

In the absence of analytical models for the techniques described in this paper, the validation of the simulator has been approached as follows. First, ECC without sparing has been evaluated against an analytical model from [7] for memories with SEC. Results averaged over 5,000 trials show small differences in the MTTF (1.4%) for a 1024x39-bit array. Second, ECC and sparing have been evaluated though the analysis of simulation trace outputs and by comparison against a previously developed simulator that implemented different algorithms, data structures (event-driven vs. chaining), random number generation (per-bit Poisson distribution vs. ordered statistics), and programming language (C vs. Perl).

IV. Conclusions and Future Work

A hierarchical memory organization for projected heterogeneous CMOS/molecular systems has been proposed. It is based on the use of processing nodes in a hierarchical fashion for dynamic remapping and error correction. Initial results have been investigated.

It has been shown through simulation that the use of spares in conjunction with ECCs can improve the reliability of the proposed memory system designs. These results are encouraging and motivate the investigation of additional configurations and failure modes, as well as the use of redundancy at other levels in the system.

Acknowledgements

This work material is based upon work supported by NASA under award no. NCC 2-1363

References

[1] C. L. Chen and R. A. Rutledge, "Fault-tolerant memory simulator" IBM J. Res. Develop., vol. 28, no. 2, pp. 184-194, 1984.

[2] T. Chen and G. Sunada, "Design of a self-testing and self-repairing structure for highly hierarchical ultra-large capacity memory chips" IEEE Trans. on VLSI Systems, vol. 1, no. 2, pp. 88-97, June 1993.

[3] P. J. Kuekes and R. S. Williams, "Demultiplexer for a molecular wire crossbar network" U.S. Patent 6,256,767, July 2001.

[4] M. R. Libson and H. E. Harvey, "A general-purpose memory reliability simulator" IBM J. Res. Develop., vol. 28, no. 2, pp. 196-205, 1984.

[5] Micron Tech. Inc., "DRAM SER Calculations" 1994.

[6] D. A. Patterson, G. Gibson, and R. H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)" ACM SIGMOD Record, vol. 17, no. 3, pp. 109-116, 1988.

[7] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing recovery-techniques for memory systems" IEEE Trans. on Reliab., vol. 39, no. 1, pp. 114-122, April 1998.

[8] C. H. Stapper and H. S. Lee, "Synergistic fault-tolerance for memory chips" IEEE Trans. on Computers, vol. 41, no. 9, pp. 1078-1086, September 1992.