

Error Correction Code Multi-Switch Nanomemory Demultiplexer: Performance and Reliability Analysis

Ayodeji Coker, *Student Member, IEEE* and Valerie Taylor, *Senior Member, IEEE*

Abstract— In this work, the performance and reliability analysis of a crossbar molecular switch nanomemory demultiplexer is studied and results presented. In particular, we investigate the impact on the performance of a crossbar nanomemory demultiplexer of implementing a combination of error correction coding and multi-switch junction fault tolerance schemes. Results indicate that delay and power scale linearly with increasing number of redundant molecular switch junctions. Results also show that by implementing a redundancy of $k = 3$ and error correction code Hamming distance $d = 2$, the demultiplexer reliability can be improved to approximately 99% when there is a 20% probability of errors occurring in the demultiplexer. If the Hamming distance is increased to $d = 3$, then a 30% error rate occurrence can be tolerated. The results of the access time delay analysis also indicate an improvement with respect to increasing redundancy and Hamming distance. Power dissipation penalties were shown to be relatively small when compared with reliability gains.

I. INTRODUCTION

Performance induced scaling in microelectronics is becoming increasingly difficult to sustain. This is due in large part to the fact that microelectronic devices are approaching the physical limits of their scalability. In order to prolong performance growth, engineers and scientists are investigating nanoelectronic devices as a potential solution. The dimensions at which nanoelectronic devices are fabricated leaves them highly susceptible to defects and operational faults. Consequently, one of the most pressing challenges of nanoelectronics is developing reliable methodologies for addressing circuits on a scale smaller than the resolution attainable through conventional lithographic processes.

Crossbar nanoelectronic memories [1, 2] are currently being evaluated as a potential replacement for microelectronic based memories. In this work we focus primarily on the reliability of demultiplexers which are used in crossbar nanomemories as a means of bridging microelectronic scaled circuits to their nanoelectronic scaled counterparts. Achieving reliability in demultiplexers becomes very important if functional crossbar nanomemories are to be realized. The reason being, faults which originate at the demultiplexer level can propagate through the entire nanomemory system.

II. DEMULTIPLEXER DEFECT-TOLERANT SCHEME

The role and location of the demultiplexers in the nanomemory hierarchy gives it a very critical role in the structure of a nanomemory. Faults and defects that originate at the decoder can propagate through the entire memory system leading to paralyzing system failures. In nanoelectronics, where defects are expected to be prevalent it is imperative that a high degree of fault and defect tolerance is achieved. Illustrated in Figure 1 is a demultiplexer laid out in a crossbar configuration which controls a 4×4 crossbar nanomemory. Two signals, A0 and A1, which drive four microwire (MW) (vertical wires) signal lines, are interfaced with the demultiplexer nanowire (NW) address-lines via the placement of bistable molecules at their intersection; the molecular switches are represented by the resistor junctions. The output address-lines can be viewed as having an AND gate functionality. Hence, a nanowire address-line can only be selected if its two input signals are high or “1”.

III. ECC ENHANCED MULTI-SWITCH JUNCTION DEMULTIPLEXER ARCHITECTURE

In this work we utilize a Multi-Switch Junction (MSJ) fault tolerant scheme [3, 4] which uses redundancy in the nanowires of crossbar nanomemories and demultiplexers. The microwire CMOS interfaces have a less than 1% probability of being defective, and as such do not justify incurring the large area penalties associated in making them redundant.

The Error Correction Code (ECC) control strategy employed in this work builds upon, and is similar to that introduced by Kuekes et al [5]. Error correction codes are limited in the number of defects they can tolerate—they are only able to tolerate a minimum of $(d_{\min} - 1)$ defects, where d_{\min} is the Hamming distance of a linear coding block, and the address field matrix of the demultiplexer is represented as a linear block code. This is also true of the MSJ scheme which requires at least one of its redundant NW address-lines to be fully functional to achieve defect-tolerance. We have demonstrated in this work that by incorporating the MSJ scheme with an efficient ECC demultiplexer, the overall reliability of the demultiplexer can be further enhanced.

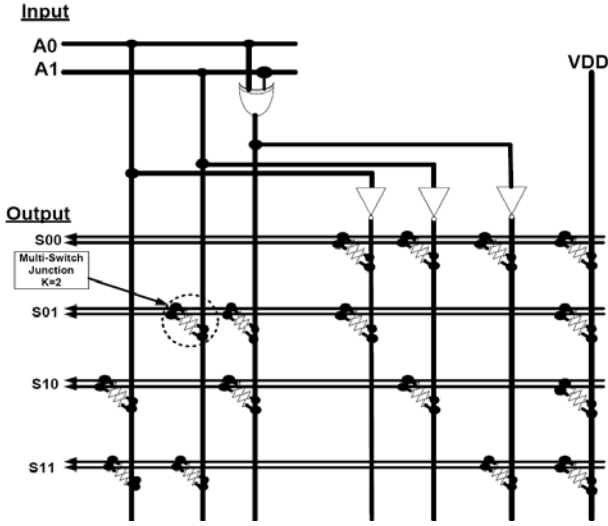


Figure 1. A molecular RAM demultiplexer with an additional EX-OR gate and inverter for ECC implementation. An additional NW is added to each row to implement the multi-switch junction scheme.

A. ERROR CORRECTION CODE (ECC) DEMULTIPLEXER ARCHITECTURE

Given the single direction signal flow from the input to output electrodes of the demultiplexer; the forward error correction strategy for one-way systems is used [6]. The address field matrix of the demultiplexer is represented as a linear block code. For any input signal length $m \geq 3$, there exists a Hamming code with parameters as follows; Code length: $n = 2^m - 1$, input signal binary code length: $k = 2^m - m - 1$, and number of added parity bits: $m = n - k$. The minimum Hamming distance d_{\min} of the linear coding block is the minimum number of bits by which the address lines differ. A one-to-one ECC parity scheme which can tolerate up to $(d_{\min} - 1)$ fault, is implemented and simulated in this work.

The reliability analysis is conducted by computing the probability of a NW functioning given a certain probability of a stuck-open fault occurring. When stuck-open faults occur on a NW causing it to become defective or fail, the NW is classified as a villain NW. Defective NWs can then propagate their errors to cause another NW to fail, the affected NW can be labeled a victim NW. The probability of obtaining functioning NWs can then be computed as the probability of a villain NW and a Victim NW not occurring in the demultiplexer. To calculate the ECC-MSJ probability, the well thought out derivation of Kuekes et al utilized is utilized and extended to include the effects of the MSJ on the reliability of the demultiplexer.

To compute the error correction probability of a linear code, the weight of the linear code must first be established. The weight of a linear code comprised of binary vectors is the number of ones contained in the code. Given any set S of binary vectors of length n , its weight profile function $W_S(i)$ can be defined, which describes for any integer $i \in [0, n]$, the number of elements of the set S with weight i . Let the set S be the code U with accompanying weight profile $W_U(i)$ —denotes

the number of codewords in U at each Hamming weight i . We can now introduce a set of dominating n -bit binary vectors $T(U)$ for code U , defined as $T(U) = \{e \mid e \text{ dominates } y \text{ for some non-zero } y \in C\}$, where the expression “ e dominates y ” denotes the fact that e has a 1 at every bit position for which y has a 1. The weight profile of $T(U)$ is given by $W_{T(U)}(i)$. A conflict occurs between two NW addresses when there are defective junctions, such as stuck-open defects, which manifest themselves as 0’s in the afflicted NW addresses, and can lead to some NW addresses becoming indistinguishable.

First we can calculate the probability of an output line being functional due to its own defects. Assuming defects to be statistically independent, $p_{villain}$ can be computed as follows.

$$p_{villain} = \sum_{i=1}^n W_{T(U)}(i) p^i (1-p)^{(n-i)} \quad (1)$$

Let the probability of an output line ceasing to be functional or usable be defined as p_{victim} . This output line is described as the *victim* because it does not lose its functionality due to its own defects, but because of interference from another defective output line. The probability p_{victim} can be calculated using the following equation.

$$p_{victim} = 1 - \prod_{i=1}^n (1-p^i)^{W_U(i)} \quad (2)$$

Where p is the probability of the redundant molecular switches not working—attributed to the implementation of the MSJ scheme. The probability p is computed using PRISM. When U is the identity code then $W_U(i)$ is the binomial coefficient $W_U(i) = \binom{n}{i}$. The probability of obtaining a functional output line can be calculated as;

$$p_{functional} = (1 - p_{villain}) \cdot (1 - p_{victim}) \quad (3)$$

Faults which occur at nanoscale dimensions are nondeterministic, to account for this fact, the probabilistic model checker PRISM [7, 8] was utilized in the computational probabilistic analysis model used to measure the reliability of the demultiplexers. The reliability analysis results presented in Table 1, compares the probability of obtaining a functional demultiplexer nanowire address-line against the probability of the occurrence of stuck-open molecular switch junction faults. Error rates cannot be too high if nanoelectronic devices are to be practical. As a result, the defects built into our simulation models were conservatively estimated to be in the 10% - 20% range. At this range a fault tolerant scheme which implements a combination MSJ-ECC scheme with redundancy $k = 3$ and Hamming distance $d = 2$ could, for

example, be used to obtain approximately 99% reliability in the demultiplexer, as is indicated in Table 1.

$$\frac{R}{L} = \frac{\rho}{A} \quad (7)$$

Array Size	128×128	128×128	128×128	128×128
Probability P _{fail} = 0.2	Demux d = 1	Demux d = 2	Demux d = 3	Demux d = 4
Redundancy	Reliability	Reliability	Reliability	Reliability
k=1	1.32%	14.32%	76.05%	88.89%
k=2	54.71%	92.05%	99.82%	99.98%
k=3	94.22%	99.62%	100%	100%
k=4	97.74%	99.99%	100%	100%
k=5	99.52%	100%	100%	100%
k=6	99.92%	100%	100%	100%

Table 1. Results showing the improvements in demultiplexer (Demux) reliability with increasing redundancy (k), and increasing Hamming distance (d).

IV. DEMULTIPLEX CIRCUIT MODEL AND ANALYSIS

In previous work, a parameterized circuit model was utilized to analyze the performance of the crossbar nanomemory [4]. Here, a full scale model of the crossbar nanomemory demultiplexer is implemented; the motivating factor behind this approach was to achieve a more detailed understanding of the fault tolerance induced performance penalties incurred in the crossbar nanomemory demultiplexer devices. The circuit models were implemented and simulated in HSPICE. The bistable molecular switch junctions were modeled as resistors and the redundant NWs were modeled as interconnecting RLC transmission lines as described by Burke et al [9] and illustrated in Figure 2.

C_{ES} and C_Q are the electrostatic and quantum capacitance, given in Equations (4) and (5) respectively. Where v_f (m/s) is the Fermi velocity of the NW.

$$C_{ES} = \frac{LA\epsilon_0\epsilon_r}{\pi \ln(4d/w)} \quad (4) \quad C_Q = \frac{2e^2}{hv_f} \quad (5)$$

The capacitance of the MW was modeled as the parallel plate capacitance [Equation (6)] between the MW and the chip substrate as symbolized by the parameter d_{dist_sub} in Equation (6).

$$C_{MW} = \left(\frac{\epsilon h L}{d_{dist_sub}} \right) \quad (6)$$

The NWs were modeled as copper NWs, with resistance per unit length was computed using the resistivity of bulk copper material [3]—according ITRS specifications— and as described in Equation (7). The resistances of the demultiplexer MW address-lines were also computed using Equation (7), adjusted to meet the MW dimension parameters.

Only the kinetic inductance L_{k_NW} is used in the NW interconnect model because in one-dimensional systems kinetic inductance always dominates magnetic inductance [10]. Typical inductance values in CMOS on-chip environments have been found to be approximately 1nH/mm or less [11]. The inductance of the demultiplexer address-line MWs were modeled using Equation (8).

$$L_{k_NW} = \frac{h}{2e^2v_f} \quad (8)$$

The redundant NWs described in Figure 1 are connected to the same contact electrode, they are assumed to be densely spaced and driven by the same voltage source. Further, because the redundant NWs switch in tandem, the effective parasitic capacitance between them is negligible [12].

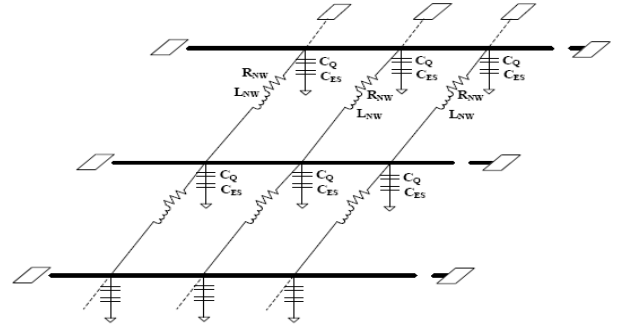


Figure 2. RLC network schematic for crossbar nanomemory.

A. SIMULATION SETUP

An HSPICE nanomemory simulator was designed and constructed to conduct the delay and power analysis. Delay is defined in this work as the time required to access and read stored information from a selected bit. A worst case analysis was performed (i.e. the target bit assessed was the bit farthest from the voltage driver source and the sense amplifiers). All bits on the assessed word-line and bit line were set to their low resistive “ON” state, while the rest of the non-accessed bits were reverse biased. The NW contact electrodes were modeled as voltage sources and the sense amplifiers as load resistors. Model parameters are provided in Table 2.

Nanowire width [3]	15nm
Nanowire pitch	33nm
Microwire width	90nm
Microwire pitch	180nm
“ON” Resistance (R_{ON}) [1]	0.48M Ω
“OFF” Resistance (R_{OFF})	9.2M Ω
Load Resistance (R_{load}) [8]	$\sqrt{(R_{ON})(R_{OFF})}$

Table 2. Model parameters

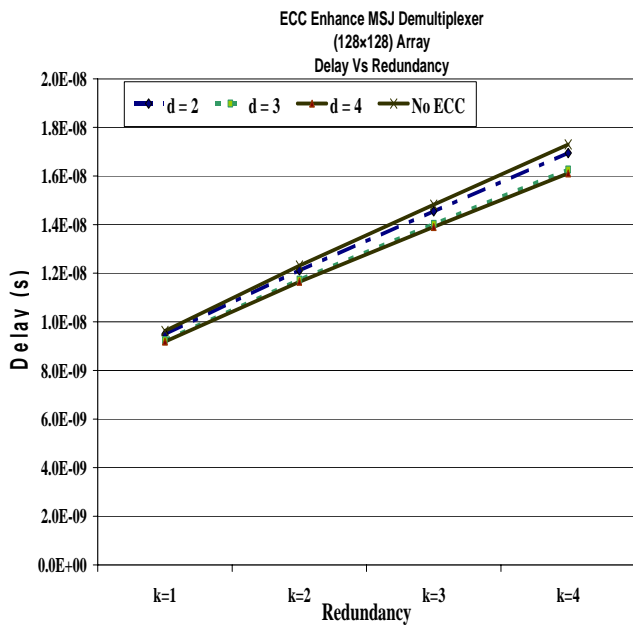


Figure 3. Graph showing the relative increase in access time delay with increasing redundancy (k) and increasing error correction code parity bits, in the crossbar demultiplexer.

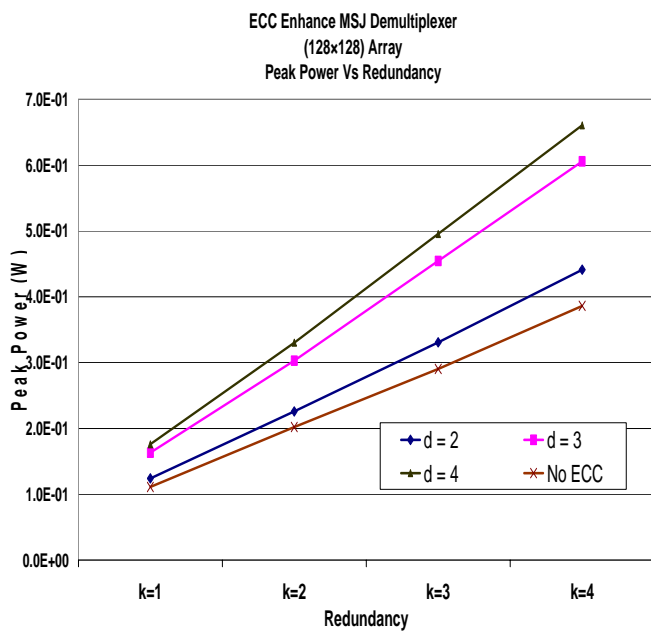


Figure 4. Graph showing the relative increase in peak power dissipated with increasing redundancy (k) and increasing error correction code parity bits, in the crossbar demultiplexer.

The results of the access time delay analysis simulation, as presented in Figure 3, can be interpreted as showing the cost with respect to access time delay, of increasing redundancy and Hamming distance. In Figure 3, delay improves with increasing Hamming distances due to the additional conduction path introduced by the CMOS ECC circuitry. In contrast, for the power analysis case presented in the plot of Figure 4, the addition of peripheral CMOS circuitry increases

the reliability induced power dissipation penalty. Also, the difference in the power dissipation penalties paid, due to the implementation of ECC codes with $d = 3$ and $d = 4$ are marginally small as a result of the single bit difference required for their implementation. For example in the ($k = 4$) case there is a 56.91% and 71.09% difference as calculated from the baseline case (No (ECC-MSJ) implemented), when the Hamming distance $d = 3$ and $d = 4$ codes respectively, are implemented, compared to a 14.25% difference in peak power dissipated when the ($d = 2$) ECC code is implemented.

V. CONCLUSION

In this work the reliability of an ECC enhanced MSJ crossbar nanomemory demultiplexer, along with its delay and power performance analysis has been presented. Results show an ECC-MSJ combination further improves the reliability of the crossbar nanomemory demultiplexer.

Results also indicate an increase in dissipated power as a consequence of the ECC-MSJ combination. Implementing ECC requires the inclusion of more microwires into the demultiplexer circuitry, which translates to an increase in power dissipation. It can thus be concluded that an efficient strategy for combining the MSJ and ECC schemes would be to first optimize the MSJ implementation before further enhancing the demultiplexer reliability via the incorporating of the ECC scheme.

REFERENCES

- [1] Y. Chen, G. Jung, D. Ohlberg, X. Li, D. Stewart, J. Jeppesen, K. Nielsen, J. Stoddart, and R. S. Williams, "Nanoscale molecular-switch crossbar circuits," *Nanotechnology*, vol. 14, pp. 462, 2003.
- [2] J. E. Green, J. Wook Choi, A. Boukai, Y. Bunimovich, E. Johnston-Halperin, E. DeIonno, Y. Luo, B. A. Sheriff, K. Xu, Y. Shik Shin, H-R. Tseng, J. Fraser Stoddart and J. R. Heath, "A 160-kilobit molecular electronic memory patterned at 10^{11} bits per square centimetre," *Nature*, vol. 445, pp. 414-417, 2007.
- [3] A. Coker, V. Taylor, D. Bhaduri, S. Shukla, A. Raychowdhury, and K. Roy, "Multi-junction fault-tolerance Architecture for Nanoscale Crossbar Memories," *IEEE Transactions on Nanotechnology*, vol. 7, no. 2, pp. 202-208, March 2008.
- [4] A. Coker, V. Taylor, "Performance and Reliability Analysis of a Scaled Multi-Switch Junction Crossbar Nanomemory and Demultiplexer", *Proceedings of 7th IEEE Conference on Nanotechnology*, Aug 2007.
- [5] Philip J. Kuekes, Warren Robinett and R. Stanley Williams, "Improved voltage margins using linear error-correction codes in resistor-logic demultiplexers for nanoelectronics." *IOP Publishing, Nanotechnology*, vol. 16, pp. 1419-1432, 2005.
- [6] S. Lin, D. Costello, *Essentials of Error-Control Coding techniques*, 2nd Edition, Prentice-Hall, Inc. Upper Saddle River, NJ, 2004
- [7] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 2.0: A tool for Probabilistic model checking," *QEST'04*, IEEE Computer Society Press, pp. 322-323, 2004.
- [8] Web Page: www.cs.bham.ac.uk/~dxdp/prism/.
- [9] P.J. Burke, "Luttinger liquid theory as a model of the gigahertz electrical properties of carbon Nanotubes," *IEEE Transactions on Nanotech*, vol. 1, pp. 129-144, 2002.
- [10] P.J. Burke, "AC performance of nanoelectronics : towards a ballistic THz nanotube transistor" *Solid-State Electronics* 48 (2004) 1981-1986
- [11] R. Kapur, J.P. McVittie and K.C. Saraswat "Technology and Reliability Constrained Future Copper Interconnects—Part II—Performance Implications," *IEEE Trans Electron Devices*, vol. 49, no. 4, April 2002
- [12] K. Hirose, H. Yasuura, "A bus delay reduction technique considering crosstalk," *Proceedings of DATE*, pp.441-445, March 2000, France.