

# Multiplexing Schemes for Cost-Effective Fault-Tolerance

Sandip Roy and Valeriu Beiu

School of EE&CS, Washington State University, Pullman, WA 99164-2752

**Abstract** — Motivated by the need for cost-effective fault-tolerant nano architectures, we explore von Neumann multiplexing (vN-MUX) at small and very small redundancy factors. We present a novel analysis of vN-MUX of 3-input majority gates (MAJ-3), using combinatorial arguments to exactly determine performance. We show that MAJ-3 vN-MUX performs very well when compared to other redundancy schemes, increasing the allowed device error probability by four orders of magnitude (for small redundancy factors). We describe in detail an extension, called MAJ-3 vN-MUX( $N,k$ ), that contributes up to four more orders of magnitude, by excluding superfluous restorative stages for very small redundancy factors. We also analytically determine the performance of MAJ-3 vN-MUX for large redundancy factors, finding that the maximum tolerable gate failure probability is 0.0197 (in contrast to 0.0107 for NAND-2 vN-MUX).

**Index Terms** — Architecture, fault-tolerance, majority logic circuits, multiplexing.

## I. INTRODUCTION

The development of nanodevices brings promise for improvement in performance, yet it also leads to new challenges, including the need for architectures that reduce the uncertainty inherent to nanocomputations. One well-known approach for developing fault-tolerant architectures in the face of uncertainties (both *defects* and transient *faults*) is to incorporate spatial and/or temporal redundancy. While redundancy is needed, cost-effectiveness constraints dictate that *redundancy factors* — the number of times a computing unit is replicated — must be small, or better very small.

Also, recent studies have highlighted the benefits of nanoscale design using threshold logic gates (TLGs) [1], [2], [3], [4], and so redundant design schemes must be adapted for TLG circuits. With these motivations in mind, we consider *redundant design of TLG circuits in the case of small and very small redundancy factors*. We focus in particular on a redundant design scheme known as von Neumann multiplexing.

Reliable operation of a circuit can be achieved using redundancy at different levels: device, gate, block, in time, and in communications [3], [5]. While most of these methods are beyond our scope, we note that improved reliability is traded off for chip area and connectivity. In [6] von Neumann introduced the multiplexing redundancy algorithm (vN-MUX) as a plausible representation for

reliable computation. vN-MUX was developed for arbitrary gates, including majority (MAJ) and NAND gates (see Fig. 1). However, a reliability analysis was performed for NAND-2 (where 2 is the fan-in) gates only, assuming independent gate failures and large redundancy factors. The performance of NAND-2 vN-MUX was compared with other fault-tolerant techniques in [5], and the technique was analyzed at ‘small’ redundancy factors (30 to 3000) in [7]. In [8] Reischuk investigated the required redundancy for circuits with unbounded fan-in, concluding that TLG circuits have a clear advantage over Boolean gate circuits: *only* using TLGs can arbitrary circuits be made reliable with a small amount of additional hardware.

The vN-MUX algorithm [6] aims to improve the reliability of a sequence of computations. This ‘multiplexing’ of each computation serves to contain error propagation down the sequence, by selecting the more-likely result at each stage. *vN-MUX can be applied to any logic, but the analysis must be adapted in each case.*

## II. SUMMARY OF OUR RECENT RESULTS

We summarize our analysis of MAJ-3 vN-MUX (first presented in [9]). A MAJ-3 vN-MUX comprises an *executive stage* and a *restorative stage* (see Fig. 2). The executive stage repeats the computation  $N$  times, operating on  $N$  different sets of inputs. The restorative stage triplicates and randomly orders the outputs from the executive stage, and chooses the MAJ-3 of each set of three signals to generate the  $N$  final outputs.

We assume that each of the three sets of  $N$  inputs—or bundle—has a nominal binary value. The number of lines in bundle  $i$ ,  $1 \leq i \leq 3$ , that are in error is  $e_i$ . Given  $e_i$ , all configurations of line errors among the  $N$  lines in bundle  $i$  are assumed to be equally probable. The random variables  $e_1$ ,  $e_2$ , and  $e_3$  are assumed to be independently distributed, with probability distributions  $P(e_1)$ ,  $P(e_2)$ , and  $P(e_3)$ . We also assume that each MAJ-3 can fail with probability  $q_{\text{MAJ-3}}$ . We denote the number of final outputs that differ from the nominal output by  $d$ . We define a variable  $w$  for the number of MAJ-3 (in the executive stage) that have at least two inputs *high*. Based on counting arguments, we find the conditional distribution for  $w$  given  $e_1$  and  $e_2$ :

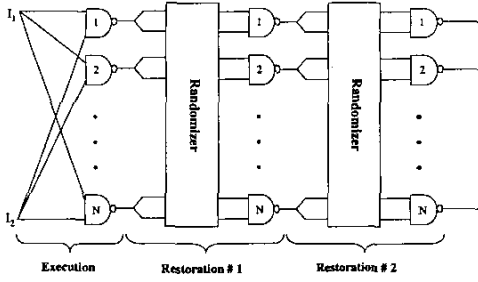


Fig. 1. NAND-2 von Neumann multiplexing.

$$P(w|e_1, e_2) = \frac{\binom{N}{w} \binom{N-w}{e_1} \binom{e_1}{N-e_2-w}}{\binom{N}{e_1} \binom{N}{e_2}}, \quad (1)$$

and compute  $P(w) = \sum_{e_1=1}^N \sum_{e_2=1}^N P(w|e_1, e_2) \cdot P(e_1) \cdot P(e_2)$ .

Further we compute the distribution for  $s$  — the number of correct outputs (i.e., the number of outputs that are *high*) after the executive stage. The conditional distribution for  $s$  given  $w$  is binomial

$$P(s|w) = \binom{w}{s} \cdot (1-q)^s \cdot q^{w-s} \quad \text{and the (unconditioned)}$$

distribution for  $s$  is  $P(s) = \sum_{w=s}^N P(s|w) \cdot P(w)$ .

We next consider  $a$ , the number of MAJ-3 in the restorative stage that have at least two inputs *high*. We compute the conditional distribution for  $a$  given  $s$ , using a combinatorial argument:

$$P(a|s) = \frac{\sum_{i_3=0}^{\min(a,s)} \binom{N}{i_3} \binom{N-i_3}{a-i_3} \binom{N-a}{3s-3i_3-2(a-i_3)} 3^{a-i_3} 3^{3s-3i_3-2(a-i_3)}}{\binom{3N}{3s}}. \quad (2)$$

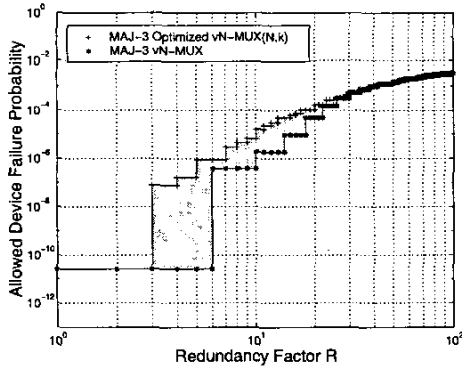


Fig. 3. Maximum allowed failure probability for MAJ-3 vN-MUX, and the optimized MAJ-3 vN-MUX( $N, k$ ).

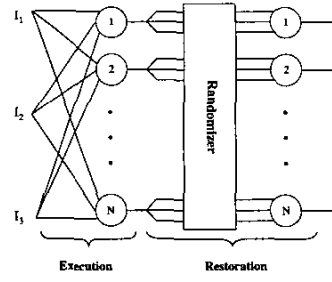


Fig. 2. One MAJ-3 von Neumann multiplexing stage.

Finally, we compute the distribution of the number of output errors  $d$  from the distribution of  $a$ , by considering failures in the restorative stage. A binomial distribution represents the conditional distribution (this analysis is identical to that of the MAJ-3 executive stage).

### III. CHIP-LEVEL ANALYSIS

We extend the analysis to the chip level, by determining the relationship between the device failure rate ( $p_f$ ) and the chip failure rate. Our analysis follows those of [5], [7]. We consider a chip with  $N_T = 10^{12}$  devices and  $N_u = 10^6$  effective devices per processing unit, each of which returns a single bit. Since the MAJ-3 vN-MUX introduces a redundancy factor of  $R = 2 \cdot N$  (see Fig. 2), the number of processing units is  $n = N_T / [(2 \cdot N) \cdot N_u]$ . We assume that the units have a logical depth of 10, and that the decision threshold level is  $\theta = 0.5$ . We calculate the probability that  $d/N \geq \theta = 0.5$ , following the method of [5]. This represents the reliability of one data bit ( $P_{bit}$ ). We note that the implementation of the threshold at the outputs requires additional circuitry, which we assume to have negligible failure probability (as in [6], [5]). Highly reliable gates can be designed by introducing redundancy and noise immunity at the device level (see [10], [11]).

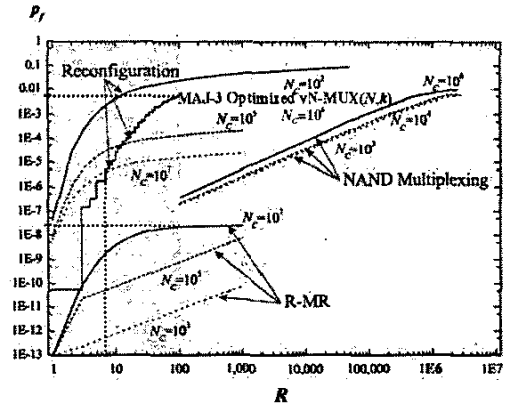


Fig. 4. Comparison of the enhanced MAJ-3 vN-MUX( $N, k$ ) with other redundancy schemes (for  $N_c = 10^6$ ).

Through simulation, we determine the maximum failure probability needed to guarantee a chip reliability of 90%, as a function of  $R$ . Our results are shown in Fig. 3, and compared to the results of [5] in Fig. 4. They are very promising.

- The allowable probabilities of error are significantly larger for MAJ-3 vN-MUX than for the other redundancy schemes that can tolerate soft/transient errors: R-modular redundancy (R-MR), and NAND-2 vN-MUX. In comparison to R-MR, the improvement is by up to five orders of magnitude (see Fig. 4 for a redundancy factor of 100).
- The allowable probabilities of error are comparable, or even better, than those achieved by reconfiguration (which can easily tolerate only hard/permanent errors) starting from very small redundancy factors  $R \geq 7$  (see again Fig. 4).

We have only recently become aware of the new work [12], which proposes a framework for computing reliability that takes into account the time-durability of the chip. Our preliminary explorations suggest that MAJ-3 vN-MUX performs very well in this framework also.

While our focus has been on performance at small and very small redundancy factors, it is worthwhile to mention that MAJ-3 vN-MUX also shows promise at large redundancy factors. Our simulations suggest that, in the limit of large redundancy  $R$ , the (worst-case) maximum tolerable gate failure probability approaches a steady-state value. We have determined this limiting probability analytically, by exploiting the Law of Large Numbers (see, e.g., [13]). For MAJ-3 vN-MUX the limiting probability is 0.0197. We note here that this limiting probability for MAJ-3 vN-MUX compares favorably with the maximum gate failure probability of 0.0107 for NAND-2 vN-MUX [6]. We also note here that this limiting probability is valuable for the efficient choice of simulation parameters, such as for the simulations in [14].

#### IV. AN ENHANCED MULTIPLEXING SCHEME

The purpose of the restorative stage in vN-MUX is to reduce error propagation from a logic computation's input to its output, by selecting for the more common outputs from the computation. The restorative stage is only effective when the probabilities of error in the inputs are sufficiently large. In fact, for small input error probabilities, the chance of error introduced by the second set of MAJ-3 gates might outweigh the advantage of the restorative stage. Thus, if the input error probabilities for a particular logic computation are small enough, we can simultaneously improve the output error probability and

economize (reduce the redundancy) of our MAJ-3 vN-MUX design by eliminating the restorative stage.

From another viewpoint, if we are seeking the best (i.e., smallest) architecture for a particular redundancy factor  $R$ , we might expect to improve on the standard vN-MUX algorithm by applying the restorative stage on only some computations, while simultaneously increasing the bundle size  $N$  (i.e., the number of parallel computations). For instance, say that we wish to design a vN-MUX architecture with a redundancy factor  $R \leq 15$ . A classical NAND-2 vN-MUX would use  $N = 5$  and apply restoration on every computation, hence  $R_{\text{NAND-2}} = 3 \times 5 = 15$ . If we were to apply the standard MAJ-3 vN-MUX, we would use a bundle size of  $N = 7$  and apply restoration on every computation, which yields  $R_{\text{MAJ-3}} = 2 \times 7 = 14$ . It turns out that the reliability of this design can be improved even further by using a bundle size of  $N = 11$  and applying restoration only every third computation (to be precise, for computations at depths 3, 6, and 9). This design yields an 'average' redundancy of  $R_{\text{OPT MAJ-3}} = 11 + 11/3 = 14.3$ . We use this principle to improve our MAJ-3 vN-MUX, *though the same principle can be applied when using any other type of gate.*

Specifically, we consider architectures in which the logical depths of all inputs to a given computation are the same. We define the new *vN-MUX( $N, k$ ) architecture* as one in which an executive stage with bundle size  $N$  is used for all computations, and a restorative stage is applied only on every  $k$ th stage (i.e., for computations with logical depth  $k, 2k, \dots$ ). The redundancy factor introduced by a vN-MUX( $N, k$ ) architecture is  $R = N + N/k$ . By applying the restorative stage on only every  $k$ th stage, the bundle size  $N$  can be increased to  $N^+ = \lceil 2k/(k+1) \rceil N$  for the same redundancy factor  $R$ .

We now consider all vN-MUX( $N, k$ ) architectures, i.e., all possible ( $N, k$ ) pairs achieving a given  $R$ . We find this probability by computing the maximum allowed probability of device failure for each architecture with sufficiently small redundancy (i.e., below the given  $R$ ), and then choosing the architecture that maximizes the allowed probability. For instance, for a redundancy factor  $R = 15$ , we find that MAJ-3 vN-MUX( $11, 3$ ) architecture provides the maximum allowed probability of device failure, of  $6.25 \times 10^{-5}$ .

The vN-MUX( $N, k$ ) is compared with other redundancy schemes in Fig. 4. It provides a significant improvement over standard MAJ-3 vN-MUX at very small redundancy factors  $R$ , i.e., those below ten (see Fig. 3). Additionally, Table I compares the standard and enhanced redundancy schemes, for  $R$  between three and ten. The table highlights that the enhanced MAJ-3 vN-MUX( $N, k$ ) scheme can improve upon the standard MAJ-3 vN-MUX scheme by a factor of as much as 36,000.

TABLE I  
MAJ-3 vN-MUX AND OPTIMIZED MAJ-3 vN-MUX( $N, k$ ) FOR VERY SMALL REDUNDANCY FACTORS  $R = 3 \dots 10$

Redundancy Factor	Maximum Allowed Device Failure Probability		Improvement Factor (due to optimization)	$(N, k)$ Pair (optimized design)
	MAJ-3 vN-MUX	Optimized MAJ-3 vN-MUX( $N, k$ )		
3	$2.5 \times 10^{-11}$	$7.8 \times 10^{-8}$	3100	(3, $\infty$ )
4	$2.5 \times 10^{-11}$	$1.7 \times 10^{-7}$	6700	(3, 3)
5	$2.5 \times 10^{-11}$	$9.0 \times 10^{-7}$	36,000	(5, $\infty$ )
6	$3.8 \times 10^{-7}$	$9.0 \times 10^{-7}$	2.4	(5, $\infty$ )
7	$3.8 \times 10^{-7}$	$3.0 \times 10^{-6}$	8	(7, $\infty$ )
8	$3.8 \times 10^{-7}$	$4.5 \times 10^{-6}$	12	(7, 10)
9	$3.8 \times 10^{-7}$	$6.5 \times 10^{-6}$	17.3	(7, 4)
10	$1.9 \times 10^{-6}$	$1.6 \times 10^{-5}$	8.4	(9, 10)

## V. CONCLUSIONS

We have developed an exact analysis for MAJ-3 vN-MUX, and compared it to other redundancy schemes. These show that MAJ-3 vN-MUX holds great promise for cost-effective (i.e., for small redundancy factors) fault-tolerant nanoarchitectures. We have also presented an enhancement that can boost the performance even more, specifically at very small (i.e., practical) redundancy factors. We believe that MAJ- $\Delta$  vN-MUX( $N, k$ ), can still be improved (here  $\Delta$  is the fan-in of the gate). Research is underway on MAJ-5 vN-MUX, and also for combining vN-MUX with redundancy schemes at the transistor (device) level [10], [11].

## REFERENCES

- [1] W. M. Kistler, W. Gerstner, and J. L. van Hemmen, "Reduction of the Hodgkin-Huxley equations to a single-variable threshold model," *Neural Computation*, vol. 9, Jul. 1997, pp. 1015–1045.
- [2] V. Beiu, "A survey of perceptron circuit complexity results," *Proc. Intl. Joint Conf. Neural Networks IJCNN'03*, Portland, OR, USA, Jul. 20–24, 2003, vol. 2, pp. 989–994.
- [3] R. Compañó, L. Molenkamp, and D. J. Paul (Eds.), *Technology Roadmap for Nanoelectronics*, European Commission, Future Emerging Technologies, 2000. Available: <http://www.cordis.lu/esprit/src/melna-rm.htm>
- [4] V. Beiu, J. M. Quintana, and M. J. Avedillo, "VLSI implementation of threshold logic: A comprehensive survey," *IEEE Trans. Neural Networks*, vol. 14, 2003, pp. 1217–1243.
- [5] K. Nikolić, A. S. Sadek, and M. Forshaw, "Fault-tolerant techniques for nanocomputers," *Nanotechnology*, vol. 13, 2002, pp. 357–362. Earlier version as: M. Forshaw, K. Nikolić, and A. S. Sadek, "ANSWERS: Autonomous Nanoelectronic Systems With Extended Replication and Signaling," *MEL-ARI #28667*. Available [http://ipga.phys.ucl.ac.uk/research/answers/reports/3rd\\_year\\_UCL.pdf](http://ipga.phys.ucl.ac.uk/research/answers/reports/3rd_year_UCL.pdf). Preliminary version in *Proc. IEEE Conf. Nanotech. IEEE-NANO'01*, Maui, HI, USA, Oct. 28–30, 2001, pp. 254–259.
- [6] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in C. E. Shannon, and J. McCarthy (eds.), *Automata Studies*, Princeton: Princeton Univ. Press, 1956, pp. 43–98.
- [7] J. Han, and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," *IEEE Trans. Nanotech.*, vol. 1, Dec. 2002, pp. 201–208.
- [8] R. Reischuk, "Can large fanin circuits perform reliable computations in the presence of faults?," *Theoretical Comp. Sci.*, vol. 240, no. 12, Jun. 2000, pp. 319–335.
- [9] S. Roy, V. Beiu, and M. Sulieman, "Reliability Analysis of Some Nano Architectures," *Neural Information Processing Systems NIPS'03, Special Workshop*, Whistler, Canada, Dec. 12–13, 2003. Available: [http://www.eecs.wsu.edu/~vbeiu/workshop\\_nips03/Presentations/S\\_Roy.pdf](http://www.eecs.wsu.edu/~vbeiu/workshop_nips03/Presentations/S_Roy.pdf)
- [10] A. Schmid, and Y. Leblebici, "Robust circuit and system design methodologies for nanometer-scale devices and single-electron transistors," *Proc. IEEE Conf. Nanotech. IEEE-NANO'03*, San Francisco, CA, USA, Aug. 12–14, 2003, vol. 2, pp. 516–519.
- [11] S. Tatapudi, and V. Beiu, "Split-precharge differential noise immune threshold logic gate," in J. Mira, and J.R. Álvarez (Eds.) *Artificial Neural Nets Problem Solving Methods (IWANN'03)*, Menorca, Spain, Jun. 3–6, Springer, LNCS 2687, 2003, pp. 49–56.
- [12] A. S. Sadek, K. Nikolić, and M. Forshaw, "Parallel information and computation with restitution for noise-tolerant nanoscale logic networks," *Nanotechnology*, vol. 15, 2004, pp. 192–210.
- [13] S. M. Ross, *A First Course in Probability*, Prentice Hall: Upper Saddle River, NJ, 2002.
- [14] D. Bhaduri, and S. K. Shukla, "Tools and techniques for evaluating reliability of defect-tolerant nano architectures," *Proc. Intl. Joint Conf. Neural Networks IJCNN'04*, Budapest, Hungary, Jul. 25–29, 2004, in press.
- [15] M. Sulieman, and V. Beiu, "Design and analysis of SET circuits: Using MATLAB modules and SIMON" *Proc. IEEE Conf. Nanotech. IEEE-NANO'04*, Munich, Germany, Aug. 17–19, 2004 (in this proceedings).