

An Approximate Voting Scheme for Reliable Computing

Ke Chen and Fabrizio Lombardi

Electrical and Computer Engineering
Northeastern University
Boston, MA, USA

chen.ke1@husky.neu.edu, lombardi@ece.neu.edu

Jie Han

Electrical and Computer Engineering
University of Alberta
Edmonton, Canada T6G 2V4
jhan8@ualberta.ca

Abstract— This paper relies on the principles of inexact computing to alleviate the issues arising in static masking by voting for reliable computing. A scheme that utilizes approximate voting is proposed; it is referred to as inexact double modular redundancy (IDMR). IDMR does not resort to triplication, thus saving overhead due to modular replication; moreover, this scheme is adaptive in its operation, i.e., it allows a threshold to determine the validity of the module outputs. IDMR operates by initially establishing the difference between the values of the outputs of the two modules; only if the difference is below a preset threshold, then the voter calculates the average value of the two module outputs. An extensive analysis of the voting circuits and an application to image processing are presented.

Keywords—Approximate computing, Reliable system, Modular Redundancy, Voting

I. INTRODUCTION

Soft errors have become a major concern in the design of nanoscale digital integrated circuits [1]. A soft error may occur due to a strike by a high-energy particle and manifests itself as a transient bit reversal in the logic value of a circuit node. Redundancy techniques are effective to address soft errors; they are commonly used for designing dependable systems to ensure high reliability and availability [2] [3]. One of the most effective fault-tolerant design schemes is the so-called N-modular redundancy (NMR); in a NMR (TMR) scheme, N (3) copies of a module are utilized [4]. A well-known alternative to TMR is double modular redundancy (DMR), i.e., the original module is duplicated. This scheme reduces the cost of redundancy by providing error detection; however, error correction is not always possible, because comparison cannot always identify the erroneous module and therefore, additional circuitry is needed.

In general, redundancy approaches are best applicable provided failure independence is retained in the operations of the modules [5]. Design diversity is needed to avoid the so-called common mode failure (CMF), e.g. to employ different redundant implementations of the original module. Thus, when the CMF occurs, the modules can produce different outputs and the error can be detected; however, different implementations may cause small differences among the module values as outputs, thus resulting in the failure of a

voting scheme such as TMR. This is caused by the strict requirement in finding the majority from the voter inputs when even slightly different values are provided. This property is often referred to as *static masking* and is one of the major disadvantages of a redundancy scheme [5].

Many applications such as multimedia and image processing can tolerate errors and imprecision in computation and still produce meaningful and useful results [6]. The paradigm of *inexact or approximate computing* relies on relaxing fully precise and completely deterministic building modules when for example, designing energy-efficient systems. This allows imprecise computation to redirect the existing design process of digital circuits and systems by taking advantage of a decrease in complexity and cost with possibly a potential increase in performance and power efficiency [7].

This paper relies on the principles of inexact computing to alleviate the issues arising in static masking by voting. A scheme that utilizes approximate voting is proposed. This scheme is referred to as inexact double modular redundancy (IDMR). IDMR does not resort to triplication, thus saving overhead due to modular replication; this scheme is adaptive in its operation, i.e., it allows a threshold to determine the validity of the module outputs. IDMR operates by initially establishing the difference between the values of the outputs of the two modules; only if the difference is below a preset threshold, then the voter calculates the average value of the two module outputs.

II. REVIEW

DMR Scheme: In DMR, the outputs of two modules are compared; an error is detected if the outputs differ, therefore, a traditional DMR does not provide error correction [8]. Recently, the use of design diversity within DMR has been investigated to provide low cost detection and correction of radiation-induced soft errors [9]. The principle of this approach is that the two modules are implemented using designs that provide different output error patterns when a soft error hits. These error patterns can be detected as a series of mismatches between the module outputs; by recognizing these patterns, the module-in-error can be identified and the output from the other module is used as the final error-protected

output. This approach is application dependent, because error detection and correction require a dedicated unit that intelligently assesses the outputs of the two redundant modules.

Majority voting: Triple modular redundancy (TMR) uses three copies of the original module [4]. In a TMR, each module operates in a disjoint (independent) mode; so, the three modules compute in parallel. If a module produces an output that is different from the outputs of the other two modules, then the system output is established by voting. Voting assumes the majority of the modules to have the correct output; hence, the single disagreeing module (corresponding to the erroneous output) is masked by utilizing a voter. In a bit voter, majority voting is performed on a bit-by-bit basis. In bit-wise voting, the voter compares each bit; it then finds the majority of each bit to form the final output value. A novel word-voter has been proposed in [10]; in this scheme, voting considers the whole word, i.e., the word majority voter requires the output signals to be exactly the same when calculating the majority. However, this type of voter is not efficient when there are slight variations in the outputs of the modules (as often occurring in the nanoscales). In some cases, no voted output is generated, thus failing in providing a voted outcome.

III. PROPOSED INEXACT DMR (IDMR)

In this section, an inexact voting scheme is proposed. This scheme is referred to as inexact double modular redundancy (IDMR). The basic principle of the IDMR is to initially establish the difference between the outputs of two modules. If the difference is less than a preset threshold, the voter calculates the average value of the two module outputs as outcome. If the difference is larger than the threshold, the voter generates an error signal. The value of the threshold is dependent on the level of accuracy that is required as output in a reliable computing system. This scheme is adaptive in its operation, i.e., it allows a threshold to determine the validity of the module outputs. The averaging of the two module outputs ensures that variation in values is mediated by adjusting the final value as outcome. IDMR does not resort to triplication, thus saving in overhead due to the modular replication.

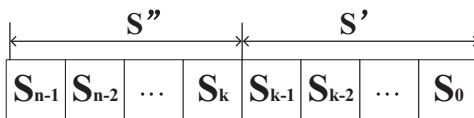


Fig. 1. Input data S.

Let the input (parallel) data word be denoted by S; this word is made of n bits. Let the subset of the lower k bits be denoted as S' , while the upper $n-k$ bits be given by the subset S'' , i.e., $S=(S'', S')$, as shown in Fig. 1. k is determined by the application using IDMR. The block diagram of the proposed IDMR scheme is shown in Fig. 2; IDMR consists of the following blocks, as discussed next.

Detector: The function of the detector block is to compare its two input signals corresponding to the two received outputs from the modules. An error signal is generated if the

difference of the two values (denoted as Input A and Input B in Fig. 2) is larger than the threshold. Else, the detector considers the two values to be valid and the following two scenarios are applicable. (1) When the upper $n-k$ bits of Input A and Input B are the same (i.e., $A''=B''$), the largest possible difference between them is 2^{k-1} . (2) If the absolute value of $A''-B''$ (i.e., $|A'' - B''|$) is 1, then the largest difference is $2^{k+1}-1$. Thus, the detector is designed by utilizing $n-k$ bit subtractors to find $A''-B''$.

The value of k is selected such that three scenarios are possible as validity conditions. (1) $A''-B''=0$: the borrow of the first bit is '0'. The difference of each bit is also '0'. (2) $A''-B''=1$: the borrow of the first bit is '0'. The difference of each bit is '0' except the last bit. (3) $A''-B''=-1$: the borrow of the first bit is '1'. The difference of each bit is also '1'.

Passing array: Following subtraction if there is no error signal, the input must be propagated for further processing. Thus, an array made of AND gates (referred to as the passing array) is needed; this array is controlled by the Enable signal. When the Enable signal is '1', the inputs are propagated; else, no propagation is allowed, thus saving power. If no error occurs, the output for the upper $n-k$ bits of the AND gates is given by $O''=A''=B''$, i.e., each of the upper $n-k$ bits of the output (denoted by Out) is equal to the corresponding Input A (or B) bit. So, $Out_n = Enable \cdot Input_n$.

Full adder: Let O' denote the average value of A' and B' . In the proposed design, full adders are used to calculate the sum of A' and B' . The average is found by shifting right the sum. However, the shift circuit is not necessary, because the first k bits (inclusive of the carry bit for $k-1$) can be used as result for this operation. For the last bit, only the carry bit needs to be considered; thus, an AND gate is used to replace the last full adder. The full adder in this paper is implemented as in [11].

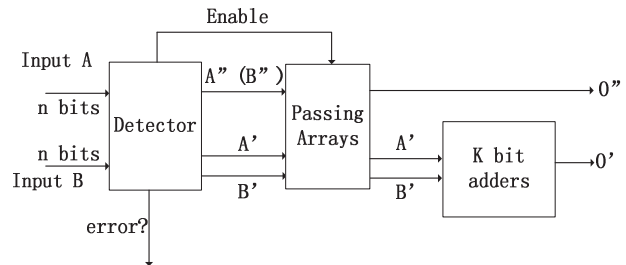


Fig. 2. The IDMR voting scheme.

IV. SIMULATION RESULTS

In the section, the designs of the proposed inexact voting scheme are evaluated; predictive technology models (PTMs) at 32nm feature sizes are used in HSPICE for the transistors.

Delay: The output delay is defined as the largest delay of each bit when no error is detected; so, the delay is the timing latency from inputs to the outputs of the voting hardware.

Table 1 shows the output delays for TMR and IDMR by varying k and n . The delay of the approximate scheme increases with n , but it decreases with k . TMR is the fastest scheme; it does not employ any adders (only a two-level logic network), so both delays are small (but increasing with n).

TABLE 1. IDMR AND TMR OUTPUT DELAY (PS)

	n=8	n=16	n=24	n=32
IDMR k=1	10.99	12.39	15.18	20.51
IDMR k=2	8.66	10.14	13.75	18.17
IDMR k=4	6.06	7.85	11.50	15.64
IDMR k=8	-	6.72	9.94	14.28
TMR	9.45	10.66	13.05	17.64

Power dissipation: Power dissipation has also been evaluated for IDMR and TMR. Tables 2 show the simulation results at different values of feature size, n and k , and TMR. As expected, the overhead is due to the larger number of modules required for TMR versus IDMR, i.e. 3 versus 2. TMR incurs in the largest power dissipation as reflected by the larger circuit complexity (analyzed next).

TABLE 2. IDMR AND TMR DYNAMIC POWER (μW)

	n=8	n=16	n=24	n=32
IDMR k=1	7.9	8.7	9.8	10.4
IDMR k=2	8.6	9.7	11	12.8
IDMR k=4	10.1	10.9	13.5	19.9
IDMR k=8	-	12.6	15.8	20.8
TMR	16.58	18.22	20.50	21.74

Circuit Complexity: Consider an input of n -bits from each of the modules to the voting hardware and a k -bits threshold for the approximate scheme. Table 3 shows the expressions for the circuit complexity of the proposed scheme as well as TMR. The proposed scheme incurs in a complexity smaller than TMR; the reduction is more pronounced at higher values of n . As linear with n , the circuit complexity of the proposed scheme slightly increases with the value of k for IDMR.

TABLE 3. VOTING CIRCUIT COMPLEXITY

	Circuit Complexity (transistor count)
TMR	$54n$
IDMR	$18n + 2k + 6$

Process variations: Next, variations in the MOSFETs of the proposed scheme are evaluated using Monte Carlo simulation. For Monte Carlo simulation, the process variations of a MOSFET consider the channel length and the threshold voltage. The variations in percentage for these parameters have been reported in [12]; as the most relevant metric in most high performance applications, the global behavior of the output delay is measured to evaluate the impact of these process variations.

The simulation results (Table 4) show that the threshold voltage has a more pronounced effect than the channel length in the operation of an approximate or exact voting system. The variation in output delay is mitigated by the approximate nature of the voting process in the proposed scheme. TMR has the largest percentage for variability, hence yet another negative feature that is caused by static masking.

TABLE 4. VARIABILITY PERCENTAGE ($3\sigma/\mu$) ON GLOBAL BEHAVIOR (OUTPUT DELAY)

	L	V_{th}
TMR	36.73%	47.18%
IDMR	28.97%	36.91%

V. APPLICATION: IMAGE PROCESSING

In this section, image processing is considered as an application of the proposed voting scheme. For analysis and ease of simulation, the system model is slightly changed to allow a controlled insertion of errors in module outputs. This also allows a better understanding of the underlying operations of the three modules in a redundant system, while still accounting for diversity in output values for voting.

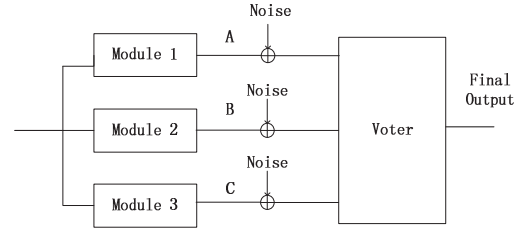
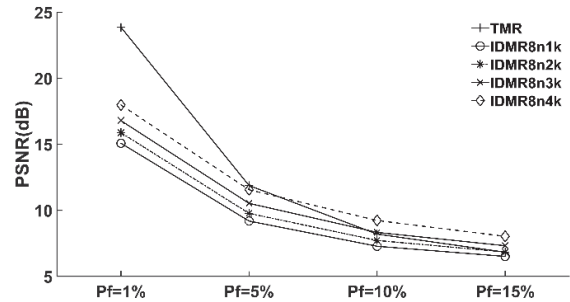


Fig. 3. Noise model of a voting scheme (TMR).

The block diagram of this model for a TMR voting scheme is shown in Fig. 3. A noise source is introduced at the outputs of each module prior to the voter. The noise sources are useful in introducing errors; if there is no error and noise, each module in a TMR generates the exact (correct) result. A similar model is used for IDMR.

Bit-wise noise: Bit-wise noise is defined as the noise affecting each bit of the inputs to the voting scheme with the same probability to flip (change) its value. The flip probability is denoted by P_f and it is assumed that each bit is independent. For simulation, a 0-1 uniform distribution random variable is generated; if the value of this random variable is less or equal to P_f , then this bit of the input word is changed.

Fig. 4. PSNR of different voting schemes (TMR and IDMR) with bit-wise noise vs P_f and variable k (for an 8-bit image).

The peak signal-to-noise ratio (PSNR) is established for the final output of the voter compared to the error-free result. Fig. 4 shows the simulation result for the PSNR of the voting schemes with the same bit-wise noise for an 8-bit image (i.e. $n=8$). From these results, the following conclusions can be drawn.

(1) For IDMR, if the value of k increases, the PSNR increases, i.e., the more inexact bits make the approximate voter to tolerate more noise. Thus, the probability of each module to generate a valid result increases too.

(2) TMR is better than IDMR at a low value of P_f .

Output noise: Suppose the correct value of the voter input is equal to q ; however, the module may produce a voter input that it is different from q (assume that the distribution of the output space follows a normal distribution with mean value q and variance σ).

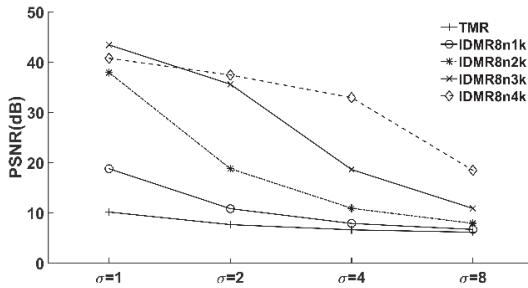


Fig. 5. PSNR of different voting schemes (TMR and IDMR) with output noise vs variance and variable k (for an 8-bit image).

Also in this case, the PSNR is established for an 8-bit image. Fig. 5 shows the simulation results for the PSNR versus the variance. Fig. 6 shows the error-free image as well as the results at $\sigma=1$ for TMR and IDMR at $k=1$ as an example.

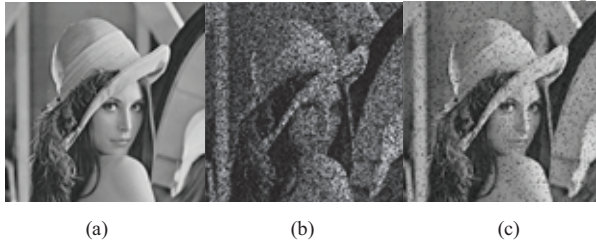


Fig. 6. (a) Error-free image, (b) TMR with $\sigma=1$ (PSNR=10.17dB), (c) IDMR with $\sigma=1$ (PSNR=18.80dB) ($k=1, n=8$).

From the results, several conclusions can be drawn.

(1) At the same k , if the variance increases, the PSNR decreases, because the error probability increases. (2) IDMR has a higher PSNR compared to TMR (the PSNR of the TMR is nearly constant, so nearly independent of the variance). This occurs, because an output error impacts in most cases the lower bits. The proposed scheme can tolerate small differences in values (mostly occurring in the lower bits) and still producing a voted output. The operation of a TMR is static, because it can only establish the strict majority. The proposed scheme achieves impressive improvements over the TMR scheme. (3) In all cases, the PSNRs decrease when the variance increases; therefore an increasing variance makes the final result more inexact for an approximate voting scheme such as IDMR.

VI. CONCLUSION

This paper has presented the analysis and design of a novel voting scheme whose operations are based on approximate computing. Approximate computing relaxes the strict static

voting and masking that modular redundancy schemes utilize to generate a correct output. This scheme is referred to as inexact double modular redundancy (IDMR) and utilizes approximate criteria when comparing and assessing the outputs of two modules for reliable computing. The IDMR voter has been designed at nanometric features sizes and simulated using HSPICE to assess different figures of merit such as delay, power dissipation, circuit complexity and process variability. TMR has the least delay, but it consumes more power and its tolerance to process variability is worse than the proposed scheme. Image processing has been analyzed as possible application of the proposed approximate voter; the PSNR has been measured and in most cases, the proposed scheme performs better than TMR. In conclusion, approximate voting by IDMR shows the following advantages over a TMR with static masking and exact voting.

(1) IDMR incurs in a reduced circuit complexity in the voting circuitry as well as in the modular replication.

(2) Except the delay, approximate voting hardware for IDMR improves over all circuit-level figures of merit such as power dissipation and tolerance to variations.

(3) For the considered application (image processing), approximate voting has a high PSNR value, thus consistently and significantly improving over an exact voting scheme such as TMR.

REFERENCES

- [1] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design and Test of Computers.*, vol. 22, no. 3, pp. 258–266, May–Jun. 2005.
- [2] J. von Neumann, "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," *Automata Studies, Ann. Of Math. Studies*, no. 34, C. E. Shannon and J. McCarthy, Eds., Princeton University Press, pp. 43–98, 1956.
- [3] N. Vaidya and D. Pradhan, "Fault-Tolerant Design Strategies for High Reliability and Safety," *IEEE Trans. On Computer*, vol. 42, no. 10, pp. 1195–1206, Oct. 1993.
- [4] J. Han, J. Gao, Y. Qi, P. Jonker, J. Fortes, "Toward hardware-redundant, fault-tolerant logic for nanoelectronics," *IEEE Design and Test of Computers*, vol. 22, no. 4, pp. 328–339, July–August 2005.
- [5] W.H. Pierce, *Failure-Tolerant Computer Design*, Academic Press, 1965.
- [6] J. Han and M. Orshansky, "Approximate Computing: An Emerging Paradigm for Energy-Efficient Design," in *ETS'13*, pp. 1–6, 2013.
- [7] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, 2013.
- [8] P. Reviriego, C.J. Bleakley, J.A. Maestro, "Diverse Double Modular Redundancy: A New Direction for Soft-Error Detection and Correction," *IEEE Design & Test*, vol. 30, no. 2, pp.87–95, April 2013.
- [9] P. Reviriego, C. Bleakley, and J.A. Maestro, "Structural DMR: A technique for implementation of soft error tolerant FIR filters," *IEEE Trans. Circuits Syst. II*, vol. 58, no. 8, pp. 512–516, Aug. 2011.
- [10] S. Mitra, E.J. McCluskey, "Word-voter: a new voter design for triple modular redundant systems," *Proc. 18th IEEE VLSI Test Symposium*, pp.465–470, 2000.
- [11] S.R. Chowdhury, A. Banerjee, A. Roy, H. Saha, "A high speed 8 transistor full adder design using novel 3 transistor XOR gates," *World Academy of Sciences*, vol. 22, 2008.
- [12] A. Rubio, J.F. Pàmies, E. Vatajelu, R.C. Corretger, "Process variability in sub-16nm bulk CMOS technology," Project: Terascale Reliable Adaptive Memory Systems, FP7-INFOS-IST -248789, 2012.