HARDWARE EFFICIENCY VERSUS ERROR PROBABILITY IN UNRELIABLE COMPUTATION

Yangyang Tang[1], *Emmanuel Boutillon*[1], *Christophe Jégo*[2] *and Michel Jézéquel*[3]

[1]Université de Bretagne Sud, UMR CNRS 3192 Lab-STICC, Lorient, France.
[2]Institut Polytechnique Bordeaux, UMR CNRS 5218 Lab-IMS, Bordeaux, France.
[3]Institut TELECOM/TELECOM Bretagne, UMR CNRS 3192 Lab-STICC, Brest, France.

ABSTRACT

For the purpose of mitigating the effect of transient error in unreliable architecture, many authors have proposed redundant computation (sensitive to error) and an error-free correction unit. In this paper, taking into account that the correction unit is also subject to error, we propose to evaluate the quality of an architecture using not only its efficiency (i.e. the normalized number of operation per area and per unit of time), but also its final output error rate. The new criteria, namely Reliability-Efficiency Criteria (RE-Criteria) thus defines a two dimensional space of solution, i.e. a Pareto distribution [1]. After revisiting well-known correcting techniques with the RE-Criteria, we give an example of Pareto distribution based on a classical FIR filter performed with the error-correcting mechanism based architectures.

Index Terms— Hardware Efficiency, Error Probability, Reliable Computation, FIR Filter, RNS.

1. INTRODUCTION

In the last two decades, the rapid development of logic circuit manufacturing has entitled the electronic device deep submicron or nanoscale. During the same period, the transient defects and faults have experienced major growth which has affected the reliability of electronic technologies. This has led to too much attention on the topic of cosmic radiation induced transient faults in nanoscale computation. Radiation-induced Soft-Error (SE) is considered as a major reliability concern to current rapid size shrinking device manufacturing. This event is called Single-Event Transient (SET) or Single-Event Upset (SEU) [2].

Several methods have been proposed to evaluate the impact of SET in Soft-Error Rate (SER) i.e., the probability of a wrong output result. A basic method is to simulate the impact of SET error on the output of the function. In the literature [3-4], several models have been presented to achieve efficient estimations of the susceptibility to transient fault in a small estimation error threshold. However, due to the circuit tending towards rather high complexity and large size fan-in, the present formal reliability analysis methods, namely analytical and symbolic methods [5], are hardly tackled with circuit evaluation. Under the circumstances, an alternative approach, based on a simple and pessimistic error-model with adequate reliability estimation ability, is more appropriate for this task. On the side of SE mitigation, a considerable amount of approaches has been proposed to mask or mitigate SE in different contexts. While those approaches laid emphasis on hardware complexity, power consumption, and delay overhead. Note that in most cases, the unit for detecting or correcting error is usually regarded as error-free. But this assumption does not conform to the trend. Accordingly, the need of formulating some metrics for the fault-tolerant and efficient featured design in undependable devices is forthcoming.

In this paper, taking into account that the correcting mechanisms are also subject to error, we present a new metric, termed as Reliability-Efficiency Criteria (RE-Criteria), to help design in unreliable computation. According to the requirement of the application, the Pareto distribution helps the designer to select the best compromise between hardware efficiency and resource reliability. In the proposed error probability metric, the model of transient error is simple: if a circuit of normalized area S has a probability p of generating a correct result during one clock cycle, a computation requiring a normalized area of $n.S$ during m clock cycles has a probability $p^{n.m}$ of generating a correct result. Since a metric implies all SETs impact the output that is not necessarily the case, the proposed metric is a worst-case metric. Indeed, this metric guarantees the level of output confidence. After revisiting well-known correcting techniques with the RE-Criteria, we detail an example of Pareto distribution based on a classical FIR filter performed with the error-correcting architectures.

The remainder of this paper is organized as follows: Section 2 presents the RE-Criteria. Section 3 derives the expressions of the RE-Criteria for the strategies. Section 4 gives examples of Pareto distribution of the RE-Criteria for a FIR filter using the different error-correcting strategies. Conclusions are given in Section 5.

2. THE RELIABLE EFFICIENCY CRITERIA

The main contribution of this paper is to characterize an

architecture in unreliable hardware by the two dimensional criteria: the Reliability (probability of non error in the output of the architecture) and the Efficiency of an architecture. From those two criteria, it is possible to plot a Pareto distribution for several architectures with different error detection and correction mechanism.

In this section, firstly the unreliable computing model is presented, as well as the unreliable decoder. Subsequently, we define a model of reliability upon the hypotheses. The RE-Criteria that consists of models of reliability and efficiency are introduced afterwards.

2.1. The unreliable computing model

A generic computing system with its capacity of correction is described in Fig. 1-a. It is composed of an encoder, a computing function, and finally, a decoder. The input of the encoder is assumed error-free. The encoder has to perform any coding prior to the computation. For example, triplication by wires or more complex coding involving computation is also subject to SET. The computing function performs the required computation in the adequate information representation. The decoder generates the output in the correct format after eventually detecting or correcting computation error due to SET. Although related work considers the decoder as error-free, in our study, the decoder can also be affected by a SET (see Fig. 1-b).

2.2. The model of reliability

We consider that the "unitary" probability of SET occurrence in the decoder is a constant for given CMOS processes, supply voltages and the external environment of the device, such as ionization level, temperature, etc. This "unitary" SET probability p_e denotes the probability of error in a unity area during one clock cycle. On the contrary, $\overline{p}_e$ denotes the probability of error-free computation in a unity area during one clock cycle:

$$\overline{p}_e = (1 - p_e), \; with \; 0 < p_e < 1. \qquad (1)$$

The error probability of the output is termed as P, as shown in Fig. 1-c, and the resulting no-error probability $\overline{P}$. In order to upper bound the value of P, three hypotheses are assumed in this study:

1) *Isotropy*: the unitary probability of SET is constant for all the design and independent of the underneath logic.

2) *Contamination*: If an error occurs in a design, or in one of its inputs, its output will be corrupted.

3) *Irreversibility*: Two successive errors cannot lead to a correct result, the final result will stay erroneous.

One should note that the hypotheses are all pessimistic and correspond to a worst-case configuration. In fact, the

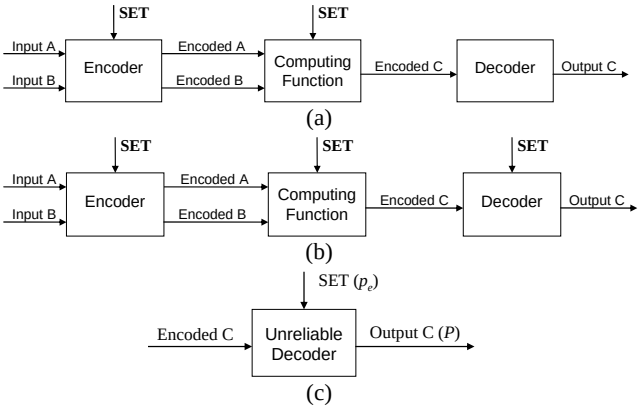


Fig. 1. a) Block diagram of a conventional computing system, b) Block diagram of an unreliable computing function system, c) A decoder diagram from unreliable computing system.

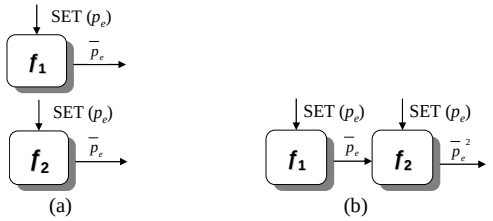


Fig. 2. a) Parallel functions, b) Serial functions.

error probability of a "unitary" area should depend on its logic. Thus, *Isotropy* is an approximation. Moreover, it is easy to find an example where *Contamination* is false. Thus, if a NAND gate has one input that is equal to zero, an error in the other input is not propagated. If two errors occur consecutively on a binary signal the result will be corrected, which again, is in contradiction with the *Irreversibility*. Finally, when dedicated error correcting hardware is used, *Contamination* and *Irreversibility* have not to be taken into account.

Nevertheless, those three hypotheses allow us to build a worst-case result, i.e., to guarantee a given probability of error for the output of the function. Although the definition of a more precise model is out of the scope of this paper, the explicit computation of error probability in output will be derived (Section 3).

Let us study how the error probability scales on a more complex design under the three hypotheses. Fig. 2 shows two configurations that both require two area units. In the first case, the parallel one in Fig. 2-a, the output is correct if the output of the two components are correct, i.e. the probability for error-free output is then $\overline{P} = \overline{p}_e^{\,2}$. In the second case, the serial one in Fig. 2-b, since an error in f_1 affects the output by *Contamination* and an error in f_2 also affects the output by *Irreversibility*, then both components

should be correct, which leads again to, $\overline{P} = \overline{p_e}^{\,2}$. This formula can be generalized to a circuit of any size n in terms of unit area. Moreover, if the same architecture is used during several clock cycles m to process an output, all cycles should be error-free. The probability of correct result is then given by

$$\overline{P} = \overline{p_e}^{\,n \cdot m}. \quad (2)$$

2.3. The RE-Criteria

A complexity-throughput metric observing the efficiency of an architecture has been proposed for Forward Error Correction (FEC) decoding in [6]. It defines the hardware efficiency γ of an architecture as the normalized number of operation per area unit and time unit. Note that the nature of the operation is not specified, it can be an FFT, a decoding iteration in an iterative decoder, a multiplier or anything else. Assuming that an architecture executes a function with a component of n area units and during m clock cycles, its efficiency is defined as,

$$\gamma = \frac{1}{n \cdot m}, \text{ operation / (area unit } \times \text{ time unit).} \quad (3)$$

Further, from (2), the error probability P in output can be expressed as,

$$P = 1 - \overline{P} = 1 - (1 - p_e)^{n \cdot m}. \quad (4)$$

Consequently, the proposed RE-Criteria consist of a couple (P, γ) of metrics that are the metric for the error probability and the metric for the hardware efficiency, expressed respectively in (4) and (3). From those two estimations, we are able to plot a Pareto distribution, which characterizes the tradeoffs between the efficiency and the reliability among the architectures. The following sections present the formal derivation of the RE-Criteria for different state of the art correcting strategies.

3. FORMAL DERIVATIONS OF THE RE-CRITERIA

After an introduction of the new metric, as well as the methodology, this section presents different conventional strategies to detect and/or correct errors. Considering single-error correction, error probabilities based on the metric are detailed for each case.

3.1. Spatial-Triple Module Redundancy

A conventional solution duplicates the functional unit. Basically, to apply Triple Modular Redundancy (TMR) [7], three identical units are designed to execute the function from the same input data. The three output results are sent into a majority voter that allows us to correct at most one

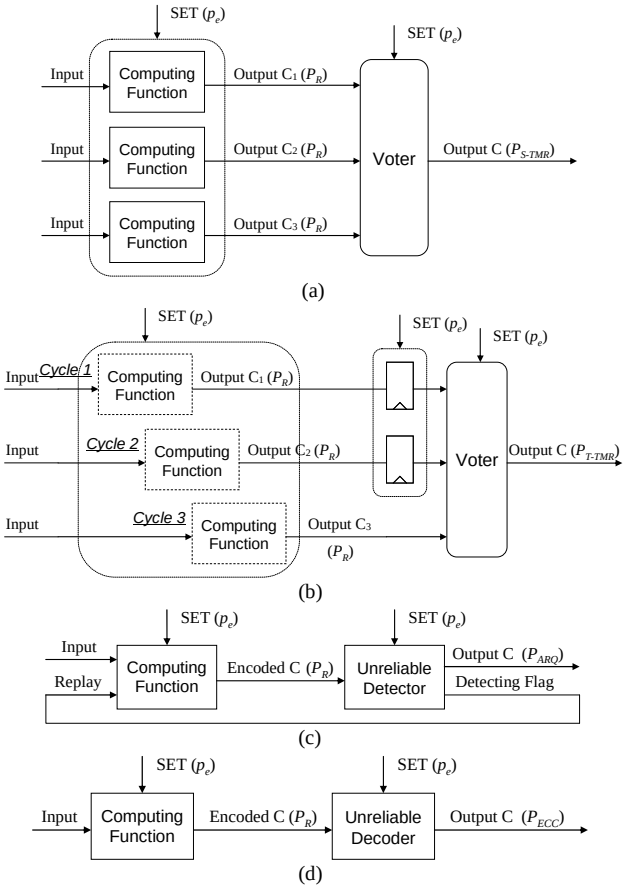


Fig. 3. a) S-TMR structure; b) T-TMR structure; c) ARQ structure; d) Error Control code structure.

error (Fig. 3-a). In summary, the output of the Spatial-TMR (S-TMR) is wrong when SETs induce at least two faulty modules (event of probability $P_{>1}$) or when there is one faulty module and the voter is faulty (event of probability Q). The resulting error probability of S-TMR decoder $P_{S\text{-}TMR}$ can be expressed as follows:

$$P_{S\text{-}TMR} = P_{>1} + Q \cdot \overline{P}_{>1}. \quad (5)$$

Let P_R be the error probability in a single module of size n that performs a computation in m cycles, as introduced in the last section, then,

$$P_R = 1\text{-}(1\text{-}\,p_e)^{n \cdot m}.$$

$P_{>1}$ is expressed as:

$$P_{>1} = \binom{2}{3} \cdot P_R{}^2 \cdot \overline{P}_R + P_R{}^3.$$

Moreover, let n_Q be the area cost of the voter that is used during only one clock cycle after the computation, thus,

$$Q = 1\text{-}(1\text{-}\,p_e)^{n_Q}.$$

With engaging the triplication and the voter, the efficiency

of S-TMR is can be reduced to:

$$\gamma_{S\text{-}TMR} = \frac{1}{(3 \cdot n + n_Q) \cdot m}. \quad (6)$$

Therefore, by using RE-Criteria, the S-TMR solution corresponds to a point ($P_{S\text{-}TMR}$, $\gamma_{S\text{-}TMR}$) in the Reliability-Efficiency two dimensional space, termed as RE-Space.

3.2. Temporal-TMR

If the same instance is considered, an alternative solution is to reuse the computing function, as shown in Fig. 3-b. Compared with S-TMR, temporal-TMR (T-TMR) [8] is regarded as a tradeoff between area and time. The resulting error probability for the T-TMR holds the same form as the S-TMR, $P_{T\text{-}TMR}$,

$$P_{T\text{-}TMR} = P_{>1} + Q \cdot \overline{P}_{>1} \quad (7)$$

However, the expression of $P_{>1}$ should be adapted because SEs can occur in the registers that contain the results of first computations during 2.m clock cycles, and the results of the second computations during m clock cycles. Let $\overline{R}_1$, $\overline{R}_2$ and $\overline{R}_3$ denote the probability of correct input into the voter in the first, the second, and the third cycle, respectively. More precisely, suppose n_R the area cost of each register, then $\overline{R}_1 = \overline{P}_R \cdot (1 - p_e)^{2 \cdot n_R \cdot m}$, $\overline{R}_2 = \overline{P}_R \cdot (1 - p_e)^{n_R \cdot m}$ and $\overline{R}_3 = \overline{P}_R$. $P_{>1}$ is thus expressed as:

$$P_{>1} = \sum_{i_1, i_2, i_3 \in (1,2,3)}^{i_1 \neq i_2 \neq i_3} R_{i_1} \cdot R_{i_2} \cdot \overline{R}_{i_3} + \prod_{i=1}^{3} R_i^3 \cdot$$

For the efficiency of the T-TMR, the clock cycle is tripled by reusing the function temporally,

$$\gamma_{T\text{-}TMR} = \frac{1}{3 \cdot (n + n_Q + n_R) \cdot m}. \quad (8)$$

As a register occupies slight area, its error probability is much smaller than P_R. Hence, $P_{T\text{-}TMR}$ is approximate to $P_{S\text{-}TMR,}$ as well as its hardware efficiency.

3.3. ARQ

The Automatic Repeat reQuest (ARQ) [9] technique performs the error-detection by adding some redundancy. When an error is detected, the detector asks to restart the computation. The data is sent once it is checked as error-free, as shown in Fig. 3-c. Generally, the output of the redundant computation includes three cases: error-free, error-detectable, and error-undetectable. Error-detectable output (probability $P_{R\text{-}1}$), that is erroneous and is able to be detected by the detector, is not a codeword. Otherwise, the output that turns into one of the codewords is termed as error-undetectable output (probability $P_{R\text{-}2}$). Consider an error-detecting code (c_n, c_k, $c_{n\text{-}k}$), $P_{R\text{-}1}$ and $P_{R\text{-}2}$ are thus expressed as:

$P_{R\text{-}1} = \frac{2^{c_n} - 2^{c_k}}{2^{c_n} - 1} \cdot P_R$, and $P_{R\text{-}2} = P_R - P_{R\text{-}1}$.

Accordingly, in the case of error-free output of computation, when the detector is faulty it restarts the computation, otherwise, the result is correct. In the error-detectable case, when the detector is faulty then the result is erroneous, otherwise the computation is restarted. At last, the detector directly gives erroneous output. Assume that all the inputs required in the replay, that are fed into the computation or stocked in the computation, are correct. Consequently, the resulting error probability for the k-time ARQ is

$$P_{ARQ} = (P_{R\text{-}1} \cdot T + P_{R\text{-}2}) \cdot P_{Replay}^k, \; k \in [0, \infty), \quad (9)$$

where P_{Replay} represents the probability of that one replay is triggered and T the error probability of the detector. If the detector takes n_T unit area and needs m_T clock cycles for each detecting process, then $T = 1\text{-}(1 - p_e)^{n_T \cdot m_T}$, and

$$P_{Replay} = P_{R\text{-}1} \cdot \overline{T} + \overline{P}_{R\text{-}1} \cdot T.$$

Moreover, the number of cycle Ns for the k-time ARQ is

$$Ns = \sum_{k=0}^{\infty} (k+1) \cdot (n + n_T) \cdot P_{Replay}^k \cdot \overline{P}_{Replay},$$

that can be simplified as:

$$Ns = \frac{(n + n_T)}{\overline{P}_{Replay}}.$$

Hence, the efficiency for the k-time ARQ is expressed as:

$$\gamma_{ARQ} = \frac{\overline{P}_{Replay}}{(n + n_T) \cdot (m + m_T)}. \quad (10)$$

Recall that, for the sake of the redundancy engaging, n and m that are used to compute the error probability P_R here are different from the original function. More precisely, the error probability rapidly decreases for an increase of k. Meanwhile, it degrades the latency and data rate. Consequently, the compromise lies within RE-Space (P_{ARQ}, γ_{ARQ}).

3.4. Error Control Code

Instead of restarting the computation, a decoder can perform both the detection and correction, known as the error control code, as described in Fig. 3-d. This technique is an extension of the S-TMS technique that uses the repetition error control code (3, 1, 2) to correct a single error. Consider the output of the computation includes three cases, error-detectable, error-undetectable and error-free. Thus, the resulting data is erroneous when the output of the computing function is error-detectable if the decoder is also faulty. Moreover, the resulting data is also erroneous when the output of the computing function is error-undetectable.

Hence, the error probability for the error control code decoding based computation P_{ECC} is

$$P_{ECC} = P_{R\text{-}1} \cdot D + P_{R\text{-}2}, \quad (11)$$

where D denotes the error probability of the decoder. Let n_D unit area and m_D clock cycles for the decoder, then $D = 1\text{-}(1\text{-}p_e)^{n_D \cdot m_D}$. If an error-correcting code (c_n, c_k, $c_{n\text{-}k}$) is considered, then $P_{R\text{-}1}$ and $P_{R\text{-}2}$ have the same expression as in the ARQ case.

Moreover, the efficiency for the error control code is:

$$\gamma_{ECC} = \frac{1}{(n+n_D)\cdot(m+m_D)}. \quad (12)$$

Naturally, a more complex coding scheme can be used to increase the efficiency/error-probability capability of the code, i.e. iterative decoding. To sum up, (P, γ) RE-Space that is obtained by using the RE-Criteria, is able to illustrate the tuning among those parameters (reliability, efficiency, latency, data rate, etc…) for the requirements.

4. PARETO DISTRIBUTION: HARDWARE EFFICIENCY VS. ERROR PROBABILITY

In this section, the RE-Spaces (P, γ) draw out the Pareto distribution for different strategies that were previously detailed. These distributions illustrate the trade-offs between the hardware efficiency and the transient error probability.

4.1. The experimental results

In this work, we aim at demonstrating the interests of Pareto distribution based on the RE-Criteria. Therefore, a well-known pipeline FIR filter is considered as a case study. Let $x(n)$ be the input integer of the FIR filter and $y(n)$ be the output integer,

$$y(n) = \sum_{i=0}^{N} x(n-i) \quad (13)$$

In order to perform the error-correction in the filter, the TMR technique that is carried out by duplicating the modules in spatial or in temporal has been first applied as shown in Fig. 4-a. Moreover, the ARQ technique applied to FIR filter is able to restart the computation by using a detector, see Fig. 4-b. We also turn our attention to FIR filters using RNS [10] arithmetic so as to implement error control code based correction. The RNS implementation of a FIR filter is decomposed into k parallel filters [11]. By adding some redundant residues, the RNS that has the error detection and correction properties is called Redundant RNS (RRNS). Moreover, the technique, called Bidirectional RRNS (BRRNS) [12], that requires redundant moduli for satisfying some constraints to achieve fast error correction, can be regarded as an error control code. A FIR filter implanted in RNS arithmetic, with applying BRRNS self-diagnosis decoder, is shown in Fig. 5.

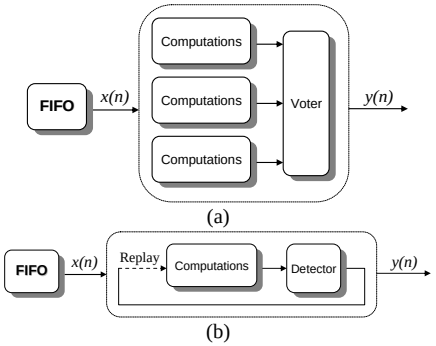


Fig. 4. a) S-TMR based FIR filter structure; b) ARQ based self-detecting FIR filter structure.

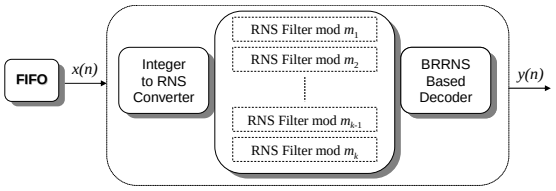


Fig. 5. FIR filter implemented in RNS arithmetic, with also a BRRNS self-diagnosis decoder.

Table I details the synthesis results of size various FIR filters in terms of number of slice (the elementary programmable logic block in FPGAs) and clock frequency. In this work, four FIR filters have been investigated for different order N and different input data bit length l. Each filter has been designed for the four strategies. Note that the simplex denotes the filters where no correcting mechanism is applied. The error correcting techniques implemented here are: the FEC kind, the S-TMR and the BRRNS based; the ARQ kind, modulo-4 check, modulo-8 check, modulo-16 check and the spatial Double Module Redundancy (DMR). The proposed criteria are under *Isotropy* hypothesis. It means that SET probability p_e is constant (see Section 2). For the sake of facility, p_e is settled to 10^{-13} (equivalent to the value of one FIT). By using (5-12), the error probability and the efficiency of the filters are estimated. Moreover,

TABLE I. SYNTHESIS RESULTS FROM XILINX VIRTEX 5 IN TERMS OF [SLICE(#)/CLOCK FREQUENCY(MHZ)] FOR THE FIR FILTERS

FIR Filter Sizes (*N, l*)	**FIR I** (16, 5)	**FIR II** (32, 6)	**FIR III** (64, 7)	**FIR IV** (128, 8)
Strategies	**Slice/Fre.**	**Slice/Fre.**	**Slice/Fre.**	**Slice/Fre.**
The Simplex	[94, 424]	[128, 417]	[171, 410]	[232, 403]
S-TMR	[249, 405]	[383, 398]	[508, 300]	[681, 309]
BRRNS Based	[477, 353]	[843, 263]	[1133,232]	[1686,227]
ARQ-Modulo-4	[148, 349]	[205, 312]	[241, 300]	[383, 298]
ARQ-Modulo-8	[168, 349]	[237, 312]	[276, 300]	[408, 298]
ARQ-Modulo-16	[191, 349]	[277, 312]	[322, 300]	[426, 298]
ARQ-Spatial DMR	[207, 349]	[302, 312]	[409, 300]	[517, 298]

both the values of error probability and hardware efficiency for different filters are normalized by the ones of the simplex considered as the reference. Since the error probability of T-TMR is estimated approximately as the S-TMR, its results are skipped.

4.2. Pareto distributions for a FIR filter

The Pareto distributions are built from the RE-Spaces that are obtained from the RE-Criteria. Not only the performances in terms of the fault-tolerant and the hardware efficient for different error correction strategies are able to be obtained, but also other features can be predicted. The features for the FIR filters obtained from the different techniques are pointed out in Fig. 6.

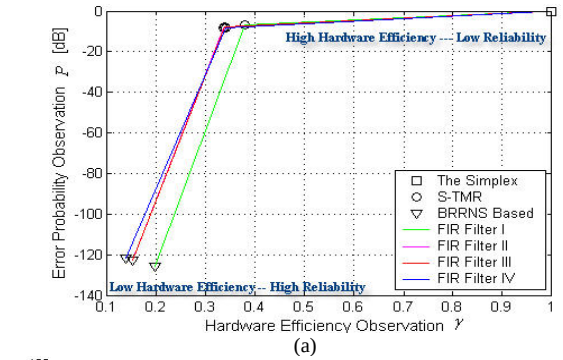
Classically, the use of correction techniques results in the degradation of the hardware efficiency, but yields an increase of the reliability. Moreover, the more sophisticated error correcting technique is used, the better error-resistant performance can be acquired. For instance, the S-TMR based filters that are carried out by the naive method, S-TMR, hold high hardware efficiency with low reliability compared to the one of error control code, see Fig. 6-a. In addition, different detecting methods tune the features, such like, since the spatial DMR based is less efficient than the other Modulo Check methods, but more reliable.

Actually, two methods can achieve high reliable feature design: the add-redundancy based method, such as the error control code and the reuse-function based method, such as the ARQ. The ARQ that is not able to correct some errors during the computation can not ensure the data rate. Nevertheless, it is the most efficient and robust one among the three strategies. With the increase of replay time *k*, its error-resistance sharply increases.

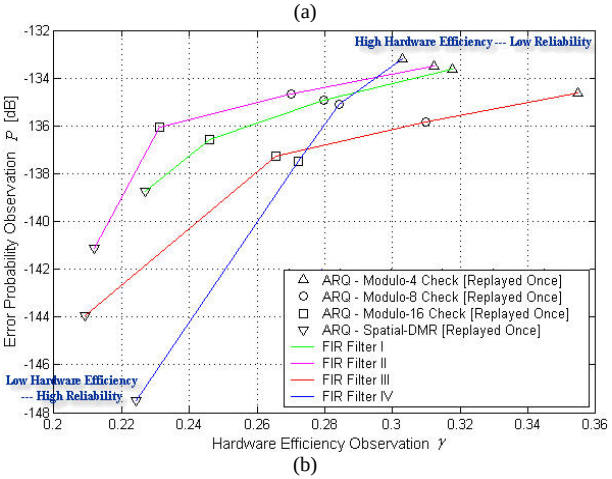
As a consequence, by building the Pareto distributions, the designer is able to explore different strategies that depend on the system constraints, such as, hardware resources, power consumption, latency, data rate, and etc.

5. CONCLUSION

Efforts are necessary to pursue the hardware efficient and robust featured design. In this paper, a Reliability-Efficiency criteria, called RE-Criteria that is a combination of error probability criteria and hardware efficiency criteria has been proposed. Subsequently, conventional mechanisms have also been revisited for the two dimensional analysis based on our criteria. Finally, some Pareto distributions built for the error-correcting techniques based FIR filters have been given to illustrate the interest of RE-Criteria. This new metric can be useful for the design of efficient and robust components in the unreliable nanoelectronic system.



(a)



(b)

Fig. 6. Pareto distributions: a) the FEC kind; b) the ARQ kind.

6. REFERENCES

[1] B. C. Arnold, "*Pareto Distributions*," International Cooperative Publishing House, Fairland, MD, 1983.

[2] R. C. Baumann, "Soft errors in advanced semiconductor devices-part I: the three radiation sources," *IEEE Trans. DMR.*, vol. 1, no. 1, pp. 17-22, 2001.

[3] M. Zhang and N. R. Shanbhag, "Soft-Error-Rate-Analysis (SERA) methodology," *IEEE Trans. CADICS.*, vol. 25, no. 10, pp. 2140-2155, 2006.

[4] N. Miskov-Zivanov and D. Marculescu, "Modeling and optimization for soft-error reliability of sequential circuits," *IEEE Trans. CADICS.*, vol. 27, no. 5, pp. 803-816, 2008.

[5] N. Miskov-Zivanov and D. Marculescu, "Formal modeling and reasoning for reliability analysis," *Proc. DAC.*, pp. 531-536, 2010.

[6] O. Muller, "*Generic multiprocessor architectures for high-throughput turbo-communications*," Ph.D. dissertation, Univ. Bretagne-Sud and TELECOM Bretagne, France, 2007.

[7] W. V. Subbarao, "fault tolerant digital system design," *Proc. SECON.*, vol. 1, pp. 124-128, 1991.

[8] B. Shim and N. R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Trans. VLSIS.*, vol. 14, no. 4, pp. 336-348, 2006.

[9] S. Lin, D. Costello and M. Miller, "Automatic-repeat-request error-control schemes," *IEEE Mag. Commu.*, vol. 22, iss. 12, pp. 5-17, 1984.

[10] H. L. Garner, "The residue number system," *IRE Trans. EC.*, Vol. EC-8, pp. 140-147, 1959.

[11] R.Conway, "Efficient residue arithmetic based parallel fixed coefficient FIR filters," *Proc. ISCAS.*, pp. 1484-1487, 2008.

[12] Y. Tang, E. Boutillon, C. Jégo and M. Jézéquel, "A new single-error correction scheme based on self-diagnosis residue number arithmetic," *Proc. DASIP.*, pp. 27-33, 2010.