
The TIMA Lab. Research Reports Collection

ISSN 1292-8062

# Cost Reduction and Evaluation of a Temporary Faults Detecting Technique

Lorena ANGHEL, Michael NICOLAIDIS

TIMA Laboratory, 46 avenue Félix Viallet, 38 000 Grenoble France

ISSN TIMA-RR--01/02-01--FR

**TIMA Laboratory**,46 avenue Félix Viallet,

38 000 Grenoble, France

# Cost Reduction and Evaluation of a Temporary Faults Detecting Technique

Lorena ANGHEL, Michael NICOLAIDIS
TIMA, France

**Abstract:** *IC technologies are approaching the ultimate limits of silicon in terms of channel width, power supply and speed. By approaching these limits, circuits are becoming increasingly sensitive to noise, which will result on unacceptable rates of soft-errors. Furthermore, defect behavior is becoming increasingly complex resulting on increasing number of timing faults that can escape detection by fabrication testing. Thus, fault tolerant techniques will become necessary even for commodity applications. This work considers the implementation and improvements of a new soft error and timing error detecting technique based on time redundancy. Arithmetic circuits were used as test vehicle to validate the approach. Simulations and performance evaluations of the proposed detection technique were made using time and logic simulators. The obtained results show that detection of such temporal faults can be achieved by means of meaningful hardware and performance cost.*

## 1. Introduction

Modern electronic circuits are highly complex systems and, as such, are prone to occasional errors or failures. Device size reduction, increased operating frequency and power supply reduction that accompany the process of very deep submicron scaling, reduce noise margins and IC reliability, but also increase the impact and complicate the behavior of defects. This makes increasingly difficult to achieve acceptable reliability levels for future ICs and maintain acceptable cost and quality for IC testing. It seems that the most significant problem concerns the sensitivity of future IC generations face to various noise sources, and in particularly face to energetic particles. Such particles are for instance, cosmic neutrons produced by the sun activity. It is predicted that the energy and flow of these particles are sufficient for creating unacceptable soft-error rates in future IC generations, even at ground level. Alpha particles produced by the disintegration of radioactive isotopes contained in the material of electronic systems are becoming another cause of increasing soft error rates in these technologies.

One basic reason for increased sensitivity to noise is the reduction of $V_{DD}$ voltage and geometry shrinking which implies the reduction of node capacitance. Thus, the charge stored on a circuit node ($Q = V_{DD} * Cnode$) is drastically reduced. In fact, a significantly lower charge needs to be deposed by a particle strike in order to reverse the logic value of a node.

In memory cells, a particle striking a node of the cell produces a transient pulse that can be captured by the asynchronous loop forming the cell, resulting on a soft-error. Traditionally, only memories were protected against SEUs (single event upset) by means of error detecting and correcting codes. Deeper submicron scaling increases the sensitivity of logic networks too. A transient pulse created by a particle strike on a node of the network can be propagated to the network outputs. However, the transient pulse is often attenuated before reaching these outputs [BAZ 97]. In addition, reaching the network outputs does not necessary imply the occurrence of a soft-error. This will happen only if the pulse reaches an output at the time the latching edge of the clock is occurring. In this case, the erroneous value is stored in the latch connected to the output of the circuit.

Transient pulses wider than the logic transition time of a gate will propagate without attenuation and can affect the correct operation of the circuit. Transient pulses induced by particle strikes have a width of few hundreds of picoseconds (the exact value depends on circuit topology and particle energies). Thus, their duration is becoming higher than the transition time of gates in deep submicron technologies. Therefore, transient pulses are not attenuated even for relatively low energy particles. In addition, as the clock frequencies are increasing significantly, the probability of latching a transient pulse is increased by the same factor. In fact the more frequent are the latching edges of the clock, the higher is the probability to have a transient pulse coinciding with a latching edge. Due to these trends, error rates in logic parts risk to become as high as error rates in memories.

Another significant problem concerns timing errors. Such errors are gaining importance in very deep submicron and nanometer technologies. Process parameter variation, various defect types (shorts, opens…) often affect circuit speed. These faults are increasing the path delays and may result on timing errors unacceptable in the future high frequency integrated circuits. In addition, these timing errors require very complex test conditions. The huge number of paths of modern circuits, together with

other timing critical conditions such as cross talk, or ground bounce, make ATPG for timing faults computationally unfeasible, and test length unrealistic. It is therefore unavoidable that a significant number of circuits with timing faults will pass fabrication tests.

In this context, it will become mandatory to design the future ICs to be tolerant for soft errors and timing faults. Because this protection will be needed for any product, including commodity ones, traditional fault tolerant design such as TMR can not be used due to its high cost. The most economic solution is to use a concurrent checking scheme combined with a retry procedure. With such a scheme, soft-error tolerance can be achieved by a retry operation after each error-detection. The same principle will be used for timing error tolerance, but the clock cycle has to be reduced during the retry procedure. One possible concurrent checking scheme is based on self-checking design. In some situations this approach may offer low-cost concurrent error detection (e.g. self-checking multipliers using arithmetic codes [PET 58] [PET 72] [AVI 73] [ALZ 99]). In other circuit cases, self-checking design may require high hardware cost, for instance in the case of arbitrary logic functions.
Several other design solutions for achieving soft-error tolerance were proposed in [NIC 99]. The idea is to take advantage of the temporal nature of transient faults, and achieve tolerance of these faults by using time redundancy. This should lead on a significant reduction of hardware cost. Among these solutions, in the present work we have considered a soft-error detecting scheme that can be implemented without any performance penalty, and can also be used to detect timing errors. In order to evaluate the scheme we have used several types of multipliers and adders as test vehicles.
The scheme of soft-error detection is described in section 2. Evaluations of speed degradation and area overhead are shown in section 5. Simulations of transient faults have been performed in order to validate the fault tolerance merits of the design and are presented in section 7.

## 2. Concurrent Checking Based on Time Redundancy

Since the transient faults are manifested for a limited duration of time, one can design circuits such that the correct values are present on the circuit outputs for a time duration greater than the duration of the transient fault. A time-domain majority voter can be used to determine the value of the circuit output for a majority of time. Various transient-fault tolerant implementations of this principle were proposed in [NIC 99]. However, these schemes affect the performance of the circuit, so they can be used without performance penalty only for blocks that are not in the critical path of an IC. A technique performing transient error detection by means of time redundancy is also proposed in the same paper. This technique is considered here because it avoids performance penalty.
Figure 1 shows the simplified scheme of the detection technique. The combinational circuit is monitored by a circuit able to detect the occurrence of soft-errors produced by a transient fault affecting the combinational circuit. The detection circuit of fig. 1a is composed of two latches per circuit output and one comparator. The input of one latch is coming directly from the combinational circuit output, while the input of the second latch is coming from the same output, but is delayed by $\delta$ = Dtr + Dsetup (where Dtr is the maximum transient pulse duration that has to be tolerated, and Dsetup is the setup time of the latches). The latching edge of Ck should occur at $t_0$+ Dtr + Dsetup, where $t_0$ is the instance that the latching edge occurs in a standard circuit design. It is obvious that the delay $\delta$ introduces a reduction of the circuit speed.
An improved alternative is shown in fig. 1b. At instant $t_0$ the clock Ck captures the output of the combinational circuit in Output-Latch.

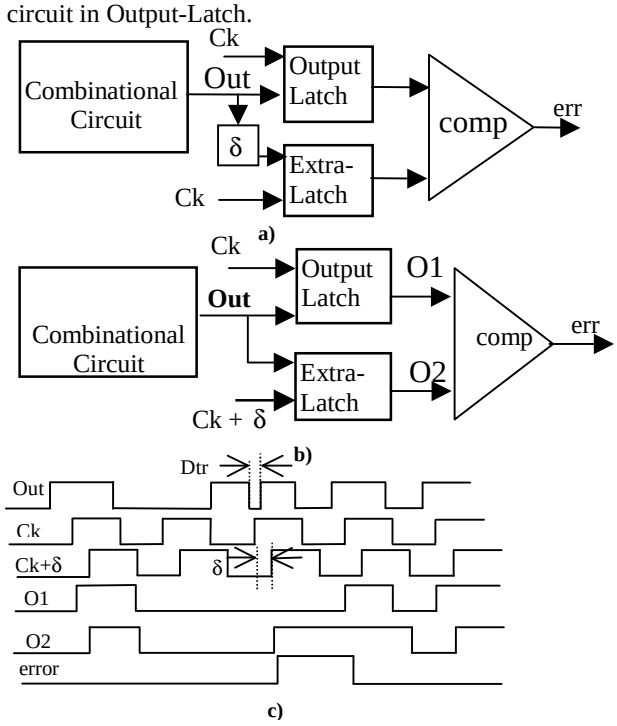



**Figure 1a), 1b) -** Transient fault detection implementation using a latch and a comparator; **1c)** Transient fault detection principle

A second clock having its latching edge shifted by $\delta$ latches this output in Extra-Latch at instant $t_0$ + $\delta$. A comparator indicates an error occurrence by a pass/fail signal. The designer determines the maximum transient pulse duration Dtr that must be tolerated in order to achieve an acceptable level of soft error rates, and sets $\delta$ = Dtr + Dsetup.

From this construction, it is obvious that no transient pulse of a duration lower than Dtr can be captured by both Output-Latch and Extra-Latch (fig 1c). Thus, the comparator detects any transient of duration lower than Dtr that could result on a latched erroneous value.
Due to the shifted clock of Extra-Latch, this scheme can also involve a performance penalty. However, a careful analysis allow us to avoid this penalty. In fact, the output results of the combinational circuit are latched in Output-Latch at $t_0$. These results can be sent to their destination without added delay. But, applying new inputs on the combinational circuit at the normal operation speed is not yet possible, since it may affect the value captures by Extra-Latch at $t_0+\delta$. This drawback can be eliminated if we exploit the delays of the combinational circuit. Let us consider the following situations:

i) The propagation delay Dmin, of the shortest path of the circuit respects the relationship Dmin > δ. Since no circuit output is affected by the new input values before $t_0$ + Dmin, then the circuit of figure 1b can operate at the same clock frequency as the conventional design without affecting the values latched by Extra-Latch.
ii) Dmin < δ. In this case, one can add delay elements in the shortest paths (use gates with higher delays or add inverters and/or buffers) to meet condition Dmin > δ.

This scheme requires a low hardware cost corresponding to some delay elements, an Extra-Latch per circuit output and a comparator (note that Output-Latch is also the functional latch of the circuit). In addition, no performance penalty occurs.

Finally, it is worth to note that the scheme of figure 1b can also be used to detect timing errors, as far as the extra delay do not exceed δ. However, when a retry procedure is activated to correct a possible soft error, the timing error can be produced again. In order to cope with this situation, one can implement a clock control circuit (driven by the error detection signal). This circuit will increase the clock cycle by duration higher or equal to δ. After the clock cycle increase, the retry procedure is activated again to correct the error.

The shifted clock (Ck+δ) used in figure 1b) can be obtained from Ck by using delay elements. Another possibility is to fix the same δ for all the blocks of an IC and use a second clock distribution network to bring Ck+δ. To ensure a precise value for δ, careful implementation is needed for delay blocks, or for the clock distribution network, in order to keep the clock skews within acceptable values. A more convenient solution consists on using the rising edge of Ck as the latching edge of Output-Latch and the falling edge of Ck as the latching edge of the Extra-Latch. This solution requires the modification of the duty cycle of Ck, so that the high level of the clock must have duration $T_1=\delta$. However, if $T_1>\delta$, it is also possible to leave unchanged the duty cycle of Ck, but we need to introduce a delay equal to $T_1-\delta$ on the input of Extra-Latch (see figure 2).

Thus, while the latching edge of Extra-Latch occurs at $t_0+ T_1$, the captured value is the value presented on Output at the instant $t_0+\delta$. This way we can use a single-clock distribution network..

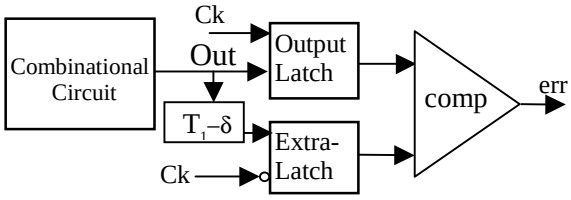


Figure 2. Clock network simplification.

Another technique allowing concurrent checking of temporary faults is presented in [FRA 94]. This technique is proposed for timing faults, but it can detect transient faults as well. It uses a transition detector to monitor the signals to be checked. Transitions occurring during the steady state phase of the monitored signals due to timing faults or transient faults are thus detected. Due to this difference, the technique proposed in [FRA 94] can not detect soft errors produced by particles striking the nodes of the Output-Latch. In addition, the implementation of the transition detector can not be done by means of standard logic design techniques. The detection of signal transition is also exploited in [MET 98] for checking timing faults and soft errors. This paper uses an improved transition detector able to detect its own faults.

## 3. Fault modeling

The fault modeling includes transients faults externally generated which affect a single node at a time. When a particle (e.g. heavy ion) strikes a semiconductor material, it creates electron-hole pairs in its track. If this particle goes into a depletion region, the electrons and holes created will drift according to the electric field in the depletion region. This movement of electrons and holes in opposite direction causes charges to be collected on the opposite sides of the depletion region. Typically the duration of such pulses is a few hundreds picoseconds.
Simulations considering this family of pulses are necessary if we know the exact shape of the transient pulse for a particular particle strike and we want to determine the exact shape of the circuit response. However, if we look for an average evaluation of a detection technique that has to work for various pulse shapes and duration, exact pulse shape is not of importance. In the present analysis, we considered square pulse shape. Duration of several hundreds of picoseconds was considered for the injected pulses. The evaluation of the present technique requires a good accuracy concerning the timing characteristics of the pulse reaching the circuit latches (width of the pulse and instance that the pulse reaches the latch). Electrical simulations (e.g. SPICE3) are very slow on large circuits and we can not use this kind of tool to

perform several hundreds of thousand of simulations at it was done in the evaluation. Thus, we have used the Verilog-XL logic simulator that offers also the necessary timing accuracy. A more dedicated simulator for transient faults could also be used for these experiments [CHA 93]. Transient pulses of duration up to 1ns have been used to validate the principle. This duration is much higher than the practical transient pulses induced by particle strikes.

As concerning injection of timing faults, we added delays within the circuit paths. Delays up to 20% of the maximum delay of the circuit were considered. Obviously this model does not represents the complex mechanisms creating timing errors in ICs, but again, the efficiency of the detection scheme does not depend on the timing fault mechanism, but on its effects (extra delay on output value).

## 4. Circuit Samples

To perform our experiments and validate the approach, we considered various multiplier and fast adder structures. Brown multipliers are unbalanced structures having a wide dispersion of propagation delays. For large operand sizes, the signal delays make this kind of multiplier unacceptable in practical applications, so multiplier structures with logarithmic delays were also investigated (Wallace trees, Booth encoding structures). To further improve the speed, various fast adder structures were used for the carry propagate adder forming the last stage of the multiplier. Booth multipliers using Wallace trees and fast carry propagate adders are the faster and most balanced structures.

The experimented adder structures include carry lookahead adders using CLU units [HWA 79][WES 94], as well as Brent & Kung [BRE 82], Kogge & Stone [KOG 73], Sklanski [SKL 60] and Han & Carlson [HAN 87] adders using Brent & Kung cells in various cascading structures. These adders present optimal trade-offs on speed, area and fan-out.

Each multiplier and adder was configured as in fig. 1b. The clock frequency was established according to the critical path of the functional circuit. That is, the circuit is operating at its maximum speed, as in the case of a design not using fault tolerant mechanisms.

These structures were considered for evaluating the hardware cost of the technique. However, concerning soft-error detection evaluation, we have taken into account examples of adders and multipliers of 8x8 size only, due to the large number of simulations to be performed.

## 5. Experimental results

To avoid performance lost, we have implemented only the technique described in figure 1b, that is, the circuit must be constrained to have Dmin>δ. This was done by performing constrained circuit synthesis using Synopsys optimization and timing analyzing tools. The multipliers and adders were synthesized by using 0.35μm AMS technology (3.3V), which was available to us.

| | | PAO(%)/TAO(%) | | | | | |
|---|---|---|---|---|---|---|---|
| | **CP (ns)** | **δ=0.46 (ns)** | **δ=0.6 (ns)** | **δ=0.8 (ns)** | **δ=1 (ns)** | **20% Dmax** | **20% Dmax improved** |
| Brown 16 | 32 | 0.2/4.3 | 1.2/5.3 | 3.2/7.3 | 4.5/8.5 | 44.2/46.8 | 16.3/19.5 |
| Brown 32 | 66 | 0.4/2.4 | 0.5/2.5 | 1.5/3.5 | 2.6/4.6 | 18.2/19.8 | 6.9/8.68 |
| Wallace 8 | 7 | 0.9/7.5 | 3.1/9.5 | 3.4/9.9 | 3.5/10 | 10.8/17.3 | 5.7/11.4 |
| Wallace 16 | 11 | 0.1/3.5 | 0.2/3.6 | 0.4/3.8 | 0.5/4 | 15.3/18.3 | 6.2/9.2 |
| Wallace 32 | 18 | .05/1.3 | .09/1.4 | .15/1.5 | 0.2/1.5 | 3.5/4.8 | 0.06/1.3 |
| Booth-wall 16 | 11 | 0.03/4 | 0.7/4.7 | 1.1/5.2 | 1.2/5.3 | 17.9/21.3 | 4.36/8 |
| Booth-wall 32 | 16 | .09/2.3 | .12/2.3 | .16/2.3 | 0.2/2.4 | 3.2/5.4 | 0.1/2.26 |
| Booth 8 | 8 | 0.3/6.5 | 1/7.2 | 1.2/7.5 | 2.4/8.8 | 19.7/25.3 | 4.2/9.6 |
| Booth 16 | 14 | 0.2/3.9 | 0.3/4 | 0.4/4.2 | 0.9/4.8 | 39.5/42 | 12.6/15.1 |
| Booth 32 | 20 | 0.1/2.6 | .16/2.7 | .2/2.78 | 0.3/2.9. | 3/5.62 | 0.09/2.6 |

Table 1. Area overhead results for different sizes and structures of multiplier.

Table 1 shows the results of these experiments over various multiplier structures (see column 1). The experiments were performed for values of δ equal to 0.46ns, 0.6ns, 0.8ns and 1ns. We have also considered the case δ = 20%Dmax, which can be used to detect timing errors of a delay up to 20% of the maximum delay of the circuit. Table 1 shows the critical path delay (CP) for each multiplier and the area overhead. The first component of the area overhead (PAO) corresponds to the cost required for achieving Dmin > δ. We remark that the area overhead is low, especially when considering the multipliers of practical interest such as Wallace and Booth-Wallace. The second component of the overhead columns (TAO) shows the total area overhead (overhead for Dmin > δ and overhead for the Extra-Latches and the comparator). For δ values up to 1ns (which largely exceed practical width of transient pulses produced by particle strikes), the area overhead is within low values and becomes insignificant as the multiplier size increase. Of course these results concern a 0.35 μm process but indicate that the technique is of low cost. For δ=20%Dmax, used for timing fault detection, the area overhead is again low, except for Brown multipliers. We will see later how we can reduce this cost to obtain the results of the last column.

Let us further analyze the technique by considering the two components of the area overhead. The first component includes an Extra-Latch per output, plus the cost of the omparator (one XOR gate and one OR gates per output). The percent overhead due to this component is determined by the (area per output) *APO* ratio = *(circuit area)/(# outputs).* The higher is *APO* ratio, the lower is the overhead. The second component is the area required to implement the constraint Dmin>δ (PAO). For circuits with

balanced delays, this cost will be low. However, for those circuits where a large number of paths have much lower delays than the critical path, the cost may become high. Although modern synthesis tools, used to reduce critical path delays, trend to generate circuits with balanced delays, there should be still some cases of unbalanced circuits.

Multipliers result on low hardware cost, because they have high *APO* ratio and are usually delay-balanced (especially the fast ones). For non-favorable cases, we have experimented the technique over various adder structures. Adders have a low *APO* (area per output) ratio. In addition, there is a large amount of paths with low delay with respect to the critical path (e.g. the delay between an input Ai or Bi and the output Si is only two XOR gates for any i). Table 2 gives the results for various adder cases. We observe that the cost is much higher than in the case of multipliers, reducing the interest of using the for adders.

| | CP (ns) | TAO(%) | | | | |
|---|---|---|---|---|---|---|
| | | δ=0.46 | δ=0.6 | δ=0.8 | δ=1 | 20% Dmax |
| **Brent&Kung 8** | 3 | 39.25 | 48.54 | 59.54 | 62.42 | 48.54 |
| **Brent&Kung 16** | 4.2 | 34.99 | 45.46 | 52.43 | 58.10 | 53.4 |
| **Brent&Kung 32** | 5.1 | 32.55 | 42.54 | 50.03 | 55.05 | 55.05 |
| **Han Carlson 8** | 2.8 | 37.53 | 47.3 | 53.6 | 70.2 | 47.3 |
| **Han Carlson 16** | 3.6 | 31.71 | 40.60 | 48.92 | 51.68 | 45.8 |
| **Han Carlson 32** | 4.2 | 27.69 | 36.8 | 41.33 | 48.39 | 42.1 |
| **Kogge&Stone 8** | 2.2 | 31.72 | 33.75 | 38.47 | 46.84 | 31.72 |
| **Kogge&Stone 16** | 2.5 | 26.6 | 27.4 | 31.21 | 37.52 | 27.0 |
| **Kogge&Stone 32** | 3.1 | 22.06 | 22.82 | 26.53 | 31.14 | 22.82 |
| **Sklanski 8** | 2.3 | 34.97 | 38.0 | 44.66 | 54.39 | 34.97 |
| **Sklanski 16** | 2.5 | 29.35 | 32.5 | 37.66 | 44.1 | 30.6 |
| **Sklanski 32** | 4.9 | 26.7 | 29.29 | 33.48 | 38.65 | 37.1 |
| **CLA 8** | 3.1 | 37.62 | 43.21 | 53.66 | 70.21 | 43.21 |
| **CLA 16** | 3.6 | 37.74 | 44.1 | 50.10 | 67.49 | 50.10 |
| **CLA 32** | 5.2 | 35.97 | 42.65 | 46.98 | 65.48 | 68.8 |

Table 2. Area overhead of different sizes and structures of adders according to the principle of fig. 1b

In the following section we propose some improvements of this technique in order to further reduce the hardware cost.

## 6. Cost Reduction

The proposed technique is detecting transient faults by comparing the values present on an output at two instances distant by δ. The same detection capability will be achieved if the second instance is $t_0$-δ, as shown in figure 3. In this figure, a part of the outputs uses the principle of figure 1b (clocks Ck and Ck+δ), and another part uses the modified principle (clocks Ck and Ck-δ). The second principle can be used only for outputs that are ready early enough (i.e. having their maximum delay lower than or equal to Dmax-δ). To reduce the number of clock signals, we can use the normal clock for the Extra-Latch 1, as shown in figure 2.

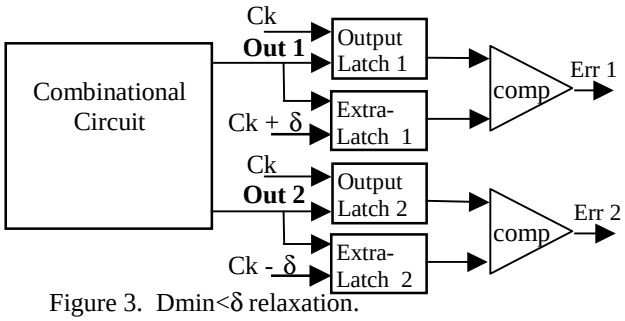



Figure 3. Dmin<δ relaxation.

The normal clock Ck can also be used to latch the input of the Extra-Latch 2 at the same time as for the Output-Latch (1,2), but a delay δ must be added at the input of this latch. This way, the actual value latched in Extra-Latch 2 is the value present at Out 2 at time $t_0$-δ.

Since the two latching instances of Out 2 differ by δ, the technique detects all transients with duration Dtr<δ-Dsetup. As concerning the detection of timing faults (delays lower than or equal to δ), we observe that the principle applied on Out 1 clearly detects them. As for Out 2, since its worst delay is less than Dmax-δ, by adding the faulty delay δ, we obtain a delay lower than Dmax. That is, the delay fault cannot affect the value used by the system (i.e. captured by Output-Latch 2).

The principle of figure 3 is interesting since it removes the constraint Dmin>δ, from the outputs with delays not exceeding Dmax-δ (Out2), and thus, it can be used for cost reduction.

A second cost-reduction principle is shown in figure 4. This principle has to be used for circuits with low *APO* because in this case, the Extra-Latch may represent a significant overhead and the principle shown in figure 4 will result on significant cost reduction.

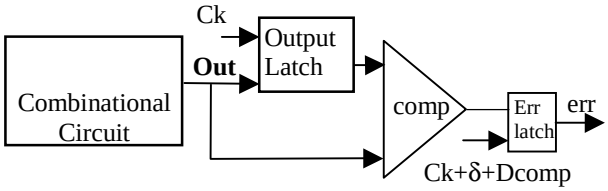



Figure 4. Extra-Latch removal

Figure 4 works as follows: consider the delay of the comparator to be Dcomp. The output value of the comparator is latched at time $t_0$+δ+Dcomp. This value is the result of the comparison of the values present on the inputs of the comparator at time $t_0$+δ. These values are on the one hand, the content of Output-Latch, which is holding the value presented on Out at $t_0$, and on the other hand the values present on Out at time $t_0$ + δ.

Thus, the scheme of figure 4 is equivalent to the scheme of figure 1b.

This scheme uses two clocks (Ck and Ck + δ + Dcomp). One can simplify it to use only one Ck for both Output-Latch and Error-Latch. In this case, the Error-Latch will be implemented to use the falling edge of the Ck as latching edge. In addition, a delay equal to $T_1$- δ - Dcomp has to be added on the second input of the comparator, where $T_1$ is the duration of the high level of the Ck. One can also use the rising edge as latching edge of Error Latch. In this case, the delay added on the comparator input will be T- δ - Dcomp, with T the period of Ck.

Note that this scheme requires that all paths between the inputs and the output of the comparator have similar delays. This will happen when the number of its inputs is a power of 2. If this is not the case, delay elements have to be added. This is illustrated in figure 5, when the comparator receives 6 input pairs. An element of delay equal to an OR gate was added to balance the circuit.

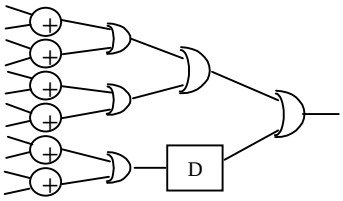



Figure 5. Comparator with balanced delays.

The two cost reduction principles are then used to reduce the cost for the case of multipliers considering δ=20%Dmax and for the case of adders. The last column of table 1 shows the obtained results for the multipliers applying the improved principle of figure 3. We remark that area cost is reduced for all multiplier cases. Tables 3 and 4 show respectively, the use of the principle of figure 3 and 4 in the case of adders.

It results a drastically reduction of the hardware overhead with respect to the results shown in table 2. The best results are obtained in table 4, excepting 14 adder cases (in gray in table 3). The columns of practical interest are the ones corresponding to δ=0.6 ns and δ=0.8ns (they will allow high detection efficiency for transient faults with a duration of several hundreds of ps), and the column corresponding to 20%Dmax, which is used for timing fault detection. We observe that the cost varies from 10.62% (Kogge&Stone 32 adder in column δ=20%Dmax, table 4) to 40.2% (CLA32 adder in column δ=20%Dmax). The cost is quite low for the Kogge&Stone adder, varying from 10.62% to 12.7% (column δ=20%Dmax) and from 14.19% to 18.87% (column δ=0.8ns). This is interesting because this is the faster adder implementation (see column CP-Critical Path). In addition, the transient fault detection efficiency for this kind of adders is high as illustrated in the next section.

| | CP (ns) | TAO(%) | | | | |
|---|---|---|---|---|---|---|
| | | δ=0.46 | δ=0.6 | δ=0.8 | δ=1 | 20% Dmax |
| **Brent&Kung 8** | 3 | 25.41 | 31.50 | 34.98 | 39.33 | 31.50 |
| **Brent&Kung 16** | 4.2 | 24.60 | 25.24 | 29.38 | 39.37 | 32.2 |
| **Brent&Kung 32** | 5.1 | 22.02 | 23.01 | 26.97 | 35.84 | 35.84 |
| **Han Carlson 8** | 2.8 | 24.29 | 28.31 | 34.65 | 45.87 | 26.16 |
| **Han Carlson 16** | 3.6 | 22.01 | 26 | 31.63 | 37.66 | 25.8 |
| **Han Carlson 32** | 4.2 | 18.64 | 20.02 | 23.50 | 33.71 | 25.2 |
| **Kogge&Stone 8** | 2.2 | 19.89 | 22.12 | 25.25 | 28.76 | 19.89 |
| **Kogge&Stone 16** | 2.5 | 19.16 | 20.1 | 23.66 | 27.84 | 19.8 |
| **Kogge&Stone 32** | 3.1 | 14.83 | 15.83 | 17.95 | 22.91 | 15.83 |
| **Sklanski 8** | 2.3 | 22.16 | 25.41 | 30.48 | 38.41 | 22.16 |
| **Sklanski 16** | 2.5 | 21.06 | 23.89 | 27.54 | 32.93 | 22 |
| **Sklanski 32** | 4.9 | 17.98 | 19.96 | 21.89 | 25.84 | 21.9 |
| **CLA 8** | 3.1 | 23.88 | 26.51 | 30.90 | 43.63 | 26.51 |
| **CLA 16** | 3.6 | 26.97 | 30.64 | 33.31 | 35.76 | 33.31 |
| **CLA 32** | 5.2 | 24.87 | 26.24 | 27.07 | 38.84 | 40.2 |

Table 3. Area overhead for adders according to the implementation of fig. 3

## 7. Transient fault injection simulations

The error detection efficiency of the scheme is verified by using several 8x8 adders and multipliers. Transient pulses of 0.65ns width have been injected. We have considered also the case of delay fault with a duration equal to 20% Dmax (1.4 ns) for a Wallace 8x8 multiplier.

| | CP (ns) | TAO(%) | | | | |
|---|---|---|---|---|---|---|
| | | δ=0.46 | δ=0.6 | δ=0.8 | δ=1 | 20% Dmax |
| **Brent&Kung 8** | 3 | 16.79 | 26.19 | 36.37 | 38.55 | 26.19 |
| **Brent&Kung 16** | 4.2 | 15.46 | 25.98 | 32.57 | 37.88 | 35.1 |
| **Brent&Kung 32** | 5.1 | 14.56 | 24.57 | 31.87 | 36.72 | 36.72 |
| **Han Carlson 8** | 2.8 | 15.7 | 25.64 | 33.47 | 37.91 | 21.19 |
| **Han Carlson 16** | 3.6 | 13.85 | 22.79 | 30.76 | 33.19 | 24.5 |
| **Han Carlson 32** | 4.2 | 12.31 | 21.44 | 25.81 | 32.72 | 27.1 |
| **Kogge&Stone 8** | 2.2 | 12.7 | 14.82 | 18.87 | 26.62 | 12.7 |
| **Kogge&Stone 16** | 2.5 | 11.84 | 12.73 | 16.18 | 22.21 | 12.2 |
| **Kogge&Stone 32** | 3.1 | 9.84 | 10.62 | 14.19 | 18.69 | 10.62 |
| **Sklanski 8** | 2.3 | 14.01 | 17.20 | 23.07 | 32.11 | 14.01 |
| **Sklanski 16** | 2.5 | 12.58 | 15.77 | 20.61 | 26.81 | 14.9 |
| **Sklanski 32** | 4.9 | 11.88 | 14.5 | 18.54 | 23.56 | 18.8 |
| **CLA 8** | 3.1 | 14.96 | 20.67 | 30.33 | 46.13 | 20.67 |
| **CLA 16** | 3.6 | 17.3 | 23.75 | 29.31 | 46.33 | 29.31 |
| **CLA 32** | 5.2 | 17 | 23.7 | 27.83 | 46.15 | 48.6 |

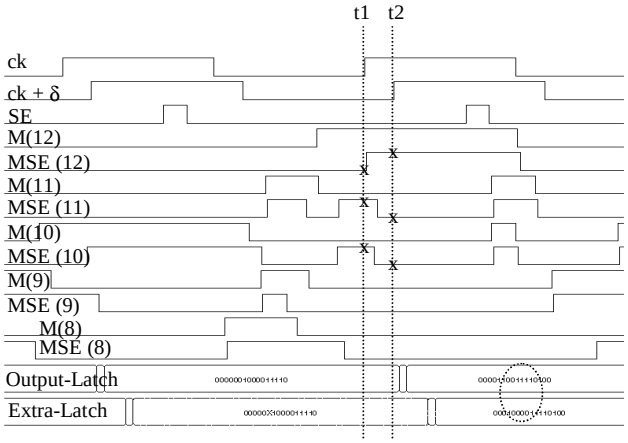
Table 4. Area overhead for adders according to the implementation of fig. 4

Since transient faults occur randomly, they can affect any node and occur at any instance during the clock cycle. In our simulations, the instance of the transient pulse injection is evenly distributed over the clock cycle. In fact the injection is repeated over several clock cycles by shifting within a clock cycle the instance of the injection at

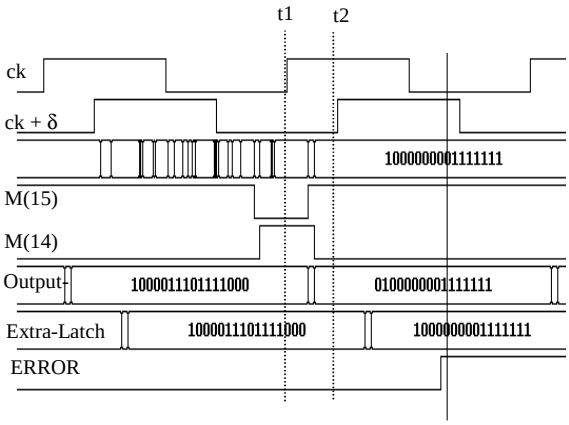
a 0.2 ns (or 0.1 ns in the case of adders). The injection locations were selected randomly, and simulations were performed for a large number of pseudorandom test patterns. On the total, 260000 simulations were performed on each selected circuit.

Fig. 6 presents several cases of the experiments done on a 8x8 Wallace multiplier. The rising edge of clock latches the inputs of the multiplier and also its outputs in Output-Latch (M signals in figure 6). The rising edge of ck + δ latches the multiplier outputs in Extra-Latch. A transient fault (SE signal on fig. 6) of 0.65ns width has been injected at the node computing the partial product 8. As explained above, the experiment is repeated several times with the transient pulse shifted of 0.2 ns in each clock cycle. This guarantees that the pulse will reach the circuit outputs at the moment of the latching edge of the clock for at least one clock cycle.

In the windows shown in fig. 6A, we can a case of transient fault detection. In this case, a transient fault is latched at instance $t_1$ by Output-Latch on three outputs of the multiplier (output M12, M11, and M10). The correct values are latched later, at $t_2$, by the Extra-Latch. This case corresponds at a transient pulse injected early in the clock cycle and for this case the errors are latched only by Output Latch. Another possible case corresponds to transient pulses latched only by the Extra Latch. It corresponds at pulses injected at the end of the clock cycle. It is also possible that the transient pulse is captured by Output Latch on a group of outputs, and by the Extra Latch on another group of outputs. In all cases, an error message is delivered at the output of the comparator. In all these cases the delayed clock used for Extra-Latch ensures that no error affecting the same outputs are latched in both Output-Latch and Extra-Latch.




A)




B)

Figure 6A. Transient fault injection logic simulation (M and MSE are the outputs of the fault-free and faulty multiplier, respectively); 6B -timing faults injection on the Wallace 8x8 multiplier

Note however that there is a possibility that an output error escapes detection. This will happen when the transient pulse width is amplified during its propagation through the circuit paths. The mechanisms creating this amplification are reconvergent fan-outs with different delays. They can propagate the original pulse through several paths, which reconverge and concatenate several pulses into a single one. This pulse can be larger than the original one due to the different delays of the propagating paths. Multiple pulses could also be created at the final node, but this will be a rare situation, since the delay difference between the paths should exceed the width of the original pulse. Due to this phenomenon, the scheme can not detect all output errors. We have computed the global efficiency of this scheme (ratio of errors occurring at the Output-Latch detected versus total number of the errors occurring in the Output-Latch).

For a transient pulse of a duration of 0.65 ns the experiments performed over 8x8 Wallace multiplier (260000 simulations) show an efficiency of 99.4% for δ = 0.8ns. The efficiency is increased to 99.7% if δ is increased by 0.1ns (δ = 0.9ns). We observe that the escapes are divided by two when δ is increased by 100ps. This shows that the efficiency can be increased significantly by slightly increasing δ.

The efficiency calculated for 8-bit adder structures, for 0.65 ns transient pulse duration, is 99.8% for Sklanski and Brent&Kung adder, 99.1% for Han and Carlson adder, and 100% for the other structures (CLA, Kogge&Stone). The results show that Kogge&Stone adder is the most advantageous adder structure for the present scheme (highest transient detection efficiency and lowest cost, in addition to its highest speed). Sklanski adder is the next structure of interest in terms of speed, detection efficiency and hardware overhead.

Figure 6 B presents the simulation of a timing fault in a 8x8 Wallace multiplier. The timing fault is injected by

adding a delay of 1.4 ns in a path segment leading to the outputs M14 and M15. The fault is activated by using an input transition that sensitizes the longest path of the multiplier. The fault creates a timing error on outputs M14 and M15 that are captured by Output-Latch. Due to the shifted clock used in Extra-Latch the timing fault disappeared when M14, M15 are latched in Extra-Latch. Thus, the output of the comparator detects the error. This simulation is presented to illustrate the detection mechanism for timing faults. This scheme achieves 100% detection of timing faults as far as the total delay added in any path of the circuit will not exceed $\delta$. For this reason, we do not need to use fault simulation to compute the efficiency of the scheme. As far as the total delay does not exceed $\delta$, the detection is guaranteed, independently to the mechanism creating the delay fault and the faulty delay distribution along the circuit paths.

## Conclusions

In the advent of very deep submicron and nanometer technologies, geometry shrinking, reduced power supply, and increasing operating speeds are making circuits increasingly sensitive to soft-errors and in particularly to single event upsets. At the same time, small delay deviation provoked by distributed or localized defects result on timing faults difficult to detect by fabrication testing. Due to these trends, it is becoming mandatory to design timing-error and soft-error tolerant ICs in order to maintain acceptable levels of reliability. Since this kind of solutions has to be used for any application, including commodity ones, traditional fault tolerant approaches are of reduced interest due to their high cost.

This paper considers a new scheme that exploits the temporal nature of timing errors and soft-errors in order to detect them by means of time redundancy. The scheme was experimented over various multiplier and adder structures. The experiments show that the new scheme requires low hardware cost and no performance penalty and achieves complete timing error and very high soft-error detection. Thus, the scheme is of high interest for achieving increasing robustness to timing errors and soft-errors and push aggressively the limits of technological scaling. The hardware cost is higher for adders, but it remains low for the more interesting (faster) adder cases. Further cost reduction could be obtained by obtained by combining the principles of figure 3 and 4 into a single circuit.

### REFERENCES


**[ALZ99]** I Alzaher-Noufal, M. Nicolaidis "A Tool for Automatic Generation of Self-Checking Multipliers Based on Residue Arithmetic Codes", 1999 DATE Conference, March 1999, Munich, Germany.

**[AVI 73]** AVIZIENIS A., "Arithmetic Algorithms for Error-Coded Operands" IEEE Trans. on Comput., Vol. C-22, No. 6, pp.567-572, June 1973.

**[BAZ97]** M. BAZE, S. BUCHNER, "Attenuation of Single Event Induced Pulses in CMOS Combinational Logic" IEEE Trans. on Nuclear Science, Vol. 44, No 6, December 1997.

**[BRE 82]** R. Brent and H. Kung, “A regular layout for parallel adders”, IEEE Transactions on Computers, vol. C-31, n 3, pp. 260-264, March 1982.

**[CHA 93]** H. CHA et al, “A Fast and Accurate Gate-level Transient Fault Simulation Environment”, Proceedings of FTCS, June 1993

**[FRA 94]** Franco P., McCluskey E. J., "On-Line Delay Testing of Digital Circuits", 12th IEEE VLSI Test Symposium, Cherry Hill, N.J., April 1994.

**[HWA 79]** K. Hwang, Computer Arithmetic, Principles, Architectures and Design, Jhon Wiley and Sons, New York, 1979

**[HAN 87]**T. Han, D. Carlson, “Fast area efficient VLSI Adders”, 8th Symposium on Computer Arithmetic, pp. 49-56, May 1987.

**[KOG 73]** P.M. Kogge and H. Stone, “A Parallel algorithm for efficient solution of a general class of recurrence equations”, IEEE Transactions on Computers, vol. C-22, n 8, pp. 786-792, August 1973

**[MET 98]** C. Metra, M. Favalli, B. Ricco, "On-Line Detection of Logic Errors due to Crosstalk, Delay, and Transient Faults", ITC October 18-23, 1998, Washington, DC.

**[NIC 99]** M. NICOLAIDIS, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies", In proceedings VTS 99.

**[PET 58]** PETERSON W.W. "On checking an Adder", IBM J. Res. Develop. 2, pp.166-168, April 1958

**[PET 72]** PETERSON W.W., WELDON E.J., "Error-Correcting Codes", second Ed., The MIT press, Cambridge, Massachusetts, 1972

**[SKL 60]** J. Sklanski, “Conditional-sum addition logic”, IRE Transaction on Electronic Computers, vol. EC-9, n 2, pp. 226-231, June 1960.

**[WES 94]** N.H.E. Weste and al. “Principles of CMOS VLSI Design- A System Perspective- second edition, Addison-Wesley Publishing Co., 1994.