Exploring the feasibility of selective hardening for combinational logic

S. N. Pagliarini\textsuperscript{a,}\textsuperscript{*}, G. G. dos Santos\textsuperscript{a,b}, L. A. de B. Naviner\textsuperscript{a}, J-F. Naviner\textsuperscript{a}

\textsuperscript{a}Institut Mines-Télécom/Télécom ParisTech, CNRS-LTCI UMR 514, COMELEC, Paris, France
\textsuperscript{b}Électricité de France, EDF R&D, Paris, France

Abstract

In this work we introduce a cost-aware methodology for selective hardening of combinational logic cells, which provides a list of the most effective candidates for hardening. Two heuristics are proposed in order to define when selective hardening becomes unfeasible. The methodology and the heuristics are applied to a set of benchmark circuits using costs extracted from an actual standard cell library. The results then show that both heuristics might be appropriate for different scenarios.

1. Introduction

Drastic device shrinking, increased complexity, power supply reduction, and increasing operating speeds that accompany the technological evolution to nanometric technologies have reduced dramatically the reliability of deep submicron ICs [1]. Due to these trends, the amount of defects and soft errors in electronic circuits is expected to increase, becoming a major concern in current and future technologies.

Several hardening techniques have been proposed over the years in order to increase the reliability of circuits. However, most of them consume too much area and/or energy to be cost-effective [2]. An attractive alternative is to protect only the most “error-sensitive” logic gates in the circuit [3, 4]. This technique, known as selective hardening, offers a straight trade-off between reliability and overhead, and therefore will be explored in this paper.

Selective hardening has been vastly investigated in the literature [2, 3, 4, 5, 6]. The difficulty to create a fault-tolerant design based on this technique is to find which is the best set of gates that has to be protected in order to meet a given reliability requirement. In [5], Polian et al. have described an algorithm with linear complexity in order to deal with this problem. However, in order to achieve a linear complexity, the authors have accepted the use of inaccurate values to estimate the relative reliability gains of hardening a given gate in a circuit. We have used the same premise in our work. Nevertheless, in this approach all gates are considered as having the same cost to be protected. In our approach, each type of gate has a different cost. For instance, it is quite less expensive to triplicate an inverter than to triplicate an OR gate with several inputs.

In this paper we make use of an efficient methodology for selective hardening of the most critical gates of a digital circuit [4]. This methodology takes into account the logical masking under the effect of multiple faults. A parameter similar to a hardening cost is also used in a cost function, which allows designers to drive the methodology using accurate cost values for hardening each gate. Nevertheless, in our previous work we have used this cost function with a constant improvement target \( T \) for the reliability of a given circuit (e.g., improve the reliability relatively by \( T = 10\% \)). In this paper we eliminate such constant target and propose two heuristics to determine when selective hardening is no longer feasible. Thus, the main contribution of this paper is to improve the methodology by presenting and studying the feasibility of these two heuristics.

This paper is organized as follows: Section 2 describes the methodology that classifies the blocks. An analysis of the profile of the cost function is then given in Section 3, together with two heuristics for evaluating the feasibility. Finally, in Section 4 we have applied the proposed methodology and heuristics to a set of circuits of different types and sizes, while in Section 5 we present our final remarks.
2. Selective Hardening Methodology

The reliability of a given circuit is the degree of confidence observed in the outputs of this circuit, given a scenario in which faults are expected to occur with a given probability. From the analysis point of view these faults could be either defects or transients induced by single event effects.

In this work we obtain the reliability figures of a circuit by applying the Signal Probability Reliability Analysis (SPRA) algorithm [7]. SPRA uses both the reliability of the gates and signal reliability computation to determine the cumulative effect of multiple faults, as shown in Fig. 1.

One of the particularities of the algorithm is that a signal reliability matrix is attributed to each signal in the circuit. It is assumed that a binary signal \( x \) can also carry incorrect information. This results in the fact that \( x \) can take four different values: correct zero (0\( _{c} \)), correct one (1\( _{c} \)), incorrect zero (0\( _{i} \)), and incorrect one (1\( _{i} \)). Then, the probabilities of occurrence of each one of these four values are represented in matrices, as shown below:

\[
P(x = 0_{c}) \quad P(x = 1_{c}) \quad P(x = 0_{i}) \quad P(x = 1_{i}) = \begin{bmatrix} x_0 & x_1 \ x_2 & x_3 \end{bmatrix}
\]

(1)

For instance, in Fig. 1 we have the inputs \( A_1 \) and \( B_4 \) represented as probabilities matrices. For the sake of simplicity, in this example both inputs are assumed to be always correct and evenly distributed. The goal then is to find the reliability of the output signal \( S \) given the gate fails with a probability \( q_{OR} \).

The propagation of the signal reliability through the logic gates is done by performing operations (tensoring and multiplying) to calculate the actual reliability of the entire circuit. Tensoring is performed in the inputs of the gates to guarantee that all combinations of inputs are considered. The result of the tensoring is then multiplied by the Probabilistic Transfer Matrix (PTM) of the gate, which is a different form of representation of a truth table. Such representation is responsible for inserting errors in the analysis (in this particular example the gate works properly 95% of the time). More details about the PTM representation and the algorithm itself can be found in [7].

The effort that is required to evaluate each gate of the circuit (in order to find the best hardening candidate) is only possible since the complexity of the SPRA algorithm is linear with the number of gates in the circuit.

Let us consider that a given circuit is comprised of \( K \) gates \( [g_1, \ldots, g_k] \). Each gate has an associated reliability, given by \( [q_{1}, \ldots, q_{k}] \). The circuit as a whole has a reliability value \( R \). Then, if we consider any reliability change (i.e., improvement) of a single gate \( g \), brings in its new reliability to \( q_{i}^{*} \), the circuit’s reliability becomes \( R^{*} \). Two different gates, \( g_{1} \) and \( g_{2} \), may have different contributions to the reliability of the circuit, therefore producing different values \( R^{*} \) and \( R^{*} \).

It is important to note that the SPRA algorithm is not 100% accurate. The sources of inaccuracies come from incorrect evaluation of (multiple) reconvergent fanout branches. An accurate analysis is possible using the multi-pass algorithm described in [7], referred as SPRMP. It is well known that both algorithms produce different values for the reliability of a circuit. Yet, in [4] we have shown that SPRA is capable of estimating the critical nodes (from a hardening perspective) with a small degree of error (in comparison with SPRMP).

In our methodology we assume that a hardening technique is applied, and such technique is able to improve the reliability of a gate such that \( q_{i}^{*} = 1 \). This is a simplification, not a restriction, other values are also possible. Then, for all gates of the circuit we perform an evaluation run of the SPRA algorithm. In each evaluation run we select a gate \( g_{i} \), allow \( q_{i}^{*} \) to be 1, and obtain the new reliability value \( R^{*} \).

After all evaluation runs are performed, we obtain an ordered list of \( R^{*} \) values. At this point one could elect the gate with the highest \( R^{*} \) to be hardened. This is a common approach applied in many works [8, 9]. Yet, this approach could be considered naive since it does not take into account the hardening cost of each gate. Both mentioned works define a maximum area target that cannot be surpassed.

Thus, our goal is to establish a trade-off between the cost of hardening a gate against the cost of hardening any other gate. In order to do so, we introduce a new parameter to express the hardening affinity of a gate, given by \( Cha_{i} \). This parameter defines how easy/hard it is to harden a gate. The \( Cha_{i} \) value of each gate type is user-defined and it must be constrained in the interval [0,1]. The higher the value of \( Cha_{i} \), the better. This parameter...

Fig. 1: SPRA of an OR gate.
is generic and can be used to express any type of hardening trade-off: area, delay, power or combinations of the previous. The decision of which circuit characteristic should be used to define $Cha_i$ falls to the user.

In Tab. 1 we show some of the values that were used in our experiments. These are extracted from an actual 90nm standard cell library provided by Synopsys [10]. In our experiments we considered only the area to calculate the hardening affinity. For each gate we have divided the area of the smallest inverter (INVX0) in the library by the given gate actual area, in order to normalize all the $Cha_i$ values. Negated cells benefit from the CMOS natural inversion and have a higher $Cha_i$ value.

<table>
<thead>
<tr>
<th>Cell</th>
<th>Area (µm²)</th>
<th>$Cha_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVX0</td>
<td>5.5296</td>
<td>1</td>
</tr>
<tr>
<td>NAND2X0</td>
<td>5.5296</td>
<td>1</td>
</tr>
<tr>
<td>NOR2X0</td>
<td>5.5296</td>
<td>1</td>
</tr>
<tr>
<td>AND2X1</td>
<td>7.3728</td>
<td>0.75</td>
</tr>
<tr>
<td>OR4X1</td>
<td>10.1376</td>
<td>0.55</td>
</tr>
<tr>
<td>XOR3X1</td>
<td>22.1184</td>
<td>0.25</td>
</tr>
</tbody>
</table>

It is then possible to apply the $Cha_i$ values in a cost function which takes into account both the cost and the reliability gain. The reliability gain (or reliability difference) is given by $R_{gi}$, and it is the difference from the circuit reliability before and after a single gate $g_i$ was hardened:

$$R_{gi} = R_i^* - R_i$$

(2)

These values are then used in a cost function that is expressed as follows:

$$C_i = \frac{R_{gi}}{Cha_i}$$

(3)

Once the value of $C_i$ has been obtained for all gates, these are sorted and the highest value is chosen. The gate that corresponds to the highest value of $C_i$ is then assumed to be hardened and the new circuit reliability ($R_i^*$) is obtained.

In [4] this reliability value is compared against a user-given reliability target. If it is lower than the target the algorithm starts again and all gates still not hardened are considered as candidates. Otherwise, if the target is met, the algorithm ends and outputs the ordered list of gates to be hardened. In the next section we will define two heuristics in order to avoid for the user to set a reliability improvement target.

3. Cost Function Profiling

The methodology described in Section 2 was applied to several ISCAS benchmark circuits [11]. The profile of the cost function was then obtained for circuits of different sizes and topologies. Figures 2 and 3 illustrate the cost function profile for the circuits c432 (a channel interrupt controller) and c499 (32-bit single-error-correcting circuit). These circuits were chosen particularly because they represent two contrastive profiles that are of interest.

The illustrations in both figures were obtained using the parameters $q_i = 0.999$ and $q_i^* = 1$. Other combination of values cause slight changes in the plots, i.e., the profile of the function remains the same. In other words, the profile of the function is highly related to the logic masking capabilities and the affinity of each gate. The closer a gate is to the $y$ axis, the better candidate for hardening it is.

The illustration in Fig. 2 represents a profile that contains a fast drop in the function, observed in the very first gates. Circuits that have some degree of regularity (e.g., adders and multipliers) have a profile with some similarities with the one in Fig. 3, where a ‘step-like’ pattern is observed. Each ‘step’ or plateau represents a set of gates that has a similar functionality in the circuit, therefore they can be hardened in any given order. Taking into account both profiles that were presented, we have defined two heuristics to decide when selective hardening starts to impose an impractical cost. Those heuristics are explained in details in the next subsections.

![Fig. 2: Cost function profile for the circuit c432.](image)
3.1. Sum of elements heuristic

This heuristic was defined to create a stop point when the sum of the \( C_i \) terms from the elements that were already hardened reaches a threshold. Let \( C_0 \) be the value of the cost function for the best hardening candidate. Then the target becomes to find a value \( j \) such that:

\[
\sum_{i=2}^{j} C_i \leq K \times C_0
\]

where \( K \) is an empirically chosen constant. In other words, the threshold is defined as \( K \) times the value of the cost function for the first hardened gate. This heuristic can be interpreted as an integral that sums the area under a curve. For the sake of comparison, in the text that follows we have empirically set the parameter \( K = 10 \).

3.2. Percent wise heuristic

This heuristic was defined to create a stop point at the first \( C_i \) value that is lower than \( X\% \) of the first term (\( C_0 \)). This heuristic can be interpreted as an horizontal threshold value. When the function crosses that threshold it is no longer feasible to perform selective hardening for the remaining gates.

For the sake of comparison, in the text that follows we have empirically set the parameter \( X = 50\% \). In other words, any gate that improves the circuit reliability with a \( C_i \) value that is less than half of \( C_0 \) should not be hardened, i.e., we only harden cells that are at least half as effective as the first candidate.

3.3. Comparing the heuristics

Both heuristics were applied to the circuit \( c1355 \) (which is also a 32-bit single-error-correcting circuit). Figure 4 contains the plot of the cost function for all elements of the target circuit. The dashed vertical lines represent the points where the heuristics decided that selective hardening was no longer feasible.

Deciding which parameter value is more appropriate for each circuit is a complex task. For instance, for the circuit \( c1355 \), the first heuristic would select 11 gates for hardening, while the second heuristic would select 201 gates. Hardening 201 out of 546 gates (around 36%) might be a hard assignment, since most of the times the area budget will not allow for such hardening (the total circuit area would become 76% larger).

Nevertheless, selecting 11 out of 546 gates (around 2%) might be a better and more suitable choice. Along the same lines, applying the percent wise heuristic to the circuit \( c432 \) would result in only 2 gates being selected for hardening, which could left some of the hardening budget unused.

In the next section we present the results for other circuits and we also extend the discussion regarding which heuristic (and associated parameter) is more appropriate for which scenario.

4. Experimental Results

The methodology described in Section 2 was applied to several ISCAS benchmark circuits. Each gate from each circuit was set using \( q_i = 0.9999 \). The results are presented in tables 2 and 3. The former table contains the results for the first heuristic defined in Subsection 3.1 (with \( K = 10 \)) while the latter contains the results for the second heuristic defined in Subsection 3.2 (with \( X = 50\% \)).

In tables 2 and 3, the meaning of each column is as follows: the column denoted “Original area” contains the sum of the area from each gate in each circuit (therefore placement utilization rate and routing overhead are not considered). The column denoted “Hardened gates” contains the amount of gates that are selected for hardening. Then, the column denoted “Hardened area” contains the circuit area of from the hardened version of the circuit, while the column denoted “Area increase” contains that same value but percent wise.

A fairly simple assumption was made: when hardening a gate its area become three times larger than before. This metric is inspired by classical Triple Modular Redundancy (TMR), although other techniques with different metrics might be applied (e.g., hardening by design). The additional area that would be required for a majority voter, given TMR is considered, is neglected. Therefore the area figures given in the tables are a minimum value estimate for TMR.

An analysis of the area increase values in Tab. 2 reveals that the sum of elements heuristic is not prone for
Fig. 4: Both heuristics applied to the circuit c1355.

Table 2
Results for the sum of elements heuristic, $K = 10$.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Number of gates</th>
<th>Original area ($\mu m^2$)</th>
<th>Hardened gates</th>
<th>Hardened area ($\mu m^2$)</th>
<th>Area increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>c17</td>
<td>6</td>
<td>33.1776</td>
<td>6</td>
<td>99.5328</td>
<td>200%</td>
</tr>
<tr>
<td>74283</td>
<td>40</td>
<td>306.5096</td>
<td>20</td>
<td>547.9688</td>
<td>78.7%</td>
</tr>
<tr>
<td>c432</td>
<td>160</td>
<td>1134.4672</td>
<td>33</td>
<td>1541.4208</td>
<td>35.8%</td>
</tr>
<tr>
<td>c499</td>
<td>202</td>
<td>2155.1680</td>
<td>12</td>
<td>2414.1504</td>
<td>12.0%</td>
</tr>
<tr>
<td>c1355</td>
<td>546</td>
<td>3194.7328</td>
<td>11</td>
<td>3316.3840</td>
<td>3.8%</td>
</tr>
<tr>
<td>c1908</td>
<td>880</td>
<td>5273.7488</td>
<td>13</td>
<td>5417.5184</td>
<td>2.7%</td>
</tr>
<tr>
<td>c2670</td>
<td>1269</td>
<td>8018.0632</td>
<td>19</td>
<td>8233.7176</td>
<td>2.6%</td>
</tr>
<tr>
<td>c3540</td>
<td>1669</td>
<td>10855.1824</td>
<td>25</td>
<td>11177.7424</td>
<td>2.9%</td>
</tr>
<tr>
<td>c5315</td>
<td>2307</td>
<td>15293.5992</td>
<td>20</td>
<td>15518.4696</td>
<td>1.4%</td>
</tr>
</tbody>
</table>

Table 3
Results for the percent wise heuristic, $X = 50%$.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Number of gates</th>
<th>Original area ($\mu m^2$)</th>
<th>Hardened gates</th>
<th>Hardened area ($\mu m^2$)</th>
<th>Area increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>c17</td>
<td>6</td>
<td>33.1776</td>
<td>5</td>
<td>88.4736</td>
<td>166.6%</td>
</tr>
<tr>
<td>74283</td>
<td>40</td>
<td>306.5096</td>
<td>9</td>
<td>406.0424</td>
<td>32.5%</td>
</tr>
<tr>
<td>c432</td>
<td>160</td>
<td>1134.4672</td>
<td>2</td>
<td>1187.5264</td>
<td>4.6%</td>
</tr>
<tr>
<td>c499</td>
<td>202</td>
<td>2155.1680</td>
<td>41</td>
<td>2854.6752</td>
<td>32.4%</td>
</tr>
<tr>
<td>c1355</td>
<td>546</td>
<td>3194.7328</td>
<td>201</td>
<td>5647.1232</td>
<td>76.7%</td>
</tr>
<tr>
<td>c1908</td>
<td>880</td>
<td>5273.7488</td>
<td>119</td>
<td>6611.9112</td>
<td>25.3%</td>
</tr>
<tr>
<td>c2670</td>
<td>1269</td>
<td>8018.0632</td>
<td>10</td>
<td>8128.6552</td>
<td>1.4%</td>
</tr>
<tr>
<td>c3540</td>
<td>1669</td>
<td>10855.1824</td>
<td>8</td>
<td>10963.9312</td>
<td>1.2%</td>
</tr>
<tr>
<td>c5315</td>
<td>2307</td>
<td>15293.5992</td>
<td>15</td>
<td>15459.4872</td>
<td>1.1%</td>
</tr>
</tbody>
</table>
small circuits, causing a large overhead for the circuits 74283 and c432. For the smallest of the circuits (c17) the heuristic decides that all gates should be hardened, which is unacceptable when the goal is selective hardening. Nevertheless, this can be avoided by using a smaller value for the parameter $K$ (e.g., $K = 1$ elects 2 cells while $K = 2$ elects 4 cells for hardening). This is not the case for the area increase values in Tab. 3. There is no value for the parameter $X$ that will be a good fit for all circuits or even for a group of circuits. Therefore, it is quite harder to apply the percent wise heuristic.

4.1. Comparison with related works

A straightforward comparison with other methodologies is not simple since the goals are usually different. If comparing a methodology is hard, it is even harder to compare the heuristics proposed in this paper. A simple solution adopted by related works is to define a limit or target for hardening. In [5] a simple limit $L$ is defined as the maximum number of gates to be hardened. In both [8] and [9] a hardening limit in terms of area increase is applied. In [4] we have defined the hardening target as a relative improvement in the reliability of the circuit. None of the mentioned works perform an evaluation of how hard it was to reach a hardening limit/target. This is the reason why we have studied the profile of the cost function.

The work of [12] has a similar target as the one described in this paper. The hardening is achieved by increasing the gate size of some critical nodes in the circuit but no hardening against defects is mentioned. Although this is a valid solution, it can be quite complicated to apply it in a commercial design flow since the choices of logic gates are limited. Thus the technique presented here is more of a general solution since it is library and technology independent. The overheads mentioned in [12] are not directly comparable.

Nevertheless, in qualitative terms it is easily observed that certain gates have a larger impact in the reliability of the circuit than others. This observation is highlighted in [5, 8, 9, 12]. In our experiments this was also observed. There are some particular cases, like the one illustrated in Fig. 2, where choosing the correct gate to harden has a large impact in the overall circuit reliability.

5. Conclusion

In a context where defects and soft errors are a growing concern, we have proposed two heuristics that provide a better understanding of the costs related to selective hardening applied for combinational logic in digital circuits. Furthermore, we have also concerned multiple faults to determine the actual reliability of the circuits in our analysis.

Our results present the use of the methodology in conjunction with a standard cell library from an actual vendor, where the trade-off between area and reliability gain is highlighted. Thus, the methodology can be integrated in commercial design flows in a very straightforward manner. We have also presented two heuristics. The heuristic described in Subsection 3.1, entitled ‘Sum of elements’, is of straightforward use.

Acknowledgment

This work was partially funded by the CATRENE project RELY.

References