A Hybrid Fault Tolerant Architecture for Robustness Improvement of Digital Circuits

D. A. Tran  A. Virazel  A. Bosio  L. Dilillo
P. Girard  S. Pravossoudovitch
LIRMM – University of Montpellier / CNRS
Montpellier, France
{tran, virazel, bosio, dilillo, girard, pravo}@lirmm.fr

H.-J. Wunderlich
Institut für Technische Informatik
Stuttgart, Germany
wu@informatik.uni-stuttgart.de

I. INTRODUCTION

CMOS technology scaling allows the realization of more and more complex systems, reduces production costs and optimizes performances and power consumption. Today, each CMOS technology node is facing reliability problems [1] whilst there is currently no alternative technology as effective as CMOS in terms of cost and efficiency. Therefore, it becomes essential to develop methods that can guarantee a high robustness for future CMOS technology nodes.

To increase the robustness of future CMOS circuits and systems, fault tolerant architectures might be one solution. In fact, these architectures are commonly used to tolerate on-line faults, irrespective of their transient or permanent nature [2]. Moreover, it has been shown in [3, 4, 5] that they could also tolerate permanent defects and thus help improving the manufacturing yield.

Various solutions using fault tolerant techniques for robustness improvement have been studied, of which they target first and foremost the tolerance of transient and/or permanent faults. Here for the first time, our study provides a fault tolerant architecture that targets different goals at the same time. Firstly, it increases circuit robustness by tolerating both transient/permanent online faults and manufacturing defects. Secondly, it is able to save power consumption compared to existing solutions. Finally, it deals with aging phenomenon and thus, increases the expected lifetime of logic circuits.

The remaining parts of this paper are organized as follows. Section II provides the principle as well as the functioning of the hybrid fault tolerant architecture. Comparisons with the TMR approach in terms of area and power consumption are discussed in Section III. Section IV analyzes impacts of our architecture on aging phenomenon. Finally, Section V concludes the paper.

II. THE HYBRID FAULT TOLERANT ARCHITECTURE

As solutions for robustness improvement of sequential elements can be found in the literature such as razor registers [6, 7], this paper targets only robustness improvement of combinational part of circuits.

Our new hybrid fault tolerant architecture uses three types of redundancy: information redundancy for error detection, temporal redundancy for transient error tolerance and hardware redundancy for permanent error correction. The following subsections presents the principle and the possible configurations of the architecture.

A. Principle

Figure 1 shows the functional scheme of our hybrid architecture. The logic circuit is implemented three times (LC1, LC2, LC3) but only two of them are working in parallel and are selected with the help of two multiplexors (MUX_IN, MUX_OUT). The third logic circuit is normally in standby state. The comparator verifies the good functioning of the current configuration by comparing outputs of the two running logic circuits. Its output (Ok signal) controls the enable input of the registers. During fault free operations, the Ok signal is true and the current configuration does not change. As long as no error is detected, only two circuits are running. If the comparator detects an error, the Ok signal becomes false and the registers are disabled. The Finite State Machine (FSM) changes the configuration to tolerate the detected error by controlling the multiplexors.

B. Configurations

As mentioned above, the FSM manages the configuration of the architecture by selecting a couple of circuits to run in parallel. When an error is detected, two tolerant schemes are possible:

- FSM1: the FSM does not change the configuration and the two running circuits re-compute the same input data. If the error still remains at the second computation, the FSM changes the configuration. This solution puts priority in the tolerance of transient errors and requires more time for tolerating permanent faults.

- FSM2: the FSM changes the configuration each time an error is detected. This solution focuses on tolerating permanent faults and needs more time for tolerating transient faults.
III. COMPARISONS WITH THE TMR ARCHITECTURE

In order to evaluate the architecture, we compare it with the classical TMR solution in terms of area and power consumption. Logic circuits used in these comparisons are ISCAS’85 and combinational parts of ISCAS’89 and ITC’99 benchmark circuits.

In this sub-section we compare TMR and the hybrid fault tolerant architecture in terms of silicon area and power consumption. For the power comparison, both architectures were synthesized using a 90nm technology with RTL Compiler™ [8]. Then, the power consumption of each architecture was evaluated with NanoSim™ [9]. For the area comparison, we use the transistor count method which makes results independent of the targeted technology. Results are presented in Table I.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>n</th>
<th>m</th>
<th>N_LC</th>
<th>N_TMR</th>
<th>N_HFT</th>
<th>AO</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>c3515</td>
<td>178</td>
<td>123</td>
<td>4183</td>
<td>18977</td>
<td>20509</td>
<td>8%</td>
<td>16%</td>
</tr>
<tr>
<td>c6288</td>
<td>32</td>
<td>32</td>
<td>8846</td>
<td>28010</td>
<td>28531</td>
<td>2%</td>
<td>36%</td>
</tr>
<tr>
<td>c7552</td>
<td>206</td>
<td>107</td>
<td>4960</td>
<td>21188</td>
<td>23026</td>
<td>9%</td>
<td>20%</td>
</tr>
<tr>
<td>s15850</td>
<td>611</td>
<td>684</td>
<td>9851</td>
<td>59995</td>
<td>63556</td>
<td>6%</td>
<td>8%</td>
</tr>
<tr>
<td>s35932</td>
<td>1763</td>
<td>2048</td>
<td>25976</td>
<td>168146</td>
<td>177533</td>
<td>6%</td>
<td>11%</td>
</tr>
<tr>
<td>s38417</td>
<td>1664</td>
<td>1742</td>
<td>27717</td>
<td>162191</td>
<td>171706</td>
<td>6%</td>
<td>10%</td>
</tr>
<tr>
<td>s38584</td>
<td>1464</td>
<td>1730</td>
<td>34546</td>
<td>179494</td>
<td>187249</td>
<td>4%</td>
<td>9%</td>
</tr>
<tr>
<td>b14a</td>
<td>277</td>
<td>299</td>
<td>13328</td>
<td>53430</td>
<td>55267</td>
<td>3%</td>
<td>25%</td>
</tr>
<tr>
<td>b15s</td>
<td>485</td>
<td>519</td>
<td>27347</td>
<td>105439</td>
<td>108416</td>
<td>3%</td>
<td>21%</td>
</tr>
<tr>
<td>b17s</td>
<td>1452</td>
<td>1512</td>
<td>81557</td>
<td>313383</td>
<td>321756</td>
<td>3%</td>
<td>20%</td>
</tr>
<tr>
<td>b18</td>
<td>3357</td>
<td>3342</td>
<td>210655</td>
<td>785907</td>
<td>805331</td>
<td>2%</td>
<td>22%</td>
</tr>
<tr>
<td>b19</td>
<td>6666</td>
<td>6669</td>
<td>424235</td>
<td>1579437</td>
<td>1617563</td>
<td>2%</td>
<td>21%</td>
</tr>
<tr>
<td>b20s</td>
<td>522</td>
<td>512</td>
<td>27397</td>
<td>105883</td>
<td>100126</td>
<td>3%</td>
<td>24%</td>
</tr>
<tr>
<td>b21s</td>
<td>522</td>
<td>512</td>
<td>28523</td>
<td>109261</td>
<td>112594</td>
<td>3%</td>
<td>26%</td>
</tr>
<tr>
<td>b22c</td>
<td>767</td>
<td>757</td>
<td>42330</td>
<td>161952</td>
<td>166674</td>
<td>3%</td>
<td>26%</td>
</tr>
</tbody>
</table>

In Table I, the three first columns present respectively the name (Circuit), the number of input (n) and the number of output (m) of each LC. The three next columns show the transistor count of the LC (N_LC), of the TMR architecture (N_TMR) and of the hybrid architecture (N_HFT). The seventh column (AO) gives the area overhead of our architecture with respect to the TMR architecture. Finally, the last column (PR) gives the power reduction achieved with our architecture compared to the TMR implementation.

As shown in Table I, the proposed solution for robustness improvement has a comparable cost to the TMR solution since the area overhead is about 2% to 3% for the largest considered benchmark circuits. Moreover, most of the time, the architecture save more than 20% of power consumption compared to TMR except for ISCAS’89 benchmark circuits. In fact, these circuits have many more inputs/outputs than other circuits of the same size. Consequently, for these circuits, the consumption of the logic part does not dominate the overall architecture power consumption. Therefore, the fact that only two LCs are running instead of three does not reduce the power consumption as expected.

IV. IMPACT OF THE HYBRID FAULT TOLERANT ARCHITECTURE ON AGING PHENOMENON

In this section we discuss the ability of the hybrid architecture to deal with aging phenomenon. In fact, since only two LCs are running, the remaining one does not compute any data and hence has no activity. Consequently, for a fault free functioning, the two running circuits are those that suffer the most from the aging phenomenon. The one in standby mode normally will have a higher expected aging time and may even recover from previous activity.

Our architecture must be modified in a way to balance the using time period of each LC. This can be done by modifying the FSM in a way to change the configuration periodically using one of the following methods:

- **Time:** The configuration is changed after a certain number of fault-free clock periods. This solution requires a simple counter.
- **Pattern:** The configuration is changed each time specific input patterns are applied. This solution requires a small memory to store these patterns.

V. CONCLUSION

In this paper, we have proposed a hybrid architecture to improve the robustness of logic CMOS circuits. This architecture combines different types of redundancy to tolerate transient as well as permanent faults: information redundancy for error detection, temporal redundancy for transient error correction and hardware redundancy for hard error tolerance. Adding only 2% to 3% of area compared to TMR, the hybrid architecture can save about 24% of power consumption for largest benchmark circuits. In addition, it has been shown that its expected lifetime will be longer than that of TMR fault tolerant structure.

REFERENCES