Analysis of the Equivalences and Dominances of Transient Faults at the RT Level

Luis Berrojo, Isabel González, Luis Entrena, Celia López, Fulvio Corno, Matteo Sonza, Giovanni Squillero
Alcatel Espacio
Madrid, Spain
Universidad Carlos III de Madrid
Madrid, Spain
Politécnico di Torino
Torino, Italy

Abstract
This work presents a study for tackling transient faults at the RT-level and outlines the techniques devised and implemented to speed-up fault-injection campaigns, detecting the equivalences and dominances between faults in order to collapse them. Experimental results are provided on an industrial case study, demonstrating the effectiveness of the approach.

1. Introduction
In this poster we outline a set of techniques devised and implemented, within a fault-injection platform, to speed-up fault-injection campaigns. This research is performed in the context of the European IST project AMATISTA (#11762), whose main target is the development of a set of tools for the design of fault tolerant circuits at RT-level. Fault-injection campaign grades possible faults (SEUs), classifying them as Failure, Silent, Latent or Error, [1]. Fault-collapsing techniques can reduce notably the amount of faults to be simulated.

2. Fault Collapsing

WorkLoad Independent Fault Collapsing techniques are uniquely based on the analysis of the design. Circuit topology study will help us to know the controllability and observability of flip-flops, and to determine dominances and equivalences in faults injected. Some rules are set.

RULE 1. All faults in flip-flops whose output is connected directly to primary outputs, are detected.
RULE 2. All faults in flip-flops (A) whose output is connected directly to another flip-flop (B) are equivalent to faults of the connected flip-flop (B).
RULE 3. Faults affecting shift registers can be categorised by analysing the faults in last flip-flop of the register, if its behaviour is serial.

WorkLoad Dependent Fault Collapsing techniques are based on the analysis of fault-free simulation. It is possible to classify faults in flip-flops, according to equivalences and dominances in the time domain. We have adapted Life Periods theory, [2], and 3 rules are set.

RULE 1. All faults between two write operations are marked as silent, because their effect will be masked before any possible propagation.

RULE 2. All faults between a write operation and the subsequent read operation are equivalent, as their effect can only be propagated on the read operation.
RULE 3. All faults between a read operation and the subsequent read or write operation comply with the following dominance relation: if a fault is proven silent, all SEUs that activate later can be categorized as silent. Also if a fault is proven failure, all faults that activate earlier can be categorized as failure.

Dynamic Fault Collapsing technique is exploited to discover equivalencies between faults during fault simulation by comparing temporal results.

3. Experimental results
Optimization techniques proposed have been applied to a real industrial design. Solar Array Drive Electronics (SADE) is a module developed by Alcatel Espacio that will be hosted on satellites. The number of faults obtained for SADE circuits are 15M fault approx. The static fault collapsing phase causes SADE fault list to be collapsed to about 13% of its initial size. Then, dynamic equivalencies prevent simulation of 4,73% of faults.

4. Conclusions
A complete analysis on Equivalences and Dominances of Transient Faults at RTL has been presented. Experimental results show the efficacy of fault-collapsing techniques considered. Exploiting these techniques, dramatic reductions in the number of SEUs to be injected are obtained allowing feasible fault injection campaigns. For the first time a full categorization of all faults has been done for a medium size design in RTL.

The evaluation with an industrial design have shown the effectiveness of the proposed fault injection platform. The integration of the developed environment into commercial design flows will increase the usage of fault tolerant technology in application sectors where fault-tolerant was not a common and automated task.

5. References