OCD-FI: On-Chip Debug and Fault Injection

André V. Fidalgo[1,2], Gustavo R. Alves[1], José M. Ferreira[2]
[1]*Instituto Superior de Engenharia do Porto*
[2]*Faculdade de Engenharia da Universidade do Porto*
anf@isep.ipp.pt gca@isep.ipp.pt jmf@fe.up.pt

Abstract

This paper presents a set of modifications to common processor on-chip debugging infrastructures to support the execution of fault injection campaigns. The proposed solution is applicable to different target architectures and imposes a very low logic overhead, providing a flexible and efficient mechanism for verifying and validating fault tolerant components.

1. Introduction

Dependable systems are designed to handle errors that originate from software or hardware faults and recover from them, maintaining acceptable operating conditions. A common mean to achieve dependability is the use of fault-tolerant components and the verification of their correct behavior is often performed using fault injection techniques [1]. Most of today's microprocessors provide access to internal resources through dedicated built-in debug circuitry, often designated as on-chip debug (OCD), which can also be used for fault injection purposes [2]. This paper presents a proposal to improve the fault injection process, in terms of performance and triggering capabilities, by using a specialized debugger and also migrating some of the required fault injection operations to the inside of the OCD infrastructure.
All debugging components were designed to be NEXUS [3] compliant, both to benefit from the useful features defined in the proposed standard and to increase the area of immediate applicability of the proposed concepts and solutions.

2. Case Study

2.1 The OCD-FI concept

The On-Chip Debug and Fault Injection (OCD-FI) concept proposed on this paper consists of additional hardware included in the target device in order to automatically insert faults on the occurrence of a triggering condition. To minimize overhead, this fault injection (FI) module is integrated within the OCD circuitry reusing some of the already implemented debugging functions. It requires the OCD to provide some type of watchpoint or breakpoint support and also the ability to read and write on the target memory elements. After an initial set-up, the FI module will wait for a watchpoint (or breakpoint) to activate the writing of a faulty value into the target memory. The fault-free value at the moment of the injection must be determined beforehand, using either knowledge of the running application code or an initial golden-run up to the fault injection instant. All OCD resources related to program trace must be operational during the entire process to allow the reconstitution of the program flow.

2.2 Fault Injection Environment

The fault injection environment is presented in Figure 1 and is composed of the target system, a debugger to manage the fault injection process and a host running the experiment.

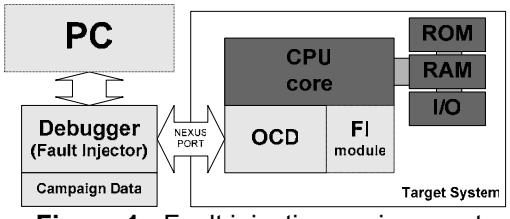

Figure 1 - Fault injection environment

The target microprocessor cores used were developed using the *cpugenerator* [4] building tool which automatically creates configurable CPU cores in VHDL format. The OCD infrastructure was defined as a customizable model that can be adapted to each particular target. The debugger is oriented for fault injection campaign management and includes all common debugging functions, plus some dedicated to

fault injection operations. It communicates with the OCD through a NEXUS port using full-duplex messaging, and operates by reading scripts stored in memory and recording trace data from the target (via the OCD). Moreover, if the target system is implemented on a FPGA the debugger can also be included on the same device and later discarded.

All experiments were structured into fault injection campaigns, each one defining a set of fault injection operations where a specific fault location and trigger condition were selected. The selected fault model is used in most common fault scenarios for microprocessor based critical systems [5] and consists of single bit-flip faults in random memory elements at random moments during the application execution. The fault location can be any resource accessible for writing through the OCD, including memory, internal registers and stack. These fault injection campaigns can be used for experimental evaluation of the target device fault tolerant characteristics, as indicated by the analysis of the preliminary results.

2.3. Preliminary Results

The target system, the debugger, the fault injection module and the different memories were synthesized with the ISE 7.1i development environment and simulated using the Modelsim 6.0a simulation engine. Each fault injection campaign was executed using the OCD alone first, and then repeated using the OCD-FI infrastructure. Table 1 compares both approaches.

Table 1 – Main features comparison

	OCD	OCD-FI
Fault Triggering	Breakpoint or watchpoint hit	
Communication of Trigger	Message from OCD to Debugger	Signal from OCD to FI module
Fault Activation	Message from Debugger to OCD	Signal from FI module to OCD
Fault Injection	Write operation by the OCD	
Equivalent Gates (8 bit version) (32 bit version)	6985 18876	7060 (+ 1,1%) 18951 (+ 0,4%)
Activation Delay (in clock cycles)	14 to 21	2

From the executed experiences, we extracted the following conclusions:

- Both approaches allow the injection of faults in memory space without halting processor execution.
- When only using the OCD (alone), about 2% of the experiments return meaningless results due to CPU writes during the fault injection process. This did not happen with the OCD-FI (as the activation delay is reduced by an order of 85%).
- The OCD-FI infrastructure does not affect the maximum microprocessor clock frequency.

3. Conclusions

From the preliminary results it is possible to conclude that the proposed OCD-FI infrastructure has the potential to become an efficient mechanism for verifying and validating the fault tolerance characteristics of microprocessor based systems. When compared with other alternatives, it provides an efficient methodology for fault injection, both in terms of reusability, resource coverage, performance and cost. The FI module main advantage is the possibility of injecting precisely controlled faults on CPU memory elements without stopping the program execution and without changing the target software or hardware. Major benefits are the minimum delay that minimizes the risk of the CPU accessing the target memory during the fault injection process and the possibility of using it in the simulation phase, prototyping phase or on the final device.

The compliance with the NEXUS proposed standard provides a common basis for the development and enhancement of the proposed methodology. In this sense, the OCD-FI concept can easily be extended to any NEXUS compliant microprocessor and even to other architectures as the more complex functions are performed by the OCD infrastructure. As a downside, there is the need of an adequate OCD infrastructure and the ability to modify it (to include the FI module).

The ongoing work is focused on verifying the applicability of this fault injection approach to different fault tolerance solutions and microprocessor architectures.

5. References

[1] A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing”, IEEE Transactions on Dependable and Secure Computing, Volume 1, Issue 1, Jan 2004.

[2] P. Yuste , D. de Andrés , L. Lemus , J.J. Serrano, P.J. Gil, “INERTE: Integrated NExus-Based Real-Time Fault Injection Tool for Embedded Systems”, The International Conference on Dependable Systems and Networks, San Francisco, USA, June 2003.

[3] “The Nexus 5001 Forum Standard for a Global Embedded Processor Interface version 2.0”, IEEE-ISTO 5001, 2003.

[4] Giovanni Ferrante, “CPUGEN 2.00”, 2003.

[5] R. Velazco, R. Ecoffet, F. Faure, “How to characterize the problem of SEU in processors & representative errors observed on flight”, 11th IEEE International On-Line Testing Symposium, Saint Raphael, France, July 2005.