

A Register-Transfer-Level Fault Simulator for Permanent and Transient Faults in Embedded Processors

C. Rousselle¹, M. Pflanz², A. Behling³, T. Mohaupt², H.T. Vierhaus²

¹ Kisters AG
Christian.Rousselle@Kisters.de

² BTU Cottbus – CE Group
pflanz@TU-Cottbus.de

³ Philips Semiconductors Hamburg
Andreas.Behling@philips.com

Abstract

HEARTLESS (Hierarchical Register-Transfer-Level Fault-Simulator for Permanent & Transient Faults) was developed to simulate the behavior of complex sequential designs like processor cores in case of transient and permanent faults. HEARTLESS can be enhanced by propagation over macros described in a C++-function. Available is a C-interface for access to internal signals during the simulation.

1. HEARTLESS Concepts

HEARTLESS (Fig. 1) can handle complex hierarchical sequential designs. It allows to observe and alter every single signal at all times. Furthermore HEARTLESS supports the use of high level constructs and functions to speed up the simulation, test the interfaces of different macros or screen signals automatically. The tool was developed in ANSI C++. It is easy to port it to every desired platform that supports ANSI C++.

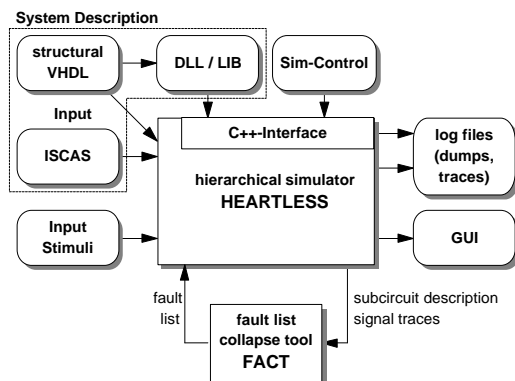


Figure 2: Basic structure of HEARTLESS

HEARTLESS supports VHDL and ISCAS as input formats. The VHDL reader supports a structural description with hierarchies, macros and reuse of components. VHDL can be used to handle large designs whereas ISCAS can be considered as a net list format that will usually not be used in the process of development. The VHDL input supports, apart from basic gates all

types of complex gates (macros). Additionally it is possible to define C/C++ functions that are treated as macros. These C/C++ functions can be used to replace large VHDL blocks that are a part of the circuit but are of no interest in the current simulation. It is also possible to use functions which may be included in dynamic link libraries (on Windows based systems) or in static libraries (on UNIX/Linux) systems. Another feature of the simulator is the capability to define functions that are executed during runtime. This allows to apply every kind of analysis at all times and can be used to implement basic invariant checks or to monitor signals.

2. Fault-Injection, -Models and -Collapsing

A fault-injection (FI) is possible at the gate and RT-level. At the RT-level, FI may encompass registers or busses. With modeling of functional blocks in VHDL or in 'C++', this scheme allows for a high degree of freedom with respect to efficient models for building blocks and also holds the promise to be faster than an approach that relies on a VHDL simulator. The fault models supported in this approach are: *single node stuck-at-0/1* (sa0, sa1), *transition fault* (delay fault), *single node flip-to-0/1* (ft0, ft1). The *fault collapsing tool FACT* was developed to minimize fault lists for sa0/1, ft0/1, or delay) at the gate level. Within HEARTLESS, the user can select blocks of the hierarchy. The selected structures and the signal values are passed to FACT in a modified ISCAS 89 format. Collapsing algorithms are based on dominance- and equivalence relationships. A modified ISCAS 89 format is used to describe gate-level circuits (e.g. an edge-triggered D-Flip-Flop with additional accessible clock and reset lines and flip-flop-state)

3. Summary

A flexible simulation environment was developed to analyze the fault-behavior of large designs. In the foreground of interest are complex microprocessor designs. On-line check units for processor components and signals can be validated easily with HEARTLESS. In current works we investigate the performance with the help of processor designs of various complexity.