A Structural Test Methodology for SRAM-Based FPGAs

Michel Renovell
LIRMM, Montpellier, France
renovell@lirmm.fr

Tutorial

Abstract

Programmable logic in the form of field-programmable gate arrays has become a widely accepted design approach for low- and medium-volume computing applications. Low development costs and inherent functional flexibility have spurred the spectacular growth of this technology. FPGAs are regular structures of logic modules communicating via an interconnect architecture of lines and switches. Users program the logic modules and interconnect structures to perform particular functions and realize the FPGA's global function. Manufacturers provide programmability in their FPGA architectures in various ways. They may use RAM to store configuration information, EPROM switches based on FAMOS transistors and fuses, or antifuses that permanently open or close connections. Of these alternatives, the presentation will focus on SRAM-based FPGAs. Their simplicity and flexibility for user changes in the field make this type of FPGA a very popular choice among designers.

The architecture and design of these devices have been widely investigated during the last decade, but their test challenges have received less attention. Only recently researchers have addressed the problem of testing FPGA after manufacturing, in other words, before user programming of specific functions. Testing before programming presents a wide spectrum of problems, for which a number of researchers have proposed innovative solutions. Testing an FPGA chip poses a challenging problem for test engineers. It requires implementing various configurations (programmings) of the FPGA. But changing configurations incurs reprogramming costs. So the fundamental question for FPGA testing is how can we determine the minimum number of test configurations and corresponding vector test sequences that will cover all the faults of a given FPGA's fault model? Solutions are presented in the presentation for various architectural elements of SRAM-based FPGAs.