The constantly growing requirements in terms of higher computing power, throughput, and storage capacity leads to development of highly sophisticated devices. Due to production costs of such advanced devices and to the most often mission-critical applications to which they are devoted, testing even at the very early stages of design is becoming mandatory. This consideration justifies the advent of design for testability techniques matched with thorough testing strategies performed during the design/production phase. In the present session we will deal with different aspects of the testing issue.

In the first paper, Pseudorandom versus Deterministic Testing of Intel 80x86 Processors, the authors apply a pseudo-random instruction generator to Intel’s 80x86 processor. A set of tools is capable of evaluating the effectiveness of the generated test programs. The approach adopted allows coverage of either simple stuck-at faults or more complex faults such as bridging, pattern sensitivity, and delay faults.

In the paper, On the Adequacy of Deriving Hardware Test Data from the Behavioral Specification, a technique usually adopted for software testing (mutation analysis) is transported so as to generate tests for behavioral VHDL descriptions. The results of such a technique are evaluated by comparing the coverage ration obtained with the one derived by traditional ATPGs, yielding comparable results in the case of combinatorial circuits and better results for sequential ones.

DELFIM: Error Detection by Thin Memory Protection, deals with another aspect of testing, i.e. concurrent error detection. Memory protection has been implemented by dynamically defining unused memory areas based on program modularity and accessibility. A watchdog system checks whether the executing function correctly addresses its own memory area. Results obtained increased error detection by at least 5% with respect to traditional mechanisms.
The last paper of the session, *SCAN/BIST Techniques for Decreasing Test Storage and their Implications to Test Pattern Generation*, addresses the problem of Design for Testability. The authors analyze two SCAN/BIST approaches, identifying conditions which grant shorter test sequences with respect to a scan method. Such analysis leads to a new concatenation strategy for test sequences based on implicit techniques.