Stanford Professor Edward J. McCluskey's address dealt with a design for testability using a cell library. Many industrial semicustom and custom parts are designed in the absence of strong design rules checking discipline. For these circuits, a library of cells and macrofunctions can be provided that include scan path features. Replacing existing cells of the chip with their corresponding parts from this library will create the scan path required for testability. In a recent application of this method, the size of a chip was actually reduced.

Melvin Breuer of USC described the development of a system, ADAM, to assist in choosing among testability enhancement techniques of VLSI chips—PLA circuits in particular. ADAM has a knowledge base containing information about seven of the approximately 30 design-for-testability methods for PLAs. Accepting as input a description of the PLA as well as user requirements and restrictions, ADAM is failure driven; given stringent requirements, it announces the failure to find a completely acceptable solution and then helps the user modify some of his requirements to reach the 'nearest' solution.

**Test generation.** Sentry Digital Test System's Kerry Kurakasi presented the merits of test definition language, or TDL, the basis of an automatic test generation tool. Using TDL, test program generation software increases test engineer productivity by transferring test-system expertise to a rule-based system.

Consider the following inadequacies: current test programs are not transportable; there is usually no DUT definition; tester limitations are hopelessly intermingled with DUT parameters; test engineers need to be expert with both the DUT and the tester; finally, there is no tester-independent test language on the market. TDL will solve most of these problems by providing a non-procedural description medium with the emphasis on DUT definition. TDL's structured tabular format provides a natural form for test parameters, DUT definition, and simulation and test strategy definition.

A growing problem—the cost of test generation—was addressed by Major Phil Gordon of the USAF. The Department of Defense proposes, as a solution, the integration of test specification and generation. Although still in the early design stage, the tester independent support software system, or TISSS, will be tester and simulator-independent, deriving its data directly from CAD tools, automatically translating the specifications to test programs.

Mark Shirley of MIT discussed the impracticality of classical test generation methods for large and complex circuits; the experiences of a test programmer can be encapsulated in the form of cliches from which test program fragments can be generated—fragments that can then be automatically combined within applicable temporal, structural, and resource constraints to produce a test program.

Alternatively, a simulator such as MARS can be used to convert testing goals into event patterns from which code can be extracted. Both techniques emphasize test program construction rather than the currently pervasive use of random test pattern generation.

To alleviate design-test cycle difficulties and to reduce testing costs, Louis McDonald of Hughes pointed out, software tools have been developed to assist users in test generation and evaluation tasks. A processor, used in design rule checking and circuit analysis, partitions the circuit into combinational parts and set-scanable sequential parts; combinational circuits are further divided if necessary. An ATPG module then creates tests, which are graded using a fault simulator. This system, according to McDonald, is currently being refined.

Traditionally, ATPG programs depend on the sensitization of a single path that leads to numerous searches the algorithm must conduct to produce tests for all elements in the DUT. Based on research he has conducted with Mahieddine Ladadji, Rensselaer Professor John McDonald said that gang testing accelerates the rate of test pattern generation by finding patterns that simultaneously sensitizes many paths for many circuits—an acceleration achieved by using the subscripted D-algorithm that sensitizes all control and observation paths of a gate and assigns flexible values (signals that can be set to 0 or 1 as the need arises) to these paths.

To reduce conflicts while preserving the independent identities of the mergers, merging of the observation paths at various gates is allowed. Examples of gang testing, the D-algorithm, and the improved D-algorithm demonstrated gang testing's superiority—improvements quantified by the number of test patterns needed and the time required to achieve a desired fault coverage.

**Causal reasoning diagnosis.** Using primary input values and observed inputs, logical inference can be used to identify potentially faulty elements—elements that can be tested and eliminated based on logical inconsistency between expected and observed values. Therefore, according to Bruce Havlicek of Westingham, system fault diagnosis can be performed without the use of explicit fault models or test programming.

Diagnosis covers several inference phases. Primary inputs are set in the first phase and the correct values of circuit nodes are determined. Observations are then entered and potential fault candidates are determined by noting inconsistencies between expected value and inferred value. During the next inference phase, the behavior of each candidate is disabled; if inconsistencies disappear, then the candidate can be faulty—if inconsistencies persist, then the disabled candidate cannot be faulty.

Some audience members noted the use of a fault model despite the no-fault-model claim, since only single faults are handled and no time dependencies are assumed.

**Workshop conclusions.** Some of the topics discussed in the 1983 ATPG workshop on expert systems appeared as prototype working systems in this year's workshop. At the next workshop, we may hear about hardware-accelerated test generation algorithms.

An interest was expressed in conducting panel discussions, and in continuing to conduct the workshop every other year rather than annually.