1989 ATE & Instrumentation Conference West

TEST 1989

ATE & Instrumentation Conference West

ATE & Instrumentation Conference
Proceedings

INNOVATIONS IN TEST

January 23-26, 1989
Disneyland Hotel
Anaheim, CA

Sponsored By:
Electronics Test, Circuits Manufacturing and EOS/ESD Technology magazines

Endorsed By:
The American Society of Test Engineers (ASTE)

ATE & Instrumentation Conference West

Library of Congress Catalog Card Number: 88-63715
International Standard Book Number: 0-87930-212-7

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and are published as presented and without change, in the interests of timely dissemination. Their inclusion in this publication does not necessarily constitute endorsement by the editors, MG Expositions Group, Div. of Miller Freeman Publications, Inc.

Order: **MG Expositions Group**
1050 Commonwealth Avenue
Boston, MA 02215
(617) 232-3976 or (800) 223-7126

Staff

Conference Management Staff
Steve Schuldenfrei, *Director*
Caroline Carby, *Accountant*
Caroline Chesnut, *Technical Program Manager*
Cathy Gibson, *Director of Marketing Communications*
Pam Farrell, *Production Manager*
Maureen Grady, *Operations Assistant*
Paul Grazda, *Sales Manager*
Maureen Joyce, *Art Director*
Alicia Leahy, *Director of Operations*
Ed Martin, *Director of Sales*
Lisa Palange, *Administrative Assistant*
Maureen Sheppard, *Sales Assistant*

Technical Advisory Committee
Bob Bergstrom, *John Fluke Manufacturing Co.*
John Cessna, *Bendix/King*
Dave Daugherty, *TRM Inc.*
Robert Hanson, *Boeing Electronics*
Mike Horton, *Hewlett-Packard*
John Huggins, *NCR Corp.*
Ken Johnson, *Testech Ltd.*
Daryl Layzer, *Teradyne*
Norbert Laengrich, *HILEVEL Technology*
Clayton Mohr, *Tektronix*
Len Prednis, *TRW*
Chet Pribonic, *Schlumberger Technologies*
Jim Ricks, *Texas Instruments*
Dave Smith, *Computer Automation*
Jon Turino, *Logical Solutions Technology*
Steve Walsh, *John Fluke Manufacturing Co.*

Conference Track Chairmen
Design Track
Robert Hanson, *Boeing Electronics*

Manufacturing Track
Len Prednis, *TRW*
Ken Johnson, *Testech Ltd.*

System Support Track
Steve Walsh, *John Fluke Manufacturing Co.*

Joint Sessions Coordinators
Mike Horton, *Hewlett-Packard*
Bob Bergstrom, *John Fluke Manufacturing Co.*

Associations
Martin Barton, *Surface Mount Technology Association*
Ken Johnson, *American Society of Test Engineers*
George Keller, *Association of Field Service Managers*

Publications
Jim Brinton, *EOS/ESD Technology*
Chris Everett, *Electronics Test*
John Tuck, *Circuits Manufacturing*
Brook Taliaferro, *Computer/Electronic Service News*

Session Chairmen
Design Track
K.C. Sun, Apollo Computer
Pat Griffin, *TYX*
Frank Binnendk, *Mentor Graphics*
Peter Hansen, *Teradyne*
Jim Kroening, *Digital Equipment Corp.*
Bill DenBeste, *Electronic Tools*
Ronald Fonden, *Boeing Electronics*
Jon Turino, *Logical Solutions Technology*

Manufacturing Track
David Dunham, *TDS Production Services*
Alan Sugarman, *TRW*
Rudy Arp, *Western Digital Corp.*
Robert Beauchesne, *Honeywell Bull*
Louis Klahn Jr., *Colorado Data Systems*
Ron Wolfe, *National Instruments Corp.*
John Cessna, *Bendix/King*

System Support
John Huggins, *NCR Corp.*
Arnold Greenspan, *Softech Inc.*
Shirish Trivedi, *Westinghouse Electric*
Al Hahn, *Testech Ltd.*
Dave Grant, *RESTOR Communications*
Nick Gerbracht, *Tektronix*
Steve Walsh, *John Fluke Mfg. Co.*

Joint Sessions
Dr. James Peckol, *John Fluke Mfg. Co.*
Frank Krizan, *Rockwell International*

Advisory Board and Committee
Industry
Lois Blackburn, *Support Technologies*
Carl Derry, *LeCroy*
Christine Hall, *Aetrium*
Kathryn Henderson, *Exatron*
Robert Kirke, *Thermonics*
Kyle Kraus, *GenRad*
Jean Manera, *HILEVEL Technology*
William Porter, *Hewlett-Packard*
Joyce Tomasco, *Teradyne-Zehntel*
Meda Warnick, *Everett/Charles Test Equipment*
Debra Weinblatt, *Computer Automation*
Jan Woods, *Tektronix*

MG Expositions Group
1050 Commonwealth Avenue
Boston, MA 02215
(800) 223-7126• (617) 232-3976
ATE & Instrumentation Expositions & Conferences, for professionals who use test and instrumentation equipment, are produced by MG Expositions Group and sponsored by *Electronics Test, Circuits Manufacturing* and *EOS/ESD Technology.*

Program Track Chairmen

Design Track
Robert Hanson

Bob Hanson is currently a Senior Research Specialist for Boeing Electronics Co. He has over twenty-nine years experience working for The Boeing Company, Honeywell, Rockwell and Goodyear Aerospace. He has lead both hardware and software projects designing testers for factory, base level and depot level environments, and has worked with AF System Project Offices and various Air Logistics Centers and Air Force bases on many testing programs. He has worked and managed companies in CAE/CAD/CAT, CIM, JIT, simulation and automatic assembly environments.

Manufacturing Track
Len Prednis

Len Prednis is currently the manager of test technology for TRW Electronic Systems Group in the advanced avionics manufacturing facility located in San Diego, CA. During his 25-year career in the electronics industry, he has worked directly for G.E., Burroughs and Research Cottrell. In addition, Len founded and served as president and chief executive officer of National Electro-Mechanical Systems, Inc. His experience includes engineering consulting contract assignments with G.E., IBM, Avco, Honeywell, Mitre and Lincoln Labs.

System Support Track
Steve Walsh

Steve Walsh is the Marketing Manager for John Fluke Manufacturing Co. in the Customer Support Service department. After receiving an International Marketing degree in Europe, Steve returned to the United States and held product planning and marketing positions with both test equipment and computer peripheral companies. Steve is an active member of the Association of Field Service Managers International (AFSMI).

Manufacturing Track
Ken Johnson

Ken Johnson is the president of Testech, Ltd., a test-strategy consulting company, and president of the American Society of Test Engineers. During his 12-year career in industry, he has operated a service business, sold components for a major distributor, and managed a test engineering department responsible for designing and implementing test strategies for high and low-volume manufacturing of analog and digital systems.

"BEST AT TEST" Paper Awards

ATE & Instrumentation East '88

The Award--The "BEST AT TEST" Awards are given by conference attendees. Three papers are selected, as determined by their individual ratings on the session evaluation cards to receive the "BEST AT TEST" Award in design, manufacturing and system support. Winners receive a commemorative plaque at the Plenary Session and are asked to present paper at the next conference.

Design Track

CAE-Based Testability Rule Checker Tool

Ronald Fonden, Boeing Electronics
Ron is a CAE Applications Engineer at Boeing Electronics Co. He has worked at Boeing for five years in Manufacturing Test Engineering, Circuit Design and Design Automation. He received a BSEE from North Dakota State University and a BS in Astronomy from Moorhead State University, Minnesota.

Manufacturing Track

Evaluating the Limitations of High-Speed Board Testers

John Arena, Teradyne
John is an Applications Manager for Teradyne's L290 and L210 board test products. He joined Teradyne in 1979 as a hardware design engineer. He received a BSEE from Rensselaer Polytechnic Institute and a MBA from Boston University.

System Support Track

An Investigation of Predictive Support as a Board Test Service Tool

Jeannette Herring, Hewlett-Packard
Jeannette is a Product Support Engineer with Hewlett-Packard, Manufacturing Test Division. Since joining Hewlett-Packard in 1985, she has worked in on-line hardware and software support, off-line hardware support and board test programming. She received a BSEE and a BS in Secondary Math Education from Mississippi State Univesity.

KEYNOTE ADDRESS

John Young
Hewlett-Packard Co.

Life Begins at Fifty

As Hewlett-Packard enters its fiftieth year as a major supplier of test and measurement instrumentation, the question arises: Will the next half-century provide the technological excitement and growth opportunities we've seen in the past? Or is T&M a "mature" business, destined to eventually lose the battle against old age? The title of this address indicates how HP has answered its self-directed questions. In his keynote address, John Young examines some key economic and technological trends that ensure an exciting future for the instrument business and some of the technological frontiers that remain to be crossed.

About the Keynote Speaker--John A. Young is president and chief executive officer of Hewlett-Packard Co., located in Palo Alto, CA.

Young has served as HP's chief executive officer since May 1978 and as chairman of the executive committee of HP's board of directors since March 1983. He had served as the company's chief operating officer and president since September 1988.

Young joined Hewlett-Packard's marketing planning staff in 1958 after receiving a master's degree in business administration from Stanford University. He subsequently served as a regional sales manager, as a member of the corporate finance staff, and as marketing manager of the former Microwave Div. In 1963, he was appointed Microwave Div. general manager.

In 1968, Young was named vice president of the company and assumed responsibility for the newly formed Electronic Products Group, which included instruments, components, and measuring systems produced by Hewlett-Packard.

Young was appointed an executive vice president and elected to the company's board of directors in September 1974. At the same time, he was named to the executive committee, which was established to coordinate all phases of the company's operations. As executive vice president, Young was responsible for HP's Instrument, Computer Systems, and Components Groups.

Young was born April 24, 1932, in Nampa, Idaho, graduated from Oregon State University with a bachelor's degree in electrical engineering.

Table of Contents

*Papers available at time of session.

TEST GENERATION FOR MIXED SYSTEMS OF SCANNABLE AND CONVENTIONAL LOGIC

John L. Braden

APOLLO COMPUTER, INC.
330 Billerica Road, Chelmsford, MA 01824

ABSTRACT

A method for partitioning systems of logic containing both scannable and non-scannable elements is presented. This approach uses a program to perform a depth-first traversal of the system-level design beginning at scannable observation points, splitting the model into subsets which we call test targets. These test targets are easily manageable pieces of logic bounded by scannable latches or statically initialized control elements which can be treated as independent designs. The test targets can be processed with commercially available automatic test pattern generation and simulation programs to produce scan-based test vectors. The resulting test vectors are then processed by a scan test formatter program which uses the scan order files from the original design to build packed binary test data files. The test data files are run on the hardware to identify failing boards and components. This approach allows us to take practical advantage of the increased controllability and observability offered by scannable components even in designs which mix scannable and conventional logic. The practical application of these tools to the design of the new Apollo *PRISM*™ Architecture product, the Series 10000 Personal Supercomputer™, and the lessons learned from this experience are discussed. Topics for future research in the scan tools area are also indicated.

INTRODUCTION

As designs become increasingly complex and make use of dense gate arrays with high gate-to-pin ratios, the issue of testability becomes crucial to the eventual success of a product. Although such testability features as scannable latches have been known and discussed for many years, they have been used at the system level by only a few manufacturers who design and manufacture large numbers of their own custom circuits. Examples are IBM[1][2] and Control Data[3] who extensively use these techniques in the design of their high-performance computer systems. Major semiconductor manufacturers have been slow to offer testability features in their commodity parts since they invariably add complexity and thus increase the cost of components. We have just recently begun to see a shift in this view as VLSI chips have increased in physical size and pin count to the point where wafer probe testing of these parts is very costly for the component manufacturer. The addition of boundary scan, circuits which scan the I/O pins of the device, can greatly reduce the cost and complexity of testing these components at all levels of assembly. If this is combined with scan access to internal latches in the device, the test coverage can be significantly increased and the number of test vectors required greatly reduced.

Most manufacturers provide access to the test pins primarily to check the parts after packaging, with very little planning as to how they could be used at the systems level to increase testability. There are currently no industry standards which define how scan paths should be controlled or clocked, although organizations such as the Joint Test

Action Group[4] have begun to work in that direction. The choice of off-the-shelf scannable SSI or MSI components is very limited and those available are often ruled out because of timing or other design constraints. As a result, in commercial designs today we face, at best, a limited scan test capability at the systems level.

In this paper we will explore how to best exploit this scan test capability and use it to provide both a high percentage of fault coverage and isolation of faults to the component level, without expensive automatic test equipment. With the proper tools and planning, even this limited scan capability can significantly increase the controllability and observability of your design, resulting in a more testable product with lower manufacturing and maintenance costs. The design for which we developed these tools is the new Apollo *PRISM*™ Architecture product, the DN10000. This is a personal supercomputer graphics workstation with up to four processors, multiple memories, a utilities board and a graphics system each containing many scannable CMOS and ECL gate arrays.

SCAN PATH CONTROL REQUIREMENTS

In order to use the component scan capabilities at the system level, a uniform method of addressing and controlling the scan paths and clocks was required. Although the details of this architecture is discussed in much greater detail in other recently published papers[5,6,7], a brief overview of the scan subsystem will help clarify the problems we faced and the approach we pursued. The DN10000 workstation contains a microprocessor and associated memory used exclusively as a maintenance or diagnostic processor. This processor is used to perform all scan testing as well as to initialize all scannable parts before the main processors are given control. Each board in the system contains a gate array called the "scan/clock resource" (SCR) which can be controlled by the service processor over a diagnostic bus, and which itself can control up to eight separate scan paths on its circuit board. The SCR can apply system and scan clocks and can directly control a number of test signals on its board. For power-up testing it can generate pseudo-random test vectors using a built-in linear feedback shift register, apply these to the scan paths, and accumulate signatures for each scan path. The design of the SCR is general enough that a variety of scan path control methods can be supported, and in the DN10000 we have everything from scannable MSI registers to CMOS & ECL gate arrays.

The combination of a separate processor devoted to test and a gate array on each board devoted to scan and clock control meant that complete system level testing down to the component level was now possible. Our task was to put together the tools, procedures, and tests to make this possibility become a reality.

CONCEPT OF TEST TARGETS

At the system level, we are dealing with in-circuit test, and can no longer use the same test vectors which were created for testing the individual components. Although pseudo-random testing provides good fault coverage, it does not provide the component-level isolation which we require for board repair. Most currently available commercial Automatic Test Pattern Generation (ATPG) tools are unable to deal with any nets containing sequential logic, and if circuit designs become extremely large their efficiency drops dramatically. In order to deal with this problem, we needed a tool which could partition the system-level design into cones of logic which contain a set of observation points and a set of control points, both of which are either directly controllable/observable or accessible via scan paths. The logic between the control and observation points must be purely combinatorial. These design fragments are referred to as "test targets", and the tool we use to carve the targets from the full design we called a "target model generator". Partitioning of logic systems in a similar manner has been presented as a means of reducing the number of inputs for pseudoexhaustive testing of partial dependence circuits[8]. A

significant reduction in the computation time required by present day automatic test pattern generation (ATPG) programs can also be achieved by this approach. The fault simulation time has been shown to grow in proportion to at least the square of the number of gates in the circuit using the best test generation techniques[9].

In the DN10000 we used three different types of scannable components. Most of the ASIC designs were 1.5 micron CMOS gate arrays based on the ICS10000 series from Integrated CMOS Systems. These components have two scan rings, internal and external. The external scan ring includes one shift latch for each I/O pad (except power & ground). There is no isolation from attached circuitry, so test vectors can only be driven into the parts at the wafer probe or component test level via this path. The internal scan ring is composed of all scannable latches in the gate array implementation. These are DSRL master slave latches with an additional scan master. A second type of scannable component used was the AM29818 serial shadow register. It is an 8-bit pipeline register with a scannable serial shadow register. Finally, the floating point parts chosen were scannable ECL gate arrays from Bipolar Integrated Technologies. The design also required a significant number of conventional non-scannable components which included both combinatorial and sequential circuits. We were unable to find an automated way of generating patterns for all of these parts, so a conventional functional test approach was necessary, still making use of scan for better access and visibility.

The Target Model Generator (TMG) had to to identify different types of logic cells and understand the relationship between the various inputs and outputs of each cell. In carving the targets from the design, TMG traverses the design backwards beginning at an identified set of observation points (usually scannable latches or I/O cells with boundary scan), and including all combinatorial elements feeding these points until each path reaches a stopping cell. Stopping points include scannable latches as well as non-scannable register or memory elements. As noted above, the boundary scan latches do not have sufficient power to drive logic off-chip, and therefore can only be used for observation. Because of this they cannot be not used as stopping points, but do add additional observation points for fault isolation. We found it useful to consider two major categories of test target:

- Logic cones ending at an internal scan latch in which all nets trace back exclusively to other internal scan latches on the same gate array.

- Logic cones ending at an internal scan latch in which one or more nets trace back to some origination point other than an internal scan latch on the same gate array.

In the first case, called INTERNAL targets, no setup of other gate arrays or conditioning of external signals is necessary. This type of target is applicable only to portions of scannable gate arrays, but because most very large gate arrays contain lots of internal logic, it may still be necessary to partition the logic to provide reasonable ATPG run times. We can identify the observation points for this logic by traversing the design database inward for a particular gate array starting at all input and bi-directional buffers and identifying all latches directly affected by these input buffers. Eliminating these latches from the list of all latches in the design gives the set of observation points for the internal targets. We begin from these observation points and use TMG to carve the logic back to the scannable control points, creating a new design for this target. The test vectors and expected results are generated by expanding the target design(s) output by TMG and running ATPG.

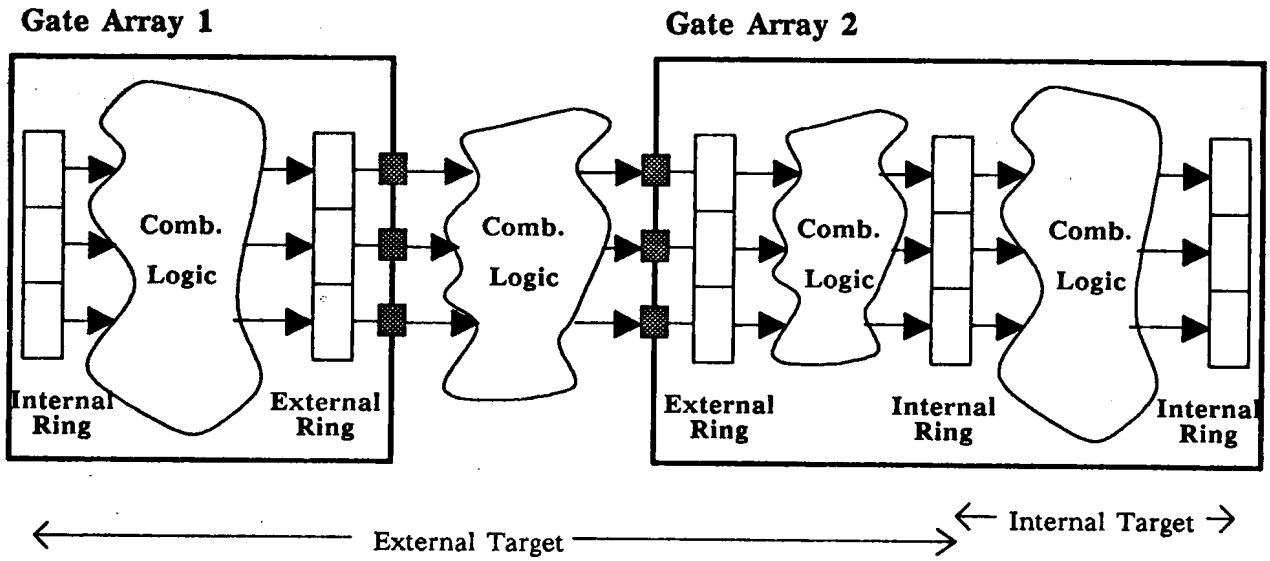

Figure 1

The second case, EXTERNAL targets, requires more complexity in the TMG, since we are handling bi-directional (tri-state) busses and a number of non-scannable component types. All logic cones which have been found to contain one or more remote origination points must be traced back to their respective control or stopping points. We define a control point as a scannable latch on a gate array or other device. Other non-control stopping points include outputs of non-scannable latches or memories and signals tied to or indirectly controllable to some known state (ie. Vcc). These non-control stopping points require special setup before ATPG to identify the state in which they will be held throughout the test process. If the state of these latches cannot be determined, we set their state to 'X' for test pattern generation and later analyze faults which are called out as undetectable.

TARGET MODEL GENERATOR

The Target Model Generator program was developed to provide a means of creating complete functional designs containing subsets of logic for which tests could be automatically generated. These tests can be targeted at individual gate arrays or across many different components at the board or system level. Through use of a cell library, the program supports all cell and component types appearing in the design of the DN10000 workstation. We used an integrated commercial design system for logic simulation and ATPG, and developed our carving tools to work directly from the logic description files used to define the gate array and system design. The TMG program requires as input a *cell library* database, the *design* file describing the system from which test targets are to be carved, and the *target input* file, a list of cell instances which will serve as observation points for this test target. The program produces a number of informational files as well as a complete *target design* file ready to be run through the design system tools. To aid in identifying and partitioning test targets, TMG also allows logic to be traversed in a forward direction from arbitrary points in the design. This allows all scannable latches affected by these nets to be identified and placed in a new target input file, ready for use. In the following paragraphs we describe in much greater detail the functions supported by TMG and how these are used, as well as algorithms and techniques used.

Supported Functions

TMG as currently implemented actually consists of two programs. The first piece reads the cell library and the master *design* file and constructs a database optimized for design traversal. The TMG program uses this database to perform the following functions:

- Target Carving– This is the main function and purpose of TMG. It will create a new *target design* file based on the master design and using the observation points identified in the *target input* list. A list of scannable I/O latches which are part of the target is also produced to identify additional observation points which can aid in fault isolation

- Forward Traversal– This function starts with a list of starting cells and produces a *target input* file of all observation cells affected by these cells.

- Postorder Traversal– This function is used to provide a definition–before–use ordered list of cells which can be used for constructing schematic drawings of the targets. This information is output with both unique integer values used by the drawing tool and user–readable ASCII net and cell names.

There are a number of options which can be selected by the user to produce additional files of information or influence the way in which stopping cells are identified.

Data Structures and Algorithms

Although the design of TMG continues to evolve, a brief outline the data structures & algorithms used in our first implementation should help in understanding. All development work was done on Apollo workstations and most code was written in Pascal. In the first phase, the *system design* file is read and each entry is parsed. Two dictionaries[10] are built, one for cells (or objects) and a second for signals (or nets). Each of the dictionary entries contains pointers to linked lists which cross–reference cells to input and output signals and signals to source and destination cells.

The cell dictionary entries include the cell name, its type, and any other information needed to re–create the entry as well as linked list pointers to signal lists for both input and output signals. The signal list entries contain the signal dictionary entry index for this signal, and a pointer to the next signal list entry. The signal dictionary is similar to the cell dictionary, but a little bit simpler. It also has an entry for the signal name text, but doesn't need as much supporting information. It has linked list pointers to cell list entries for both the source and destination cells of this signal. These cell list entries have a cell dictionary entry index and a link to the next cell list entry.

Reading the full design definition file, parsing all of the cell types and building the dictionaries takes significantly more time than the actual carving of targets, so the construction of this database is now done in a separate program of the TMG package. Using the first implementation, for a board–level design containing approximately 50,000 cells (300–400K gates) the database build time was approximately 17 minutes on a DN570–T with 32MB of memory, and carving a typical–size target took about 30 seconds.

Once the data structures are built, most of the work of the main program is done simply traversing these structures. The actual cutting of a target begins by reading an entry from a *target input* file. This cell name is hashed with the end–weighted multiplicative hash function[11] found to give the best performance on the highly repetitive names found in the designs, and a locate routine is used to find the entry in the dictionary. A depth–first search algorithm is then used to first mark, then to recursively depth–first search each cell driving a signal which is an input to this cell. The depth–first search stops when a stopping cell such as a DFF is encountered. When all of the logic feeding the cells listed in the *target input* file has been marked, the new *target design* file is constructed and the appropriate

ports, nets, and all marked cells are output to this file. Forward tracing of a net is performed in a similar fashion, except that the outputs rather than the inputs of cells are traced, and the names of the stopping cells are output directly to a *target input* file rather than constructing a *target design* file.

The postorder traversal of a target uses an identical search algorithm, but in this case we do not mark a cell as visited until the last time we traverse it. This allows us to output information about the logic connectivity in a "definition before use" order which provides the flattened description necessary to construct machine-generated schematics.

COMMERCIAL DESIGN SYSTEM TOOLS

After partitioning the design with TMG and generating a *target design* file for a portion of the system we wish to test, this new test target can be treated as if it were just another ASIC design ready for automatic test pattern generation. Any of a number of commercial packages are available which combine ATPG, fault simulation, and logic simulation for combinatorial designs. These packages may use the D-algorithm, Path Oriented DEcision Making (PODEM), RAndom Path Sensitizing (RAPS)[12], or any number of other proprietary schemes for generating test patterns. Since this design was carved from a board or system-level design, the resulting stimulus and response vectors can be applied at the system level. If additional observation points are available for improved component isolation (ie. external scan ring latches), a logic simulation using the stimulus vectors may be run with probes of these observation points placed in another *target observation vector* file. More work is required, however, before these vectors can be applied directly to the hardware, since they typically may consist of pieces of several gate arrays as well assorted SSI & MSI components spread over one or more different circuit boards. We need another tool to take the randomly ordered and distributed scan bits given in our test vector file and relate these vectors to the actual board topology at the system level, however. We call this tool a Scan Test Formatter (STF)

SCAN TEST FORMATTER

The Scan Test Formatter (STF) is the tool which is used to convert the *target test vector* files resulting from ATPG runs on diagnostic test target designs (extracted from a system-level design) into a concise database of scan test vectors which can be run on the hardware using either a stand-alone local test manager, or remotely using an interactive scan test environment running on a networked workstation. It produces both a human-readable ASCII form of the database for debug and documentation purposes and a data-compressed binary version which is used by the service processor. STF uses as its primary inputs the *target test vector* file containing the ATPG vectors output for this test target, optionally the *target observation vector* file which contains simulation results of the external scan ring state for each of the vectors, and the *target input* file which identifies the observation cells for this target. STF will also use one or more *target-level descriptor* files which identify everything about the hardware environment it needs to know. A portion of one of these files is shown in figure 2.