IEC 61445

Edition 1.0 2012-06

IEEE Std 1445™

INTERNATIONAL STANDARD

Digital Test Interchange Format (DTIF)

IEC 61445:2012(E) IEEE Std 1445-1998

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,…).
It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IEC 61445

Edition 1.0 2012-06

INTERNATIONAL STANDARD

IEEE Std 1445™

Digital Test Interchange Format (DTIF)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XD

ICS 25.040; 35.060

ISBN 978-2-83220-105-3

Contents

Digital Test Interchange Format (DTIF)

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation.

   IEEE Standards documents are developed within IEEE Societies and Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of IEEE and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards. Use of IEEE Standards documents is wholly voluntary. IEEE documents are made available for use subject to important notices and legal disclaimers (see http://standards.ieee.org/IPR/disclaimers.html for more information).

   IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE-SA) Standards Board.

3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.

8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility.

International Standard IEC 61445/ IEEE Std 1445-1998 has been processed through IEC technical committee 93: Design automation, under the IEC/IEEE Dual Logo Agreement.

The text of this standard is based on the following documents:

IEEE Std	FDIS	Report on voting
IEEE Std 1445-1998	93/321/FDIS	93/328/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The IEC Technical Committee and IEEE Technical Committee have decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IEEE Std 1445™-1998(R2009)

IEEE Standard for Digital Test Interchange Format (DTIF)

Sponsor

IEEE Standards Coordinating Committee 20 on Test and Diagnosis for Electronic Systems

Reaffirmed 9 December 2009
Approved 8 December 1998

IEEE-SA Standards Board

Reaffirmed 8 July 2004
Approved 16 November 1999

American National Standards Institute

Abstract: The information content and the data formats for the interchange of digital test program data between digital automated test program generators (DATPGs) and automatic test equipment (ATE) for board-level printed circuit assemblies are defined. This information can be broadly grouped into data that defines the following: UUT Model, Stimulus and Response, Fault Dictionary, and Probe.
Keywords: automatic test equipment (ATE), digital automated test program generator (DATPG), digital test interchange format (DTIF), Fault Dictionary data

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

IEEE Introduction

[This introduction is not part of IEEE Std 1445-1998, IEEE Standard for Digital Test Interchange Format (DTIF).]

A digital automated test program generator (DATPG) produces test pattern and diagnostic data that can be used for testing printed circuit assemblies on automatic test equipment (ATE). The use of several DATPGs, all with individual output formats, created a need for many unique post-processors to be developed and maintained for the life of the ATE. These post-processors supported the link from specific DATPGs to specific testers. The proliferation of unique formats and post-processors created logistical support problems and therefore identified a need for standardization. A DATPG and ATE independent output data format is required to limit the number of post-processors (one for each ATE) requiring life cycle support. The digital test interchange format (DTIF) was chosen because of its wide use and because it was becoming known in industry as the de facto standard.

This document provides the basis to standardize digital test information for use on ATE. The digital test information consists of the unit under test (UUT) Model information, Stimulus and Response data, Fault Dictionary data, and Probe data.

DTIF is unique from other standards such as IEEE P1450 (Draft 0.95, dated July 1998),[1] Draft Standard Test Interface Language (STIL) for Digital Test Vector Data, and IEEE Std 1029.1-1991, IEEE Standard for Waveform and Vector Exchange Specification (WAVES). STIL is being developed to standardize the output interface of existing computer-aided engineering (CAE) tools with the input interface of ATE for integrated circuit (IC) testing only. WAVES is a hardware descriptive language used for defining stimulus and response, and their associated timing for IC/board-level design. Neither STIL nor WAVES provides for board-level fault diagnostics.

A future revision of this standard will consider the use of the information model.

1 IEEE P1450 is an IEEE authorized standards project that was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining the draft, contact the IEEE.

Digital Test Interchange Format (DTIF)

1. Overview

The digital test interchange format (DTIF) is designed to provide a mechanism for digital test data interchange independent of specific digital automated test program generators (DATPGs) and test systems. The DTIF provides a neutral database for the development and delivery of digital simulation based test program sets (TPSs). DTIF functionally supports the unit under test (UUT) Model, Stimulus and Response, Fault Dictionary, and Probe databases.

1.1 Scope

This standard defines the information content and the data formats for the interchange of digital test program data between DATPGs and automatic test equipment (ATE) for board-level printed circuit assemblies. This information can be broadly grouped into data that defines the following:

a) UUT Model;
b) Stimulus and Response;
c) Fault Dictionary;
d) Probe.

1.2 Purpose

The purpose of this standard is to provide a standard output format for test data generated by a DATPG. A DATPG produces test patterns and fault diagnostic data for ATE. This data is used in applications such as board-level assemblies where diagnostic data interchange is important.

1.3 Application

This standard is primarily intended for use by digital simulator developers/maintainers and TPS developers/maintainers.

2. References

This standard shall be used in conjunction with the following standards. When the following standards are superseded by an approved revision, the revision shall apply.

ANSI X3.4-1986 (Reaff 1997), Information Systems—Coded Character Sets—7-Bit American National Standard Code for Information Interchange (7-Bit ASCII).[1]

IEEE Std 100-1996, IEEE Standard Dictionary of Electrical and Electronics Terms.[2]

3. Definitions and acronyms

3.1 Definitions

The following definitions are for use with this standard. For other uses and for definitions not contained herein, see IEEE Std 100-1996. Unless otherwise indicated, the ATPG subcommittee formulated all terms defined in this subclause.

3.1.1 burst: A set of stimulus patterns and related unit under test (UUT) responses that are set up, applied, and read as a group. A test program may employ more than one burst to provide the stimuli and responses necessary to test the UUT.

3.1.2 channel: The tester electronics associated with a digital input/output (I/O) pin that either drives or senses a particular node on the unit under test (UUT).

3.1.3 circuit simulator: A software program that predicts a circuit's response to a given stimulus.

3.1.4 digital automatic test program generator (DATPG): A program, often based on simulation, that aids in the development of test patterns and diagnostic information from the model of a unit under test (UUT).

3.1.5 dynamic patterns: A set of controlled, time-variant patterns within a time interval.

3.1.6 edge: A logic state transition that is considered instantaneous for a given pattern in the simulation process.

3.1.7 end-to-end test: A test sequence to establish pass (functioning properly) or fail (not functioning properly) conditions. *Syn:* **go/nogo test.**

3.1.8 fault set: A group of one or more faults with the same fault signature.

3.1.9 fault signature: A set of unique primary output patterns in which the fault will produce a response different from the good machine response.

3.1.10 fault title: A two-part description that includes a node name and a fault type [i.e, <U5>6 SA1 (component: U5, pin: 6, fault type: Stuck at 1)].

3.1.11 go/nogo test: *See:* **end-to-end test.**

[1]ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://www.standards.ieee.org/).

3.1.12 logic state: The representation a simulator uses to describe the state of a circuit during digital logic simulation. There are four types of logic states that exist in a typical simulator: 0, 1, Z, and X.

3.1.13 main model: The top-level unit under test (UUT) model description that includes a list of component packages and a netlist.

3.1.14 netlist: A point-to-point description of the interconnections between individual components in a circuit.

3.1.15 patterns: A set of unit under test (UUT) stimulus and expected response states. A pattern contains one unit of logic state (0, 1, X, Z) data for each UUT input and each UUT output pin.

3.1.16 phase: The time within a timing cycle when a primary input is in transition between logic states.

3.1.17 primary input (PI): A node in a circuit in which the tester can apply stimulus.

3.1.18 primary output (PO): A node in a circuit in which the tester can observe a response.

3.1.19 primary output patterns (POPAT): A set of unique responses at the node in which a fault or a group of faults are detected.

3.1.20 probe: A tester instrument used to observe the state of a node.

3.1.21 probeable node: Any node that is physically accessible to a tester probe.

3.1.22 probe window: The period of time during a pattern when a probe can capture activity on a node.

3.1.23 probing: A fault diagnostic technique that incorporates the use of a portable device (hand-held or robotic) to monitor or capture unit under test (UUT) response data. The location of the probe placement is determined by the circuit response and the circuit topology.

3.1.24 simulation time unit (STU): A fixed unit of time that is utilized during simulation for evaluation of data.

3.1.25 skew: The timing ambiguity associated with the occurrence of an automatic test equipment (ATE) Input/Output (I/O) event that is due to the physical limitations of the ATE digital driver and detector electronics.

3.1.26 static patterns: A set of controlled, time-invariant patterns.

3.1.27 stimulus: The logic states within a pattern that drives a circuit model in simulation, or a unit under test (UUT) on an automatic test equipment (ATE).

3.1.28 strobe: To record or measure the state of a particular node at an instant in time. Strobing will have a skew associated with it.

3.1.29 threshold voltage: The minimum voltage considered to be a high state or the maximum voltage considered to be a low state.

3.1.30 timing ambiguity: The period of time in a nodal transition during which the state of the node cannot be guaranteed.

3.1.31 timing generator: The function in the automatic test equipment (ATE) that stores and produces timing sets, or its analogous construct in the simulation process.

3.1.32 timing set (TSET): An automatic test equipment (ATE) timing-cycle during which stimuli are applied and unit under test (UUT) responses are measured. A timing set includes the specification of the pattern period, UUT input pin groupings that will transition at a specific time within a pattern, and UUT output pin groupings that share the same window.

3.1.33 trace: A diagnostic fault isolation program that uses a probe on a tester.

3.1.34 window: The period of time during a pattern cycle when a primary output is actively monitored by an automatic test equipment (ATE) channel.

3.2 Acronyms

ASCII	American Standards Code for Information Interchange
ATE	automatic test equipment
CAE	computer-aided engineering
DATPG	digital automated test program generator
DTIF	digital test interchange format
IC	integrated circuit
I/O	input/output
FLAP	fault set last analyzed POPAT
PI	primary input
PO	primary output
POPAT	primary output patterns
STU	simulation time unit
TP	test program
TPS	test program set
TSET	timing set
UUT	unit under test

4. Data organization overview of the DTIF standard environment

Digital circuit simulators for test, measurement, and diagnostic equipment provide an effective way to predict UUT behavior (good and bad) during test on an ATE. Results of the simulation can be used to generate the data necessary to produce a test program for the digital UUT. This standard defines the data generally available from test simulation and describes its structure for use in generating digital test programs for ATE.

Digital simulators typically provide data on the UUT Model, Stimulus and Response, Fault Dictionary, and Probe. Figure 1 shows a total of 39 American Standard Code for Information Interchange (ASCII) files, which includes a Header File and 38 data files. The Header File provides summary information and a listing of the total DTIF file set generated by a simulator for a given digital circuit. The 38 data files are organized into four functional groups. Each group portrays the functional role as it relates to UUT testing. They are

a) UUT Model Group (15 data files);
b) Stimulus and Response Group (9 data files);
c) Fault Dictionary Group (6 data files);
d) Probe Group (8 data files).

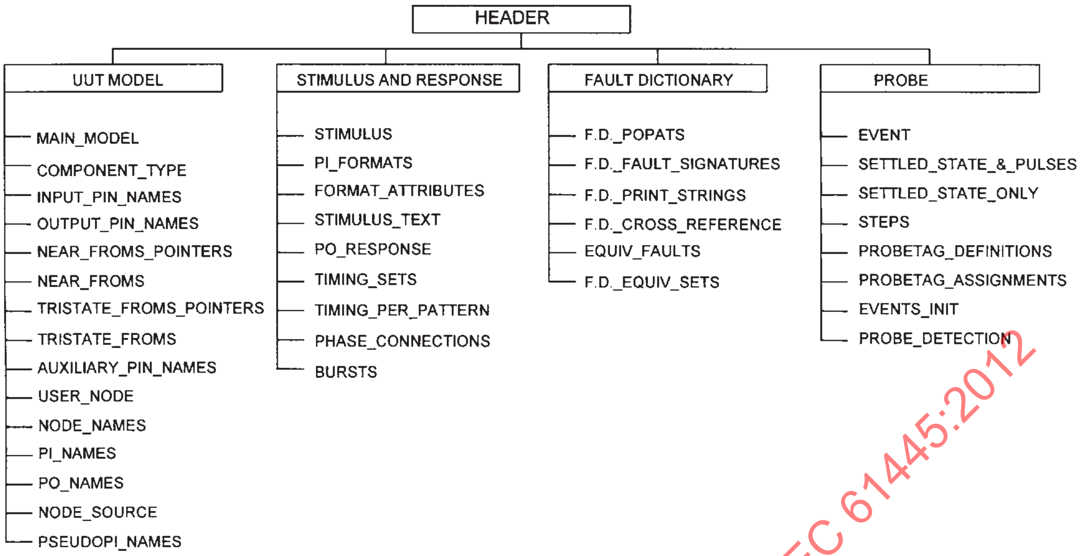

Figure 1—DTIF data files

4.1 UUT Model Group

The UUT Model Group consists of 15 data files that define the UUT Model topology. These files are used to identify the UUT/ATE interface pins, identify all the devices on the UUT, define the interconnections between devices and the structural dependencies between the inputs and outputs of a device, and to identify signal driving sources and their fan-out. They also define functional dependencies of device outputs to their inputs. The information contained in the UUT Model Group is used to develop both end-to-end and diagnostic test programs.

4.2 Stimulus and Response Group

The Stimulus and Response Group consists of 9 files. Data in these files are used to define the logic value of applied stimulus and observed good circuit response. They identify the timing of stimulus edge transitions within a pattern and the period of valid output responses within a pattern. Another function of the Stimulus and Response files is to identify groups of UUT pins with the same stimulus and response timing characteristics.

4.3 Fault Dictionary Group

The Fault Dictionary Group contains a total of 6 files. Data in these files are used on ATE to diagnose UUT failures with the fault dictionary technique. With fault dictionaries, failing responses from the UUT, captured by the ATE, are compared with the fault signatures generated in the fault simulation process and stored in ATE memory. This group identifies all faults and groups of equivalent faults, fault signatures for all detected faults, and the faults grouped within a specific fault set.

4.4 Probe Group

The Probe Group consists of a total of 8 files. They contain all the necessary information to generate probe diagnostics for an ATE. This information includes a complete history of the logic state activity and signal timing at every device pin on the UUT, and data that assists in specifying probe window placement and timing.

5. File specifications

In this clause, the formats for each of the 39 files specified by the DTIF standard are defined. Each file consists of two or more records. One record is a line up to 80 bytes long. The records are of varying size. When writing files, trailing blanks are truncated from the records to reduce the size of the resultant file. The first record of each file shall be a Header record. All data shall be in ASCII format. The DTIF file names are not case sensitive.

All alphanumeric (character) fields are left justified. They are identified in every file description by the character 'A', and the size of each field is defined by a numeric value immediately following the character 'A.' For example, (A4) describes a four-character alphanumeric field.

All integer fields are right justified. They are identified in every file description by the character 'I,' and the size of each field is defined by an integer value immediately following the character 'I.' For example, (I6) describes a six-character integer field.

Should it be required to repeat a particular field within the file description, a multiplier may be used. For example, 5(A16) describe a sixteen-character alphanumeric field that is repeated five times.

The following pages describe the DTIF file data specifications. Each subclause includes an example specific to that file to aid in understanding. The data itself has no relationship to any other DTIF file example.

5.1 HEADER file

File type name: HEADER
File name: header.tap
File number: 1

Description: This file provides a configuration accounting for all of the files that comprise the complete data set for a particular UUT.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)
Line 2:          Columns  1:24 – Model compilation date and time (A24) (note 1)
Line 3:          Columns  1:10 – Number of primary inputs (I10)
Line 4:          Columns  1:10 – Number of primary outputs (I10)
Line 5:          Columns  1:10 – Number of patterns (I10)
Line 6:          Columns  1:10 – Number of packages (I10)
Line 7:          Columns  1:10 – Number of main model (user) defined components
                                 (I10)
Line 8:          Columns  1:10 – Number of component types (I10)
Line 9:          Not used
Line 10:         Not used
Line 11:         Columns  1:10 – Number of wired nets (I10)
Line 12:         Columns  1:10 – Number of user nodes (I10)
Line 13:         Columns  1:10 – Highest user node number (I10) (note 2)
Line 14:         Columns  1:10 – Number of DTIF files generated (I10)
Line 15:         Columns  1:10 – Number of original faults considered (I10)
Line 16:         Columns  1:10 – Number of final definite detects (I10)
Line 17:         Columns  1:10 – Number of final possible detects (I10)
Line 18:         Columns  1:10 – Pattern detection limit (maximum # of times a
                                 fault is detected) (I10)
Line 19-29:      Not used.
Line 30 to N:    Columns  1:24 – DTIF file type name(A24) (note 3)
                 Columns 25:27 – DTIF file type number (I3)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The number of user nodes and the highest user node number may not be one and the same. There may be gaps.
3—One line is used for each file type.

Example:

```
HEADER                  1   2UnitUnderTestAA         10-JAN-2000 12:00
1-JAN-2000
        10
         5
        15
        20
        20
         8


         7
        75
       115
        39
       350
       225
        50
```

8

HEADER	1
STIMULUS	2
PO_RESPONSE	3
PI_NAMES	4
PO_NAMES	5
MAIN_MODEL	6
COMPONENT_TYPE	7
USER_NODE	8
INPUT_PIN_NAMES	9
OUTPUT_PIN_NAMES	10
NEAR_FROMS_POINTERS	11
NEAR_FROMS	12
EVENT	13
SETTLED_STATE_ONLY	14
SETTLED_STATE_&_PULSES	15
NODE_SOURCE	16
STEPS	17
F.D._POPATS	18
F.D._FAULT_SIGNATURES	19
F.D._PRINT_STRINGS	20
TRISTATE_FROMS_POINTERS	21
TRISTATE_FROMS	22
PSEUDOPI_NAMES	23
TIMING_SETS	24
TIMING_PER_PATTERN	25
PHASE_CONNECTIONS	26
AUXILIARY_PIN_NAMES	27
PI_FORMATS	28
FORMAT_ATTRIBUTES	29
F.D._CROSS_REFERENCE	30
PROBETAG_DEFINITIONS	31
PROBETAG_ASSIGNMENTS	32
BURSTS	33
STIMULUS_TEXT	34
NODE_NAMES	35
EVENTS_INIT	36
EQUIV_FAULTS	37
PROBE_DETECTION	38
F.D._EQUIV_SETS	39

5.2 STIMULUS file

File type name: STIMULUS
File name: stimulus.tap
File number: 2

Description: The STIMULUS file contains all the input pin states required to test a UUT. Stimuli are limited to four states. They are unknown (X), high impedance (Z), low (0), and high (1). These logic states are represented by numeric characters and they are: 1=X, 2=Z, 3=Low, and 4=High. Pseudo input pins that have permanently assigned states are used as sources for fixed logic states (i.e., gnd) and are defined in the PSEUDOPI_NAMES file.

When a TSETS file is assigned, patterns in this file represent a dynamic pattern set. Without the TSETS file, they would represent a static stimulus set.

Format:

```
Line 1:         Columns  1:24 – DTIF file type name (A24)
                Columns 25:27 – DTIF file number (I3)
                Columns 28:31 – File version number (optional value)(I4)
                Columns 32:55 – UUT name (A24)
                Columns 56:72 – File creation date and time (A17) (note 1)
                Columns 73:77 – Either spaces or "ERROR" to indicate an
                                error in file generation (A5)

Line 2:         Columns  1:10 – Number of primary inputs (I10)
                Columns 11:20 – Number of test patterns (I10)
                Columns 21:30 – Number of lines per pattern (I10)
                Columns 31:40 – Number of lines containing pattern
                                information (I10)

Line 3 to N:    Columns  1:80 – Stimulus data (note 2)
```

NOTES

1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

2—Stimulus data is stored using the input pin order listed in PI_NAMES file. A new line is required for every pattern; however, a pattern greater than 80 pins would continue on the next line.

Example:

```
STIMULUS                  2   2UnitUnderTestAB         10-JAN-2000 12:00
        86         4         2         8
34343431243212111343323212232343212112213433332232132322122322122223334223223243
223432
31232221111134332321223234323323433211122234322121244323112343321232123322341122
343343
44343332322111212443222112122221331333422211232342123442231233312443323211123333
321122
43444332211122123344323222212211233342223344433211212332332332113212122333233322
233233
```

5.3 PO_RESPONSE file

File type name: PO_RESPONSE
File name: response.tap
File number: 3

Description: The PO_RESPONSE file contains all of the expected output pin response states required to compare to that of the UUT. Responses are limited to four states. They are unknown (X), high impedance (Z), low (0), and high (1). These logic states are represented by numeric characters rather than alphabetic characters, and they are: 1=X, 2=Z, 3=Low, and 4=High.

Format:

```
Line 1:         Columns  1:24 – DTIF file type name (A24)
                Columns 25:27 – DTIF file number (I3)
                Columns 28:31 – File version number (optional value)(I4)
                Columns 32:55 – UUT name (A24)
                Columns 56:72 – File creation date and time (A17) (note 1)
                Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                file generation (A5)

Line 2:         Columns  1:10 – Number of primary outputs (I10)
                Columns 11:20 – Number of patterns (I10)
                Columns 21:30 – Number of lines per pattern (I10)
                Columns 31:40 – Number of lines containing pattern
                                information (I10)

Line 3 to N:    Columns  1:80 – Response data (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—Response data is stored using the output pin order listed in PO_NAMES file. A new line is required for every pattern; however, a pattern greater than 80 pins would continue on the next line.

Example:

```
PO_RESPONSE               3   2UnitUnderTestAC         10-JAN-2000 12:00
         3         5         1         5
341
413
334
443
344
```

5.4 PI_NAMES file

File type name: PI_NAMES
File name: pinames.tap
File number: 4

Description: This file identifies each primary input pin by name and by user node number. The name of the primary input pin for each user node is typically the schematic or etched board name. Each primary input pin must have a name. This name is used for I/O connection and in probe algorithms. Primary bus I/O pins are considered in both the PI_NAMES file and the PO_NAMES file. All primary input pins and output pins not connected to any other I/O pin are in connectivity group 0. Bus pins or I/O pins are assigned a unique connectivity group number starting with 1. Identical connectivity group numbers are assigned to I/O pins when they appear in both the PO_NAMES file and the PI_NAMES file. In addition, there is no requirement that the name of the PI and PO be the same.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)

Line 2:          Columns  1:10 – Total number of primary input pins (I10)
                 Columns 11:16 – Total number of I/O connectivity groups
                                 (I6)(note 2)

Line 3 to N:     Columns  1:24 – Primary input name (A24) (note 3)
                 Columns 25:29 – User node number (I5)
                 Columns 30:34 – Connectivity group number (I5)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—This number is one larger than the highest numbered group since there is a pseudo group 0 representing all PIs and POs that are not connected to any other.
3—The information for each primary input pin is contained on its own line.

Example:

```
PI_NAMES                  4   2UnitUnderTestAD         10-JAN-2000 12:00
         6     3
WR                          1    0
RESET                       2    0
AD_LOAD                     3    0
AD_CLK                      4    0
DATA2                       5    1
DATA1                       6    2
```

5.5 PO_NAMES file

File type name: PO_NAMES
File name: ponames.tap
File number: 5

Description: This file identifies each primary output pin by name and by user node number. The name of the primary output pin for each user node is typically the schematic or etched board name. Each primary output pin must have a name. This name is used for I/O connection and in probe algorithms. User node numbers are sequentially assigned by a model compiler. Primary bus I/O pins are considered in both the PI_NAMES file and the PO_NAMES file. All primary input pins and output pins not connected to any other I/O pin are in connectivity group 0. Bus pins or I/O pins are assigned a connectivity group number starting with 1. Identical connectivity group numbers are assigned to I/O pins when they appear in both the PI_NAMES file and the PO_NAMES file. In addition, there is no requirement that the name of the PI and PO be the same.

Format:

```
Line 1:         Columns  1:24 – DTIF file type name (A24)
                Columns 25:27 – DTIF file number (I3)
                Columns 28:31 – File version number (optional value)(I4)
                Columns 32:55 – UUT name (A24)
                Columns 56:72 – File creation date and time (A17) (note 1)
                Columns 73:77 – Either spaces or "ERROR" to indicate an
                                error in file generation (A5)

Line 2:         Columns  1:10 – Total number of primary output pins (I10)
                Columns 11:16 – Total number of I/O connectivity groups
                                (I6) (note 2)

Line 3 to N:    Columns  1:24 – Primary output name (A24) (note 3)
                Columns 25:29 – User node number (I5)
                Columns 30:34 – Connectivity group number (I5)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—This number is one larger than the highest numbered group, since there is a pseudo group 0 representing all PIs and POs that are not connected to any other.
3—The information for each primary output pin is contained on its own line.

Example:

```
PO_NAMES                  5   2UnitUnderTestAE         10-JAN-2000 12:00
         4     5
DATA4                     100    1
DATA3                     101    2
DATA2                     102    3
DATA1                     103    4
```

5.6 MAIN_MODEL file

File type name: MAIN_MODEL
File name: mainmodel.tap
File number: 6

Description: The Main Model file is a description of the UUT that is simulated and contains several major elements. They are component types, component pins, package numbers, user node numbers, and user node names. A component type and its pins are defined once for each component. When used in the main model of a UUT, the components are assigned unique package names as well as package numbers to distinguish their multiple usage in the UUT main model. Each connection from package pin to package pin(s) must have a unique user node number, and can optionally have a user node name. The combination of package names, component pins, and user node numbers completely defines the schematic interconnection of the UUT model.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an
                                 error in file generation (A5)

Line 2:          Columns  1:10 – Number of circuit packages(I10) (note 2)

Line 3 to N:     Columns  1:24 – The package name (A24)
                 Columns 25:28 – The package type number (I4) (note 3)
                 Columns 29:32 – Number of inputs on the package (I4)
                 Columns 33:36 – Number of outputs on the package (I4)
                 Columns 37:42 – Index for the first input pin in the
                                 USER_NODE file (I6) (note 4)
                 Columns 43:48 – Index for the first output pin in the
                                 USER_NODE file (I6) (note 4)
                 Columns 49:58 – User component number of this package (I10)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—This number includes wired nets, Z-components, and all probeable components on the UUT.
3—This number is an index into the COMPONENT_TYPE file.
4—'Index' means 'index into the relevant section of the USER_NODE file's body' starting from 1.

Example:

```
MAIN_MODEL                6   2UnitUnderTestAF         10-JAN-2000 12:00
         6
U1                         1   2   1     1     3          1
U2                         2   1   1     4     5          2
U3                         2   1   1     6     7          3
R1                         5   1   1     8     9          4
R2                         5   1   1    10    11          5
R3                         5   1   1    12    13          6
```

5.7 COMPONENT_TYPE file

File type name: COMPONENT_TYPE
File name: types.tap
File number: 7

Description: This file sequentially lists every component type by name, the corresponding number of input and output pins, and pointers to where those pins may be found in other DTIF files in the set.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an
                                 error in file generation (A5)

Line 2:          Columns  1:10 – Number of component types (I10) (note 2)

Line 3 to N:     Columns  1:24 – Component type name (A24)
                 Columns 25:28 – Number of inputs for that component type (I4)
                 Columns 29:32 – Number of outputs for the component type (I4)
                 Columns 33:38 – Index of the first input pin name in the
                                 INPUT_PIN_NAMES file (I6) (note 3)
                 Columns 39:44 – Index of the first output pin name in the
                                 OUTPUT_PIN_NAMES file (I6) (note 3)
                 Columns 45:50 – Line number of the first auxiliary pin in
                                 the AUXILIARY_PIN_NAMES file (I6) (note 4)
                 Column     51 – Either a 'W' for wired net component or
                                 blank for all others (A1)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—This number includes wired nets, Z-components, and all probeable components on the UUT.
3—'Index' means 'index into the relevant section of the USER NODE file's body' starting from 1.
4—This is the line where the ground and power pin names are given, not counting the first two lines in the file which are the header-type lines, thus an index value of 1 refers to the third actual line in the AUXILIARY_PIN_NAMES file.

Example:

```
COMPONENT_TYPE           7   2UnitUnderTestAG         10-JAN-2000 12:00
         4
SN74LS32                  8   4     1     1     1
8216                      6   8     9     5     2
PULLUP                    1   1    15    13     3
WN                        3   1    16    14     4W
```

5.8 USER_NODE file

File type name: USER_NODE
File name: usernodes.tap
File number: 8

Description: This file sequentially lists the user node numbers assigned to each package found in the MAIN_MODEL. The order is to first list all input pins found in the INPUT_PIN_NAMES file followed by all output pins found in the OUTPUT_PIN_NAMES file for each package in ascending order (package 1, package 2, etc.). If wired nets are used in the UUT, there will be more user nodes than physical nodes to accommodate the bidirectional workaround associated with wired nets. The USER_NODE file used in conjunction with the MAIN_MODEL file represents a netlist for the circuit.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an
                                 error in file generation (A5)

Line 2:          Columns  1:10 – The number of unique user nodes in the
                                 circuit (I10)
                 Columns 11:20 – Number of lines used to hold all of the user
                                 nodes (I10)

Line 3 to N:     Columns  1:80 – User node numbers (8(I10))
```

NOTE
1 — Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

Example:

```
USER_NODE                 8   2UnitUnderTestAH         10-JAN-2000 12:00
        25         4
         3        15        16        17        18        19        20        21
         5        22        23        24         1        20        21        22
        23        24        25         4        10        11        12        13
         6         7        14        15        16        17        18
```

5.9 INPUT_PIN_NAMES file

File type name: INPUT_PIN_NAMES
File name: inputpins.tap
File number: 9

Description: This file provides a list of input pin names for every component as listed in the COMPONENT_TYPE file. The input pin names are arranged in the same order that the input pins are defined.

Format:

```
Line 1:         Columns  1:24 – DTIF file type name (A24)
                Columns 25:27 – DTIF file number (I3)
                Columns 28:31 – File version number (optional value)(I4)
                Columns 32:55 – UUT name (A24)
                Columns 56:72 – File creation date and time (A17) (note 1)
                Columns 73:77 – Either spaces or "ERROR" to indicate an
                                error in file generation (A5)

Line 2:         Columns  1:10 – Number of input pin names in the circuit (I10)
                Columns 11:20 – Number of lines that contain pin names (I10)

Line 3 to N:    Columns  1:80 – Input pin names (5(A16))
```

NOTE
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

Example:

```
INPUT_PIN_NAMES           9   2UnitUnderTestAI         10-JAN-2000 12:00
        17         4
               1               2               3               1               2
               1               1               2               3               4
               5               8               9               1               3
               4               7
```

5.10 OUTPUT_PIN_NAMES file

File type name: OUTPUT_PIN_NAMES
File name: outputpin.tap
File number: 10

Description: This file provides a list of output pin names for every component as listed in the COMPONENT_TYPE file. They are arranged in the same order that the output pins are defined.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an
                                 error in file generation (A5)

Line 2:          Columns  1:10 – Number of output pin names in the circuit (I10)
                 Columns 11:20 – Number of lines that contain pin names (I10)

Line 3 to N:     Columns  1:80 – Output Pin Names (5(A16))
```

NOTE
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

Example:

```
OUTPUT_PIN_NAMES         10   2UnitUnderTestAJ         10-JAN-2000 12:00
        14         3
              4               5               6               8               9
              3               6               8              11               2
              4               6               4               5
```

5.11 NEAR_FROMS_POINTERS file

File type name: NEAR_FROMS_POINTERS
File name: nerfrmptr.tap
File number: 11

Description: NEAR_FROMS are inputs of a device which are associated with an output of that device such that if the output ever fails, and all the NEAR_FROMS of that output are good, then it is the device itself that is bad. Each device output pin has one NEAR_FROMS pointer and uses the same order as in the OUTPUT_PIN_NAMES file. The data in NEAR_FROMS_POINTERS defines a NEAR_FROMS packet and is structured as groups of two integers, where the first integer indicates the number of NEAR_FROMS in the packet associated with the device's output pin, and the second integer is the index into the NEAR_FROMS file pointing to the first NEAR_FROMS in the packet.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an
                                 error in file generation (A5)

Line 2 to N:     Columns  1:80 – NEAR_FROMS pointer packages (8(I4,I6)) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—Each pointer consists of two integers. The first integer (4 columns) is the number of NEAR_FROMS associated with the packet's output pin, and the second integer (6 columns) is an index into the NEAR_FROMS file pointing to the first NEAR_FROMS.

Example:

```
NEAR_FROMS_POINTERS       11   2UnitUnderTestAK         10-JAN-2000 12:00
   1     1   2     2   3     4   4     7   2    11   4    13   1    14   3    17
   2    20   1    22   3    23   5    26   8    31   2    34   3    36   2    39
   4    41   2    45   1    47   3    48   5    51
```

5.12 NEAR_FROMS file

File type name: NEAR_FROMS
File name: nearfroms.tap
File number: 12

Description: NEAR_FROMS are inputs of a device which are associated with an output of that device such that if the output ever fails, and all the NEAR_FROMS of that output are good, then it is the device itself that is bad. NEAR_FROMS may include unbuffered outputs.

The integers in this file are organized in packets, the size of which is defined by the NEAR_FROMS_POINTERS file. For each packet, the NEAR_FROMS integers in each packet are an index into the INPUT_PIN_NAMES (for positive integers) and OUTPUT_PIN_NAMES (for negative integers) files. Since every NEAR_FROMS packet is associated with a specific device output, the NEAR_FROMS for the output specify the device inputs that would be probed to determine the location of the failure.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an
                                 error in file generation (A5)

Line 2 to N:     Columns  1:80 – NEAR_FROMS codes (20(I4)) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The NEAR_FROMS codes are as follows:
+N → N is a pointer into the INPUT_PIN_NAMES file. This pointer assumes that the starting position is the pointer from the COMPONENT_TYPE file minus 1.
–N → N is a pointer into the OUTPUT_PIN_NAMES file. This pointer assumes that the starting position is the pointer from the COMPONENT_TYPE file minus 1.

Example:

```
NEAR_FROMS               12   2UnitUnderTestAL        10-JAN-2000 12:00
   1  -1   1  -2  -1   1  -3  -2  -1   1  -4  -3  -2  -1   1  -5  -4  -3  -2  -1
   1  -6  -5  -4  -3  -2  -1   1  -7  -6  -5  -4  -3  -2  -1   1  -8  -7  -6  -5
  -4  -3  -2  -1   1  -9  -8  -7  -6  -5  -4  -3  -2  -1   1   1   2   3   4   5
   6   7   8   1   2   3   4   5   6   7   8   1   2   3   4   5   6   1   2   3
   4   5   6   7   8   6  -5   5   1   6  -6   5   2   6  -7   5   3   6  -8   5
   4   5   6   1   5   6   2   5   6   3   5   6   4   1   2   3   4   5   6   7
   8   9  10  11  12   1   2   3   4   5   6   7   8   9  10  11  12   1   2   3
   4   5   6   7   8   9  10  11  12   1   2   3   4   5   6   7   8   9  10  11
  12   2   1   3   8   7   9   2   1   4   8   7   9   3  -1   2   1   5   8   7
   9   3  -1   4  -2   2   1   6   8   7   9   5  -3   4  -2   3  -1   9   1   2
   3  -1   4  -2   5  -3   6  -4   8   7   1   1   2   3
```

5.13 EVENT file

File type name: EVENT
File name: events.tap
File number: 13

Description: This file presents every logical event which the simulator predicts at every node in the UUT model for each pattern. The fidelity of this file is dependent on the timing resolution specified on line 2.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an
                                 error in file generation (A5)

Line 2:          Columns  1:10 – Timing resolution value(I10) (note 2)
                 Columns 11:14 – Unit of time (I4) (note 2)

Line 3 to N:     Columns  1:80 – Contiguous stream of events data (note 3)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The timing resolution value is a positive integer, that in conjunction with the units of time, represents the Simulation Time Unit (STU). The unit of time is the exponent, base 10, required to calculate the given time unit starting from seconds. If the resolution is 5 and the units of time is –9, the STU is 5 nanoseconds.
3—An event is composed of a two-part data grouping consisting of an integer followed by an alphanumeric operational code (op code), which determines the meaning of the integer value. The integer value represents either time in STU, node number, or a pattern number depending on the op code it precedes. Time values are interpreted as either delta time from the last event or absolute time from the beginning of the pattern depending upon the op code. If the integer value is missing, event time is assumed to be the same as the most recent time value within the pattern. The single-character op codes are as follows:

op code	Meaning
@	Absolute time relative to start of a pattern
A	A relative (delta) time field; specifying the minimum time at which the nodal event might occur. The specified nodes are in a stable state prior to this point in time.
B	A relative (delta) time field; specifying the minimum time at which the nodal event might occur. All of the specified nodes are in an unstable state prior to this point in time.
C	A relative (delta) time field; specifying the maximum time at which the nodal event might occur. These specified nodes are in a stable state after this point in time.
D	A relative (delta) time field; specifying the maximum time at which the nodal event might occur. All of these specified nodes are in an unstable state after this point in time.
E	User node number that just went 'low' or '0'
F	Unused
G	Unused
H	User node number that just went 'high' or '1'
I	Unused
J	Unused
K	User node number that just went to 'Z' or high impedance
L	Unused
M	Unused
N	User node number that just went to 'X' or unknown
O	Unused
P	End of pattern
Q	End of Burst
R	End-of-file (This should never be preceded by an integer.)

Example:

```
EVENT                    13   2UnitUnderTestAM        10-JAN-2000 12:00
         1 -12
11000A2H5H9H10H11H12H13H14H15H16H17H18HC2H5H9H10H11H12H13H14H14H16H17H18H6667A51
E64E2242A48H3158A61H2933C51E64E7000C48H7567A73K74K75K76K433C61H35000C73K74K75K76
K24000A3H7HC3H7H50000A8HC8H50000A8EC8E200000A19H20H21H22HC19H20H21H22H10A100H101
H102H103HC100H101H102H103H12030A69N70N71N72N17970C69N70N71N72N69990A1H4H6HC1H4H6
H6667A33H63E5400A62H1263A34HC33H1670C63E5000A35H6670A36HC34H2897A77K78K79K80K10A
104N105N106N107N423C62H3330A37H6670A38HC35H6670A39H6660A40HC36H6670A41H5000C77K7
8K79K80K10C104N105N106N107N1660A42HC37H2239A47H4267A43H4267A44H2400A57E2A69N70N7
1N72NC69N70N71N72N10A104N105N106N107NC104N105N106N107N145C38H4757A65K66K67K68K37
19A69K70K71K72K10A104K105K106K107K4844C39H13370C40H13300C41H13300C42H8700C47H150
00C43H15000C44H57E35000C65K66K67K68K69K70K71K72K10C104K105K106K107K242990A19K20K
21K22KC19K20K21K22K1000000@1P100000A3EC3E14290A81E82E83E84E86E87E88E89E91E92E93E
94E3710A85E90E95E10000C81E82E83E84E86E87E88E89E91E92E93E94E4000C85E90E95E18000A8
HC8H50000A8EC8E50000A3HC3H150000A19H20H21H22HC19H20H21H22H550000A19K20K21K22KC19
K20K21K22K1000000@2P1QR
```

5.14 SETTLED_STATE_ONLY file

File type name: SETTLED_STATE_ONLY
File name: setldonly.tap
File number: 14

Description: This file is an incremental representation of the settled state of each node in the circuit at the end of each pattern, assuming that there is always sufficient time for all nodes to settle to one of the four basic states (X, Z, 0, and 1). Only those nodes that changed state for each new pattern are reported.

Format:

```
Line 1:          Columns  1:24 - DTIF file type name (A24)
                 Columns 25:27 - DTIF file number (I3)
                 Columns 28:31 - File version number (optional value)(I4)
                 Columns 32:55 - UUT name (A24)
                 Columns 56:72 - File creation date and time (A17) (note 1)
                 Columns 73:77 - Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)

Line 2 to N:     Columns  1:80 - Contains the final state of the node (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—Only those nodes that changed state are reported. The data is represented as a series of op codes followed by integer fields. Leading blanks are suppressed. The op codes are as follows:

op code	Meaning
1	Node settled to ‘X’ state
2	Node settled to ‘Z’ state
3	Node settled to ‘0’ state
4	Node settled to ‘1’ state
5	Not Used
6	Not Used
7	Not Used
8	End of pattern
9	Start of burst marker
10	End of burst marker
11	End of event file
* (asterisk)	The next number is an op code
. (period)	The next number is a count field, showing the number of user nodes that follow the count field
(space)	The next number is a user node number

Example:

```
SETTLED_STATE_ONLY       14   2UnitUnderTestAN         10-JAN-2000 12:00
*9*1.15 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95*2.24 19 20 21 22 65 66 67 6
8 69 70 71 72 73 74 75 76 77 78 79 80 104 105 106 107*3.13 8 51 52 53 54 55 56 5
7 58 59 60 63 64*4.11 35 36 37 38 39 40 41 42 43 44 45*8*4.1 81*8*3.4 77 79 104
106*4.2 80 83*8*10*11
```

5.15 SETTLED_STATE_&_PULSES file

File type name: SETTLED_STATE_&_PULSES
File name: setldpuls.tap
File number: 15

Description: This file is an incremental representation of the settled state of the circuit according to the simulator, meaning that only nodes that change their behavior from one pattern to the next are identified. It presents the node state, pulse, no pulse, and possible pulse data for each node that transitions on each pattern. It is assumed that sufficient time has been allocated to allow all of the nodes to settle out, and that their state is determined by a strobe at the end of the pattern. It is assumed that the pulse can occur anywhere within the pattern, and still be captured by the probe.

Format:

```
Line 1:         Columns  1:24 - DTIF file type name (A24)
                Columns 25:27 - DTIF file number (I3)
                Columns 28:31 - File version number (optional value) (I4)
                Columns 32:55 - UUT name (A24)
                Columns 56:72 - File creation date and time (A17) (note 1)
                Columns 73:77 - Either spaces or "ERROR" to indicate an error in
                                file generation (A5)

Line 2:         Columns  1:10 - Contains the minimum safe pulse width (I10) (note 2)

Line 3 to N:    Columns  1:80 - Contains SETTLED_STATE_ONLY and pulse data (note 3)
```

NOTES

1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

2—Minimum safe pulse width is expressed in STU as the minimum duration for which a pulse must transition to a known (Hi or Lo) state and return to be definitely detected. If a node transitions from Hi to Lo, or Lo to Hi only once, without entering the X or Z state at any time within a pattern, it is considered to be pulseless. If a node does not fit into either of these categories (i.e., transitions to X, Z, or pulses for less than the minimum time required), then it "possibly pulses."

3—The data is identical to that in the SETTLED_STATE_ONLY file except that the pulse information is included. The op codes for this file are an extension of the SETTLED_STATE_ONLY file. They are as follows:

op code	Meaning
(State that the node settled without pulsing)	
1	X
2	Z
3	0
4	1
5	Not Used
6	Not Used
7	Not Used
8	End of pattern
9	Start burst marker
10	End burst marker
11	End of event file
* (asterisk)	The next number is an op code
. (period)	The next number is a count field, showing the number of user nodes that follow the count
(space)	The next number is a user node number
(State that the node settled *but* it *definitely* pulsed)	
12	X
13	Z
14	0
15	1
(State that the node settled *and* it *possibly* pulsed)	
16	X
17	Z
18	0
19	1

Example:

```
SETTLED_STATE_&_PULSES    15    2UnitUnderTestAO         10-JAN-2000 12:00
        10
*9*3.7 8 51 52 55 56 60 64*4.15 1 2 3 7 9 10 13 14 15 42 45 46 48 49 50 98*16.10
 81 82 83 84 85 86 91 92 93 94*17.15 19 20 21 22 70 71 72 73 74 75 76 77 78 79 8
0*18.2 57 63*19.11 33 34 35 36 37 38 39 40 41 43 44*8*3.9 57 63 81 82 83 84 85 8
6 95*4.13 33 34 35 36 37 38 39 40 41 43 44 47 62*15.1 3*8*4.2 3 81*8*3.4 77 79 1
04 106*4.7 78 80 82 83 84 85 107*14.1 63*15.5 1 6 7 44 62*8*3.12 9 10 11 12 13 1
4 15 16 17 18 78 107*4.4 77 79 104 106*883.2 79 106*4.2 18 81*8*3.4 18 77 81 104 *4.4
17 78 82 105*8*3.4 17 78 82 105*4.4 16 79 83 106*8*3.4 16 79 83 106*4.4 15 80 84
107*8*4.11 1 9 10 11 12 13 14 78 81 82 83*14.1 57*15.8 4 33 34 35 36 37 39 47*8*3.10 9
10 11 12 13 14 15 16 17 107*4.4 77 79 104 106*8*3.2 79 106*4.2 18 81 *8*3.4 18 77 81
104*4.4 17 78 82 105*8*3.4 17 78 82 105*4.4 16 79 83 106*8*3.4 1 6 79 83 106*4.4 15 80
84 107*8*10*11
```

5.16 NODE_SOURCE file

File type name: NODE_SOURCE
File name: nodsource.tap
File number: 16

Description: This file provides the information necessary for each user node and identifies which output on the package (or PI) drives or sources that node. This information is depicted in pairs of numbers. The first number of each pair identifies the package number and the second number of each pair is a pointer to the OUTPUT_PIN_NAMES file where the source pin name is located. To use this file to find the source of a user node, use the first number of the pair for the package number in the MAIN_MODEL file to find both the package name and the component type. The component type leads to the COMPONENT_TYPE file which will in turn lead to the OUTPUT_PIN_NAMES file to identify the appropriate pin name and the INPUT_PIN_NAMES file for the node source information.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)

Line 2:          Columns  1:10 – Number of user nodes (I10)

Line 3 to N:     Columns  1:80 – Node source packets (10(I5,I3)) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The first integer of the packet is a main model component number, or zero. The second integer of the packet is an index to the OUTPUT_PIN_NAMES file (the value of the index is the starting position in the group of pins in the OUTPUT_PIN_NAMES file pointed to by the COMPONENT_TYPE file) OR if the first integer is zero, the second integer of the packet is a PI number or the number of a pseudo PI plus the number of PIs (if the second number of the packet is greater than the number of PIs, subtract the number of PIs to get the pseudo PI number).

Example:

```
NODE_SOURCE              16   2UnitUnderTestOP         10-JAN-2000 12:00
        25
    0  1    0  2    0  3    0  4    0  5    0  6    0  7    0  8    0  9    0 10
    0  0    0  0    0  0    0  0    0  0    1  1    1  2    1  3    2  1    2  2
    3  1    3  2    0 15    0 17    0 18
```

5.17 STEPS file

File type name: STEPS
File name: steps.tap
File number: 17

Description: This file provides information defining which patterns are useful for fault dictionary and probe based fault isolation testing of the UUT. It lists only those patterns that have useful fault information associated with them, i.e., only those patterns that detect previously undetected faults at internal package pins or POs, and those patterns that subdivide previously detected fault sets at POs. Only those patterns included here need to be looked at with a probe, since only those patterns contain information that can diagnose a fault. A step is a pattern that is necessary for diagnosing faults for which logic state information about probeable nodes should be available to the test job, specifically the probe. This file applies only to faults simulated and if the probe method is a strobe at the end of the pattern.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)

Line 2:          Columns  1:10 – Total number of steps (test patterns)(I10)

Line 3 to N:     Columns  1:80 – Contains steps (8(I10)) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—Each step entry is a 10-character integer that represents a test pattern number. Positive entries are used when a new fault is detected at a primary output. Negative entries are for use with patterns that make a fault visible for the first time on a probeable node (net).

Example:

```
STEPS                   17    2UnitUnderTestAQ          10-JAN-2000 12:00
         7
         1         2         5        -6        -9        11       -15
```

5.18 F.D._POPATS file

File type name: F.D._POPATS
File name: fdpopats.tap
File number: 18

Description: This file lists all of the PO pins on a pattern-by-pattern basis that were determined by fault simulation to be useful for fault isolation. This information is contained in pairs. The first number of the pair is the PO pin number index into the PO_NAMES file. The second number of the pair is the Pattern number (PAT), i.e., POPAT. The POPAT pairs are numbered sequentially.

Format:

```
Line 1:           Columns  1:24 – DTIF file type name (A24)
                  Columns 25:27 – DTIF file number (I3)
                  Columns 28:31 – File version number (optional value) (I4)
                  Columns 32:55 – UUT name (A24)
                  Columns 56:72 – File creation date and time (A17) (note 1)
                  Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                  file generation (A5)

Line 2:           Columns  1: 5 – Total number of POPATs (I5)

Line 3 to N:      Columns  1:80 – POPAT packets (8(I4,I6)) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The first integer in the packet contains the primary output number, whereas the second integer contains the pattern number.

Example:

```
F.D._POPATS              18   2UnitUnderTestAR          10-JAN-2000 12:00
   15
   1     1   2     1   3     1   4     1   2     4   4     4   1     5   3     5
   3    10   2    11   4    11   4    12   3    13   2    14   1    15
```

5.19 F.D._FAULT_SIGNATURES file

File type name: F.D._FAULT_SIGNATURES
File name: fdfltsig.tap
File number: 19

Description: This file identifies the fault set and the corresponding POPATs that are predicted by the simulator to detect it (its fault signature). Each fault set starts on a new line.

Format:

```
Line 1:         Columns  1:24 – DTIF file type name (A24)
                Columns 25:27 – DTIF file number (I3)
                Columns 28:31 – File version number (optional value)(I4)
                Columns 32:55 – UUT name (A24)
                Columns 56:72 – File creation date and time (A17) (note 1)
                Columns 73:77 –  Either spaces or "ERROR" to indicate an error in
                                file generation (A5)

Line 2:         Columns  1: 6 – Total number of fault sets (I6)
                Columns  7:12 – Most number of POPATs for a fault set (I6)

Line 3 to N:    Columns  1: 6 – Fault set Last Analyzed POPAT (FLAP) number (I6)
                                (note 2)
                Columns  7:12 – Total number of POPATs in the signature for this
                                fault set (I6)
                Columns 13:18 – Contains the first POPAT of the fault signature
                                (I6) (note 3)
                Columns 19:24 – Contains the second POPAT of the fault signature
                                (I6) (note 3)
                Columns 25:30 – Contains the third POPAT of the fault signature
                                (I6) (note 3)
                Columns 31:36 – Contains the fourth POPAT of the fault signature
                                (I6) (note 3)
                Columns 37:42 – Contains the fifth POPAT of the fault signature
                                (I6) (note 3)
                Columns 43:48 – Contains the sixth POPAT of the fault signature
                                (I6) (note 3)
                Columns 49:54 – Contains the seventh POPAT of the fault signature
                                (I6) (note 3)
                Columns 55:60 – Contains the eighth POPAT of the fault signature
                                (I6) (note 3)
                Columns 61:66 – Contains the ninth POPAT of the fault signature
                                (I6) (note 3)
                Columns 67:72 – Contains the tenth POPAT of the fault signature
                                (I6) (note 3)
                Columns 73:78 – Contains the eleventh POPAT of the fault signature
                                (I6) (notes 3/4)
```


NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The FLAP number identifies to the user whether or not to use all POPATs for analysis. A positive FLAP number of +N indicates that the first *N* POPATs were useful in fault simulation and that only these *N* POPATs should be used on the tester for fault signature analysis for this fault set (all faults in this fault set were dropped from the simulation process after POPAT *N*). Therefore, no POPATs after *N* should be used in signature analysis for this particular fault set. Signature analysis is the comparison of on-tester error pattern test results to fault simulation results predicted for each fault set. A "–1" indicates that all patterns were used in fault simulation of each fault.
3—A minus "–" sign preceding the POPAT signifies that the POPAT is a possible detect.
4—If more than eleven POPATs are contained in the fault set, they will be continued on the next new line.

Example:

```
F.D._FAULT_SIGNATURES     19    2UnitUnderTestAS          10-JAN-2000 12:00
    10      6
    -1      2      1     15
    -1      3      3      9     13
    -1      1      8
    -1      1     10
    -1      2     10     11
    -1      1     11
    -1      5     10     11     12     13     14
    -1      2     11     12
    -1      1     15
    -1      6      9      10     11     12     13     14
```

5.20 F.D._PRINT_STRINGS file

File type name: F.D._PRINT_STRINGS
File name: fdprint.tap
File number: 20

Description: This file sequentially lists each fault set in terms of faulty component information identified by the fault simulation process. The contained information is used on the tester, after fault signature analysis, to identify the faulty components that should be removed and repaired on the UUT to correct the cause of failure.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)

Line 2:          Columns  1: 6 – Total number of fault sets (I6)
                 Columns  7:10 – Most number of characters in a fault title (I4)
                 Columns 11:15 – Most number of faults in a fault set (I5)

Line 3:          Columns  1: 6 – Number of lines in this fault set (I6)
                 Columns  7:12 – Number of fault titles (fault names) in
                                 this fault set (I6)

Line 4:          Columns  1: 6 – Number of characters in the fault title (I6)
                 Columns  7:80 – The fault title (note 2)

Repeat line 4 until end of fault set. Line 3 and Line 4 loop repeated for all fault
sets. Each fault set can be of any length, occupying one or more lines in the file.
```

NOTES

1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

2—A fault title is specified as an individual fault using point_name and fault state, where point_name is:

- package name and pin of a component.
- a package name followed by a signal name (signifying an input or an output of the package).
- <package.internal>pin name (signifying internal to the package).
- <^>primary I/O name (signifying primary I/O pin).
- signal name at main model level.

Fault state is represented by one of the following:

- Stuck at faults are represented by <point_name>@state, where state is 0, 1, Z.
- Open faults are represented by <point_name>/state, where state is 0, 1, Z. If state is unspecified, it is selected accordingly to the family characteristics of the device.
- Short faults are represented by <point_name>@<point_name>... All point names in the string are shorted together.

Example:

```
F.D._PRINT_STRINGS       20   3UnitUnderTestAT          10-JAN-2000 12:00
     3 12    4
     3     3                       !Fault Set 1
     7<U8>4@0
    10<^>DATA4@0
     7DBUS4@0
     4     4                       !Fault Set 2
     7<U8>7@0
    10<^>DATA3@0
    10<^>DATA2@0
     7DBUS3@0
     3     3                       !Fault Set 3
     7<U8>9@0
    12<^>AD_LOAD/1
    12<^>AD_LOAD@1
```

5.21 TRISTATE_FROMS_POINTERS file

File type name: TRISTATE_FROMS_POINTERS
File name: zfromptrs.tap
File number: 21

Description: TRISTATE_FROMS are inputs of a device that are associated with an output of that device, which are tristate enables for that output. This file is parallel to the OUTPUT_PIN_NAMES file so that each output pin on each component type has an associated TRISTATE_FROMS packet.

The data in the TRISTATE_FROMS_POINTERS file defines a TRISTATE_FROMS packet and is structured as a group of two integers. The first integer indicates the number of TRISTATE_FROMS in the packet associated with the device's output pin, and the second integer is the index into the TRISTATE_FROMS file pointing to the first TRISTATE_FROMS in the packet.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2 to N:     Columns  1:80 – Eight tristate froms pointers (8(I4,I6)) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—Each pointer consists of two integers. The first integer (4 columns) is the number of tristate froms associated with this pointer. The second integer (6 columns) is the index into the TRISTATE_FROMS file for this pointer.

Example:

```
TRISTATE_FROMS_POINTERS  21   2UnitUnderTestAU         10-JAN-2000 12:00
   0     1   0     1   0     1   0     1   0     1   0     1   0     1   0     1
   0     1   0     1   0     1   0     1   0     1   0     1   0     1   0     1
   0     1   0     1   0     1   0     1   0     1   0     1   0     1   0     1
   0     1   0     1   0     1   0     1   0     1   2     3   2     5   2     7
   2     9   2    11   2    13   2    15   2    17   2    19   2    21   2    23
   0    25   0    25   0    25   0    25   0    25   0    25   0    25
```

5.22 TRISTATE_FROMS file

File type name: TRISTATE_FROMS
File name: zfroms.tap
File number: 22

Description: TRISTATE_FROMS are inputs of a device that are associated with an output of that device, which are tristate enables for that output.

The integers in this file are organized in packets, the size of which is defined by the TRISTATE_FROMS_POINTERS file. For each packet, the TRISTATE_FROMS_POINTERS integers in each packet are an index into the INPUT_PIN_NAMES (for positive integers) and OUTPUT_PIN_NAMES (for negative integers) files. Since every TRISTATE_FROMS packet is associated with a specific device output, the TRISTATE_FROMS for the output specify the device inputs that would be probed to determine the location of the failure.

Format:

```
Line 1:         Columns  1:24 – DTIF file type name (A24)
                Columns 25:27 – DTIF file number (I3)
                Columns 28:31 – File version number (optional value)(I4)
                Columns 32:55 – UUT name (A24)
                Columns 56:72 – File creation date and time (A17) (note 1)
                Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                in file generation (A5)

Line 2 to N:    Columns  1:80 – Twenty tristate froms. (20(I4)) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The TRISTATE_FROMS codes are as follows:
+N → N is a pointer into the INPUT_PIN_NAMES file. This pointer assumes that the starting position is the pointer from the COMPONENT_TYPE file minus 1.
–N → N is a pointer into the OUTPUT_PIN_NAMES file. This pointer assumes that the starting position is the pointer from the COMPONENT_TYPE file minus 1.

Example:

```
TRISTATE_FROMS           22   2UnitUnderTestAV          10-JAN-2000 12:00
   6   5   6   5   6   5   6   5   5   6   5   6   5   6   5   6  11  12  11  12
  12  11   1
```

5.23 PSEUDOPI_NAMES file

File type name: PSEUDOPI_NAMES
File name: psupinams.tap
File number: 23

Description: This file contains all of the pseudo PI pin names used in assigning a modeling logic state to package pins that are not connected to other packages. Pseudo PIs are used as sources for fixed logic states. They may be assigned to Vcc, ground, Z, X, floating, or power on reset pulse. Commonly used pseudo PI states are as follows:

$L1	Logic 1
$L1B	Alternate Logic 1
$L0	Logic 0
$L0B	Alternate Logic 0
$LX	Unknown
$LZ	Tristate
$FL1	Floating high
$FL0	Floating low
$FLX	Floating unknown
$PWRON	Power on LO/HI transition

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2:          Columns  1:10 – Total number of pseudo PI pins (I10)

Lines 3 TO N:    Columns  1:24 – Pseudo PI name (A24) (note 2)
                 Columns 25:29 – User node number connected to the specified
                                 pseudo PI (I5)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—Each line has only one pseudo PI name. A new line is started for each new pseudo PI.

Example:

```
PSEUDOPI_NAMES              23   2UnitUnderTestAW          10-JAN-2000 12:00
         3
$L1                          111
$LX                          107
$PWRON                       112
```

5.24 TIMING_SETS file

File type name: TIMING_SETS
File name: timesets.tap
File number: 24

Description: The TIMING_SETS file data is based on the presumption that a timing generator exists in the tester digital hardware that will accept and store data for an optionally defined maximum number of timing sets (TSETs). Only one of these TSETs may be in effect on any given pattern. It, and only it, is the timing source for all of the I/O pins on the tester for that pattern. TSETs allow the user to specify, for each I/O pin, when to start and when to stop applying stimulus or measuring the output response available at that I/O pin. The time period for application of stimulus is called a PHASE and the time period allocated to the observation of a response is called a WINDOW. The PHASE defines the assert and return times. The assert time defines when a logic state will be applied to an input pin and the return time defines when a logic state will be removed from the output pin. After the return time, the input pin will remain in a state defined by the PI_FORMATS and FORMAT_ATTRIBUTES files until the next assert time. Each WINDOW has an open and close time defined for it. A UUT signal must be present from open to close time to be considered a valid state detection. TSETs cannot be changed during a pattern. The timing generator is driven by a clock that has a user defined resolution of typically 1 STU. Each TSET will also have its own clock period based on some multiple of the defined STU.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2:          Columns  1: 5 – Highest TSET number defined (I5)
                 Columns  6:10 – Total number of TSETS (I5)
                 Columns 11:15 – Maximum number of PHASEs per TSET (I5)
                 Columns 16:20 – Maximum number of WINDOWs per TSET (I5)
                 Columns 21:30 – Number of simulation time units (STU) (I10)
                 Columns 31:33 – Timing units (units must match the value in
                                 the EVENTS file) (I3)

Line 3 to N:     Columns  1: 5 – Line type (1,2,3, or 4) (note 2)

        Line Type 1 (TSET header):
                 Columns  6:13 – TSET number (I8)
                 Columns 14:28 – TSET clock period in STU (I15)
                 Columns 29:32 – Number of PHASEs (I4)
                 Columns 33:36 – Number of WINDOWs (I4)

        Line Type 2 (PHASE line):
                 Columns  6:13 – Phase number (I8)
                 Columns 14:21 – TSET number (I8)
                 Columns 22:36 – Assert time in STU (I15)
                 Columns 37:51 – Return time in STU (I15)

        Line Type 3 (WINDOW line):
                 Columns  6:13 – Window number (I8)
                 Columns 14:21 – TSET number (I8)
                 Columns 22:36 – Open time in STU (I15)
                 Columns 37:51 – Close time in STU (I15)
```

```
Line Type 4 (trigger line): (note 3)
        Columns  6:13 - Phase number (I8)
        Columns 14:18 - Trigger type (1: PHASE triggers off the
                        beginning of the pattern, 2:PHASE triggers off the
                        clock pulse)(I5)
```

NOTES

1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

2—No PHASE or WINDOW line will occur before the corresponding TSET HEADER line for that TSET.

3—All timing specifications are integer only. All trigger lines occur at the end of the file.

Example:

```
TIMING_SETS              24   2UnitUnderTestAX         10-JAN-2000 12:00
   32    1   16   16         1-12
    1       1       1000000   2   1
    2       1       1         11000         950000
    2       2       1         40000         950000
    3       1       1         70000         750000
    4       1       1
    4       2       1
```

5.25 TIMING_PER_PATTERN file

File type name: TIMING_PER_PATTERN
File name: timperpat.tap
File number: 25

Description: This file is used to control the selection of one, and only one, TSET at a time for simulation or on-station testing of the UUT.

Format:

```
Line 1:          Columns  1:24 - DTIF file type name (A24)
                 Columns 25:27 - DTIF file number (I3)
                 Columns 28:31 - File version number (optional value)(I4)
                 Columns 32:55 - UUT name (A24)
                 Columns 56:72 - File creation date and time (A17) (note 1)
                 Columns 73:77 - Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2:          Unused

Line 3 to N:     Columns  1:10 - Pattern number (I10) (note 2)
                 Columns 11:18 - TSET number (I8)
                 Columns 19:26 - Clocks per pattern (I8)
                 Columns 27:36 - Pattern number (I10)
                 Columns 37:44 - TSET number (I8)
                 Columns 45:52 - Clocks per pattern (I8)
                 Columns 53:62 - Pattern number (I10)
                 Columns 63:70 - TSET number (I8)
                 Columns 71:78 - Clocks per pattern (I8)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—There are three entries per line with the last two character positions on the line unused.

Example:

```
TIMING_PER_PATTERN       25   1UnitUnderTestAY          10-JAN-2000 12:00

         1       1       1        16       1       1
```

5.26 PHASE_CONNECTIONS file

File type name: PHASE_CONNECTIONS
File name: phaseconn.tap
File number: 26

Description: This file provides the assignment of primary input and output pins to PHASEs and WINDOWs produced by the TIMING_SETS file.

Format:

```
Line 1:            Columns  1:24 – DTIF file type name (A24)
                   Columns 25:27 – DTIF file number (I3)
                   Columns 28:31 – File version number (optional value)(I4)
                   Columns 32:55 – UUT name (A24)
                   Columns 56:72 – File creation date and time (A17) (note 1)
                   Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                   file generation (A5)

Line 2:            Columns  1: 5 – Number of lines in the file giving PI PHASE
                                   assignments (I5) (note 2)
                   Columns  6:10 – Number of lines in the file giving PO WINDOW
                                   assignments (I5) (note 2)
                   Columns 11:15 – Number of PIs that have assigned PHASEs (I5)
                   Columns 16:20 – Number of POs that have assigned WINDOWs (I5)
                                   (note 3)

Line 3 to (3+M):   Columns  1: 5 – PI number (I5) (note 4)
                   Columns  6:13 – Phase number (I8)

Line (4+M) to N:   Columns  1: 5 – PO number (I5)
                   Columns  6:13 – Window number (I8)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The number of PI lines will be the number of PIs. The number of PO lines will be the number of POs.
3—Under certain conditions, not all POs will be assigned WINDOWs. In these situations, POs without assigned WINDOWs will have zero (0) as their WINDOW number.
4—M is equal to the number of PIs that have assigned PHASEs (line 2, columns 11:15).

Example:

```
PHASE_CONNECTIONS        26   2UnitUnderTestAZ         10-JAN-2000 12:00
    9    6    9    4
    1       5
    2       1
    3       3
    4       5
    5       2
    6       1
    7       1
    8       4
    9       1
    1       1
    2       1
    3       1
    4       1
    5       0
    6       0
```

5.27 AUXILIARY_PIN_NAMES file

File type name: AUXILIARY_PIN_NAMES
File name: auxpins.tap
File number: 27

Description: This file provides an alternate selection of ground and power pins on the tester at the component level. Entries in this file are pointed to by the COMPONENT_TYPE file entries.

Format:

```
Line 1:         Columns  1:24 – DTIF file type name (A24)
                Columns 25:27 – DTIF file number (I3)
                Columns 28:31 – File version number (optional value)(I4)
                Columns 32:55 – UUT name (A24)
                Columns 56:72 – File creation date and time (A17) (note 1)
                Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                in file generation (A5)

Line 2:         Columns  1:10 – Number of component Types (I10)

Line 3 to N:    Columns  1: 8 – Component type number (I8)
                Columns  9:24 – Ground pin name (A16)
                Columns 25:40 – Power pin name (A16)
                Columns 41:56 – Alternate ground pin name (A16)
                Columns 57:72 – Alternate power pin name (A16)
```

NOTE
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

Example:

```
AUXILIARY_PIN_NAMES      27    2UnitUnderTestBA         10-JAN-2000 12:00
         2
       1GND_PIN         PWR_PIN         ALT_GND         ALT_PWR
       2
```

5.28 PI_FORMATS file

File type name: PI_FORMATS
File name: piformats.tap
File number: 28

Description: This file defines the PI pin format used on each stimulus pin. Each PI has a format code assigned to it. The code is defined in the FORMAT_ATTRIBUTES file.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2:          Columns  1: 4 – Number of lines per format packet (I4) (note 2)

Line 3 to N:     Columns  1:80 – Format Packet Data ((A2,I10),#PIs*(I4)) (note 3)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—This is determined by 1+[(#PIs+2)/20].
3—Each packet contains a pattern number and one format number per PI. PI data is in the same order as the PI_NAMES file. Each packet starts on a new line, and may span several lines (depending on the number of PIs). There will be one extra entry at the end of the file, with pattern number equal to (num_patterns+1) and format packet data reflecting that of the previous line. Format number is described in the FORMAT_ATTRIBUTES file.

Example:

```
PI_FORMATS              28    2UnitUnderTestBB         10-JAN-2000 12:00
   2
           1   2   0   2   2   0   2   2   1   0   0   0   0   0   0   0   0   0
   0   4   4   4   4
          16   2   0   2   2   0   2   2   1   0   0   0   0   0   0   0   0   0
   0   4   4   4   4
```

5.29 FORMAT_ATTRIBUTES file

File type name: FORMAT_ATTRIBUTES
File name: formattrs.tap
File number: 29

Description: This file defines the relationship between the stimulus PI pin format identifiers and their associated name.

Format:

```
Line 1:           Columns  1:24 - DTIF file type name (A24)
                  Columns 25:27 - DTIF file number (I3)
                  Columns 28:31 - File version number (optional value)(I4)
                  Columns 32:55 - UUT name (A24)
                  Columns 56:72 - File creation date and time (A17) (note 1)
                  Columns 73:77 - Either spaces or "ERROR" to indicate an error
                                  in file generation (A5)

Line 2:           Columns  1: 4 - Number of format data lines (I4)

Line 3 to N:      Columns  1: 4 - PI pin format number (I4) (note 2)
                  Columns  5:19 - PI pin format name (A15)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The following are five possible formats:
$NRET - Non-return (retains the same level as the last stimulus state specified)
$RZERO - Return to zero
$RONE - Return to one
$RCOMP - Return to complement
$ROFF - Return to ‘off’ - stimulus driving HIZ

Example:

```
FORMAT_ATTRIBUTES        29   2UnitUnderTestBC         10-JAN-2000 12:00
    5
    0$NRET
    1$RZERO
    2$RONE
    3$RCOMP
    4$ROFF
```

5.30 F.D. _CROSS_REFERENCE file

File type name: F.D. _CROSS_REFERENCE
File name: fdxref.tap
File number: 30

Description: This file provides a table for cross-referencing each POPAT with the fault sets it is capable of detecting. Each list of fault sets here implicitly applies to the corresponding entry in the F.D._POPATS file.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2:          Columns  1: 5 – Number of POPATs (I5)
                 Columns  6:10 – Greatest number of fault sets in POPAT list (I5)
                 Columns 11:12 – Unused (I2)
                 Columns 13:14 – Unused (I2)

Line 3 to N:     Columns  1:80 – POPAT isolation (cross-reference) data (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—POPAT isolation data contains a list of fault sets for each POPAT. Each line is organized as lists of fault set numbers, with each list preceded by the size of the list (i.e., the number of list elements). Each list begins on a new line, with the list size starting in column 1; the list of elements uses as many lines as necessary. A fault set number is not broken over two lines. If the current line does not have room for the six-character fault set value, then the value starts in column 1 of the next line. A negative value for the fault set number indicates that the corresponding POPAT possibly detects the fault set.

Example:

```
F.D._CROSS_REFERENCE     30   2UnitUnderTestBD         10-JAN-2000 12:00
    7    6
     1      1
     1      2
     1      3
     4     17     3     14    -18
     6     10     9     17      2     15     12
     4     17     2     15     12
     2      1    16
```

5.31 PROBETAG_ DEFINITIONS file

File type name: PROBETAG_ DEFINITIONS
File name: probetag.tap
File number: 31

Description: This file describes the operating parameters for a diagnostic probe, when used. Parameters specified include: timing resolution, delays, voltage threshold, and skew limits.

Format:

```
Line 1:           Columns  1:24 – DTIF file type name (A24)
                  Columns 25:27 – DTIF file number (I3)
                  Columns 28:31 – File version number (optional value)(I4)
                  Columns 32:55 – UUT name (A24)
                  Columns 56:72 – File creation date and time (A17) (note 1)
                  Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                  in file generation (A5)

Line 2:           Columns  1: 8 – Timing resolution value (I8)
                  Columns  9:16 – Units of time (I8)
                  Columns 17:24 – The minimum detectable pulse width for use with
                                  a probe in STU (I8)
                  Columns 25:32 – The minimum probe delay in STU (I8)
                  Columns 33:40 – The maximum probe delay in STU (I8)
                  Columns 41:48 – The minimum probe skew in STU (I8)
                  Columns 49:56 – The maximum probe skew in STU (I8)
                  Columns 57:65 – The probetags in this file (I9)
                  Columns 66:67 – The number of lines per probetag (I2)
                  Columns 68:71 – Maximum number of PSETs per probetag (I4)

Line 3:           Columns  1:20 – Probetag names in character strings (A20)
                                  (note 2)
                  Columns 21:28 – Logic low voltage in millivolts (I8)
                  Columns 29:36 – Logic high in millivolts (I8)
                  Columns 37:40 – Probetag mapping class (I4) (note 3)
                  Columns 41:48 – Timing data for WINDOW open in STU for PSET1 (I8)
                  Columns 49:56 – Timing data for WINDOW close in STU for PSET1 (I8)

Line 4 to 6:      Columns  1:80 – 5-paired WINDOW open and close times in STU for
                                  PSETs 2 through 16 (5*(I8,I8)) (note 4)

Line 7 to (4*N):  Columns  1:80 – PROBETAG_DEFINITIONS data (note 5)
```

NOTES

1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

2—Actual probetag data begins with line 3. Each probetag data set requires a total of four lines.

3—This field contains one of the following values: –1, 0, 1, 2, 3, 4. These values are defined as follows:

- –1 Unspecified
- 0 Settled
- 1 Settled to X
- 2 Pulse
- 3 Highlow
- 4 Highlow–X

4—There is a maximum of 16 possible PSETs. The first pair of numbers appearing in line 4 for a set apply to PSET 2, since the open and close times for PSET 1 appear in line 3 after the mapping class. The first number in the pair is the WINDOW open time and the second number is the WINDOW close time.

5—Additional four-line sets repeat the format of lines 3–6, one set per probetag in a set.

Example:

```
PROBETAG_DEFINITIONS      31    2UnitUnderTestBE         10-JAN-2000 12:00
       1      -12    10000        0        0    -5000     5000         3 4   16
SDF_P                   1000     2000    2        0   970000
       0   970000        0   970000        0   970000        0   970000        0   970000
       0   970000        0   970000        0   970000        0   970000        0   970000
       0   970000        0   970000        0   970000        0   970000        0   970000
SDF_S                   1000     2000    4   972000   995000
  972000   995000   972000   995000   972000   995000   972000   995000   972000   995000
  972000   995000   972000   995000   972000   995000   972000   995000   972000   995000
  972000   995000   972000   995000   972000   995000   972000   995000   972000   995000
DEFAULT
  972000   995000   972000   995000   972000   995000   972000   995000   972000   995000
  972000   995000   972000   995000   972000   995000   972000   995000   972000   995000
  972000   995000   972000   995000   972000   995000   972000   995000   972000   995000
```

5.32 PROBETAG_ASSIGNMENTS file

File type name: PROBETAG_ASSIGNMENTS
File name: probeasgn.tap
File number: 32

Description: This file assigns a probe, defined in PROBETAG_DEFINITIONS, to each probeable node in the circuit.

Format:

```
Line 1:           Columns  1:24 - DTIF file type name (A24)
                  Columns 25:27 - DTIF file number (I3)
                  Columns 28:31 - File version number (optional value)(I4)
                  Columns 32:55 - UUT name (A24)
                  Columns 56:72 - File creation date and time (A17) (note 1)
                  Columns 73:77 - Either spaces or "ERROR" to indicate an error
                                  in file generation (A5)

Line 2:           Not used

Line 3 to N:      Columns  1:80 - Index into the PROBETAG_DEFINITIONS (20*(I4))
                                  (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—This field provides a probe assignment for each user node using user node order. Each assignment has four characters that specify an index into the PROBETAG_DEFINITIONS file. If the index is zero, no probetag is assigned to the node (net).

Example:

```
PROBETAG_ASSIGNMENTS      32    2UnitUnderTestBF         10-JAN-2000 12:00

   1   2   1   1   2   1   1   3   2   2   2   2   2   2   2   2   2   2   2   2
   2   2   0   0   0   0   0   0   0   0   0   1   1   1
```

5.33 BURSTS file

File type name: BURSTS
File name: bursts.tap
File number: 33

Description: This file defines the burst register storage depth requirements behind each I/O pin, and the starting pattern number for each successive burst.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2:          Columns  1: 5 – Number of bursts (I5)
                 Columns  6:15 – Number of patterns in the simulation (I10)
                 Columns 16:20 – Number of first burst (usually '1') (I5)

Line 3 to N:     Columns  1:10 – Pattern number of each new burst (I10) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—An extra burst entry signifies the end-of-file. The entry is (number of patterns + 1). This is not included in Line 2/ Columns 1:5, the number of bursts entry.

Example:

```
BURSTS                   33   2UnitUnderTestBG         10-JAN-2000 12:00
    1         15    1
         1
        16
```

5.34 STIMULUS_TEXT file

File type name: STIMULUS_TEXT
File name: stimtext.tap
File number: 34

Description: This file provides a way for the simulation developer to include messages from the stimulus set into the TPS file.

Format:

```
Line 1:           Columns  1:24 – DTIF file type name (A24)
                  Columns 25:27 – DTIF file number (I3)
                  Columns 28:31 – File version number (optional value)(I4)
                  Columns 32:55 – UUT name (A24)
                  Columns 56:72 – File creation date and time (A17) (note 1)
                  Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                  in file generation (A5)

Line 2:           Columns  1:10 – Number of patterns in simulation (I10)

Line 3 to N:      Column      1 – Op code of stimulus text information (note 2)
                  Columns  2:80 – Stimulus text information (note 3)

'P' op code       Columns  2:11 – Pattern number of new pattern (note 4)

'M' op code       Columns  2: 5 – Length of message text
                  Columns  6: n – Message text, using as many successive lines
                  as necessary

'L' op code       Columns  2: 5 – Length of label name
                  Columns  6: n – Label name, using as many successive lines as
                  necessary

'T' op code       Columns  2: 5 – Length of test text
                  Columns  6: n – Test text, using as many successive lines as
                                  necessary
```

NOTES

1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].

2—The stimulus text information is one of three types: MESSAGE, LABEL, or TEST TEXT. MESSAGE refers to documentation, specifically comments in the stimulus. LABEL simply means that the stimulus was assigned a label at this point. TEST TEXT means that there is some text that should be inserted directly, unedited, into the tester job plan. The file is a series of op codes with accompanying information. Each op code starts on a new line, regardless of where the previous op code information leaves off on the previous line. There are four possible codes:

op code	Meaning
P	Start of Pattern (note 4)
M	Message Text
L	Label Name
T	Test Text

3—The format for the stimulus text information is different for each op code.

4—There will not be a 'P' op code for every pattern, but only for those patterns that contain at least one line of text.

Example:

```
STIMULUS_TEXT             34   2UnitUnderTestBH         10-JAN-2000 12:00
        15
P        4
M   20Writing data to rams
P        10
M   22Reading data from rams
```

5.35 NODE_NAMES file

File type name: NODE_NAMES
File name: nodenames.tap
File number: 35

Description: This file identifies all of the user defined node names, other than PI_NAMES and PO_NAMES.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)

Line 2:          Columns  1:10 – Highest user node number (I10) (note 2)
                 Columns 11:20 – Number of user nodes with user-given node names
                                 (I10)

Line 3 to N:     Columns  1:10 – User node number (I10) (note 3)
                 Column      11 – Unused
                 Columns 12:43 – Node name character string (A32)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—This file will only contain the node names that were explicitly assigned by the user.
3—This information will be in ascending order of user node number.

Example:

```
NODE_NAMES              35   2UnitUnderTestBI         10-JAN-2000 12:00
       117         10
       100 DBUS4
       101 DBUS3
       102 DBUS2
       103 DBUS1
       104 IDBUS1
       105 IDBUS2
       106 IDBUS3
       107 IDBUS4
       108 $L1B
       109 $L0B
```

5.36 EVENTS_INIT file

File type name: EVENTS_INIT
File name: eventsinit.tap
File number: 36

Description: This file contains circuit initialization information using the same format as the EVENT file. The circuit initialization event is produced by simulating a pattern that sets all the primary inputs to an unknown state, and sets all pseudo PI nodes to their respective value. This file also includes the states for nodes that are determined at power-up time.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                 in file generation (A5)

Line 2:          Columns  1:10 – Timing resolution value (I10) (note 2)
                 Columns 11:14 – Unit of time (I4)

Line 3 to N:     Columns 1:80 –  Contiguous stream of events data (note 3)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The timing resolution value is a positive integer that, in conjunction with the units of time, represents the STU. If the resolution is 5 and the units of time is –9, the STU is 5 nanoseconds.
3—An event is composed of a two-part data grouping consisting of an integer followed by an alphanumeric op code, which determines the meaning of the integer value. The integer value represents either time in STU, node number, or a pattern number depending on the op code it precedes. Time values are interpreted as either delta time from the last event or absolute time from the beginning of the pattern depending upon the op code. If the integer value is missing, event time is assumed to be the same as the most recent time value within the pattern. The single-character op codes are as follows:

op code	Meaning
@	Absolute time relative to start of a pattern
A	A relative (delta) time field; specifying the minimum time at which the nodal event might occur. The specified nodes are in a stable state prior to this point in time.
B	A relative (delta) time field; specifying the minimum time at which the nodal event might occur. All of the specified nodes are in an unstable state prior to this point in time.
C	A relative (delta) time field; specifying the maximum time at which the nodal event might occur. These specified nodes are in a stable state after this point in time.
D	A relative (delta) time field; specifying the maximum time at which the nodal event might occur. All of these specified nodes are in an unstable state after this point in time.
E	User node number that just went ‘low’ or ‘0’
F	Unused
G	Unused
H	User node number that just went ‘high’ or ‘1’
I	Unused
J	Unused
K	User node number that just went to ‘Z’ or high impedance
L	Unused
M	Unused
N	User node number that just went to ‘X’ or unknown
O	Unused
P	End of pattern
Q	End of Burst
R	End-of-file (This should never be preceded by an integer.)

Example:

```
EVENTS_INIT                36   2UnitUnderTestBJ         10-JAN-2000 12:00
                      1 -12
0A1K2K3K4K5K6K7K8K9K10K11K12K13K14K15K16K17K18K19K20K21K22K117H108H116E109E111K1
12H113E115EC1K2K3K4K5K6K7K8K9K10K11K12K13K14K15K16K17K18K19K20K21K22K117H108H116
E109E111K112H113E115E1A96H97H98H99HC96H97H98H99H4266A45H26H2400A52E53E54E55E56E5
8E59E60E2242A49H50H6091C45H46H52E53E54E55E56E58E59E60E7000C49H50H978000A115HC115
H1000001@0P
```

5.37 EQUIV_FAULTS file

File type name: EQUIV_FAULTS
File name: equivflts.tap
File number: 37

Description: This file contains the fault titles organized in groups of equivalent faults with each group assigned a unique number. Equivalent faults are faults that will produce identical responses that are indistinguishable at the primary output, regardless of the stimulus.

Format:

```
Line 1:           Columns  1:24 – DTIF file type name (A24)
                  Columns 25:27 – DTIF file number (I3)
                  Columns 28:31 – File version number (optional value)(I4)
                  Columns 32:55 – UUT name (A24)
                  Columns 56:72 – File creation date and time (A17) (note 1)
                  Columns 73:77 – Either spaces or "ERROR" to indicate an error
                                  in file generation (A5)

Line 2:           Columns  1: 4 – Number 'M' of fault title types (I4)
                  Columns  5:12 – Number of equivalent fault groups (I8)
                  Columns 13:20 – Total number of faults in the file (I8)
                  Columns 21:24 – Length for largest fault title type (I4)

Line 3 to (M+2):  Columns  1: 4 – Fault title type (one type per line) (I4)
                  Columns  5:80 – Fault title type description (A76)

Line (M+3) to N:  Columns  1: 8 – Equivalent fault group number (I8)
                  Columns  9:16 – Number of faults in equivalent fault group (I4)
                  Columns 17:20 – Fault title type (I4)
                  Columns 21:24 – Length of the following fault title (I4)
                  Columns 25-80 – Fault title (A56) (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—If the length of the fault title exceeds 56 characters, a new record will be generated, up to 80 characters.

Example:

```
EQUIV_FAULTS              37   2UnitUnderTestBK         10-JAN-2000 12:00
   9       12       32  12
   1Stuck-at-0
   2Stuck-at-1
   3Stuck-at-Z
   4Stuck-at-X
   5Open-to-0
   6Open-to-1
   7Open-to-Z
   8Open-to-X
   9Shorted nodes
       1       1   7   8<U8>10/Z
       2       1   2   7<U8>5@1
       3       1   1   8<U7>16@0
       4       2   5   7<U3>4/0
       5       5   2   12<^>AD_LOAD@1
       6       1   6   12<^>AD_LOAD/1
       7       3   5   7<U1>3/0
       8       3   7   7<U2>5/Z
       9       2   5   7<U9>9/0
      10       7   2   8<U9>11@1
      11       4   1   11<^>DATA3.@0
      12       2   5   11<^>DATA3./0
```

5.38 PROBE_DETECTION file

File type name: PROBE_DETECTION
File name: probedet.tap
File number: 38

Description: This file contains a list of equivalent fault groups that will be detected by a probe placed on the indicated user node at the end of the indicated pattern.

Format:

```
Line 1:          Columns  1:24 – DTIF file type name (A24)
                 Columns 25:27 – DTIF file number (I3)
                 Columns 28:31 – File version number (optional value)(I4)
                 Columns 32:55 – UUT name (A24)
                 Columns 56:72 – File creation date and time (A17) (note 1)
                 Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                 file generation (A5)

Line 2:          Columns  1: 8 – Number of lists of equivalent fault group numbers
                                 (I8) (note 2)
                 Columns  9:16 – Size of the longest list (I8)
                 Columns 17:24 – Highest equivalent fault group number in any list
                                 (I8)

Line 3 to N:     Columns  1: 8 – Pattern number for this probe measurement (I8)
                 Columns  9:16 – User node number on which the probe would be placed
                                 (I8) (note 3)
                 Columns 17:24 – Number of equivalent fault groups (I8)
                 Columns 25:80 – Each equivalent fault group (I8) (note 4)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The actual fault titles may be determined by using the EQUIV_FAULTS file.
3—A positive number indicates that the expected fault-free state of the node is high. A negative number indicates that the expected fault-free state of the node is low.
4—Additional records are utilized when needed. The equivalent fault group numbers listed are those that are being detected now that have not been detected at this user node on an earlier pattern. Further, if the faults of an equivalent fault group are 'possibly' detected, the set number is negative in the list.

Example:

```
PROBE_DETECTION          38   2UnitUnderTestBL         10-JAN-2000 12:00
      14       9      191
       1       1        1      65
       1      -2        1      67
       1       3        2      71     109
       1       4        3      71     109     111
       1       5        4      71     109     111     113
       1       6        5      71     109     111     113     115
       1       7        6      71     109     111     113     115     117
       1       9        7      71     109     111     113     115     117     119
       1      12        8      71     109     111     113     115     117     119
     121
       1      13        9      71     109     111     113     115     117     119
     121     123
       2     -25        3     -70    -186     189
       2     -21        3     -70    -186     191
       3     -30        1      77
       3     -26        3      77    -186      18
```

5.39 F.D._EQUIV_SETS file

File type name: F.D._EQUIV_SETS
File name: fdeqvs.tap
File number: 39

Description: This file uses the equivalent fault groups from EQUIV_FAULTS to define the faults contained in each fault set.

Format:

```
Line 1:           Columns  1:24 – DTIF file type name (A24)
                  Columns 25:27 – DTIF file number (I3)
                  Columns 28:31 – File version number (optional value)(I4)
                  Columns 32:55 – UUT name (A24)
                  Columns 56:72 – File creation date and time (A17) (note 1)
                  Columns 73:77 – Either spaces or "ERROR" to indicate an error in
                                  file generation (A5)

Line 2:           Columns  1: 8 – The number of fault sets (I8)
                  Columns  9:16 – Total number of equivalence fault group numbers (I8)
                  Columns 17:24 – Highest equivalence fault group number (I8)
                  Columns 25:32 – Largest number of equivalence fault groups in any
                                  fault set (I8)

Line 3 to N:      Columns  1: 8 – Fault set number (I8)
                  Columns  9:16 – Size of the list (I8)
                  Columns 17:80 – List of equivalence fault group numbers (I8)
                                  (note 2)
```

NOTES
1—Date and Time are represented as: dd-mmm-yyyy hh:mm [dd – day, mmm – month, yyyy – year, hh – hour (24 hour clock), mm – minutes].
2—The information for each fault set begins with a new line and uses as many lines as required.

Example:

```
F.D._EQUIV_SETS         39   2UnitUnderTestBM         10-JAN-2000 12:00
      11      40     227      12
       1       2     101     220
       2       2     103     222
       3       2     105     224
       4       2     107     226
       5       2     104     223
       6       2     108     227
       7       2     102     221
       8       2     106     225
       9       6      45      50      99     100     190     191
      10      12      38      46      48      49      93      94      97      98
     192     193     196     197
      11       6      40      47      95      96     194     195
```

6. Conformance

This clause describes the capabilities required for implementing the standard. Conformance to the standard is divided into three areas that are mutually independent. An application shall satisfy the required functions in the following areas, as applicable, in order to comply with this standard:

a) End-to-end test;
b) Diagnostics using Fault Dictionary;
c) Diagnostics using Probe.

The required set of files listed for each area are indicated by file type name in uppercase letters, and shall use the file name in parenthesis.

6.1 End-to-end test

The intent of an end-to-end test of a UUT is to verify the overall functionality of the item, and to indicate a failure if one exists and can be detected by the test pattern set developed for it. There is no explicit attempt to diagnose the cause of the failure.

The required set of DTIF files that would enable the generation of an end-to-end test of a digital UUT using static patterns are

HEADER (header.tap)
STIMULUS (stimulus.tap)
PO_RESPONSE (response.tap)
PI_NAMES (pinames.tap)
PO_NAMES (ponames.tap)
TIMING_PER_PATTERN (timperpat.tap)
BURSTS (bursts.tap)
STIMULUS_TEXT (stimtext.tap)

The required set of DTIF files that would enable the generation of an end-to-end test of a digital UUT using dynamic patterns are

HEADER (header.tap)
STIMULUS (stimulus.tap)
PO_RESPONSE (response.tap)
PI_NAMES (pinames.tap)
PO_NAMES (ponames.tap)
TIMING_SETS (timesets.tap)
TIMING_PER_PATTERN (timperpat.tap)
PHASE_CONNECTIONS (phaseconn.tap)
PI_FORMATS (piformats.tap)
FORMAT_ATTRIBUTES (formattrs.tap)
BURSTS (bursts.tap)
STIMULUS_TEXT (stimtext.tap)

6.2 Diagnostic test using fault dictionary

Fault isolation by fault dictionary is a diagnostic test technique that does not require physical probing of a circuit. Data generated by a fault simulator is analyzed and generally arranged into matrices that match fault indications with a characteristic failing response at the primary output pins of the UUT. Actual failing responses are captured on the ATE and compared to the fault dictionary matrix for fault isolation.

The required set of DTIF files that would enable the generation of fault dictionary diagnostics for a UUT using static patterns are

HEADER (header.tap)
STIMULUS (stimulus.tap)
PO_RESPONSE (response.tap)
PI_NAMES (pinames.tap)
PO_NAMES (ponames.tap)
MAIN_MODEL (mainmodel.tap)
COMPONENT_TYPE (types.tap)
USER_NODE (usernodes.tap)
INPUT_PIN_NAMES (inputpins.tap)
OUTPUT_PIN_NAMES (outputpin.tap)
F.D._POPATS (fdpopats.tap)
F.D._FAULT_SIGNATURES (fdfltsig.tap)
F.D._PRINT_STRINGS (fdprint.tap)
PSEUDOPI_NAMES (psupinams.tap)
TIMING_PER_PATTERN (timperpat.tap)
AUXILIARY_PIN_NAMES (auxpins.tap)
F.D._CROSS_REFERENCE (fdxref.tap)
BURSTS (bursts.tap)
STIMULUS_TEXT (stimtext.tap)
NODE_NAMES (nodenames.tap)
EQUIV_FAULTS (equivflts.tap)
F.D._EQUIV_SETS (fdeqvs.tap)

The required set of DTIF files that would enable the generation of fault dictionary diagnostics for a UUT using dynamic patterns are

HEADER (header.tap)
STIMULUS (stimulus.tap)
PO_RESPONSE (response.tap)
PI_NAMES (pinames.tap)
PO_NAMES (ponames.tap)
MAIN_MODEL (mainmodel.tap)
COMPONENT_TYPE (types.tap)
USER_NODE (usernodes.tap)
INPUT_PIN_NAMES (inputpins.tap)
OUTPUT_PIN_NAMES (outputpin.tap)
F.D._POPATS (fdpopats.tap)
F.D._FAULT_SIGNATURES (fdfltsig.tap)
F.D._PRINT_STRINGS (fdprint.tap)
PSEUDOPI_NAMES (psupinams.tap)
TIMING_SETS (timesets.tap)
TIMING_PER_PATTERN (timperpat.tap)
PHASE_CONNECTIONS (phaseconn.tap)
AUXILIARY_PIN_NAMES (auxpins.tap)
PI_FORMATS (piformats.tap)
FORMAT_ATTRIBUTES (formattrs.tap)
F.D._CROSS_REFERENCE (fdxref.tap)
BURSTS (bursts.tap)

STIMULUS_TEXT (stimtext.tap)
NODE_NAMES (nodenames.tap)
EQUIV_FAULTS (equivflts.tap)
F.D._EQUIV_SETS (fdeqvs.tap)

6.3 Diagnostic test using probe

Isolating faults using a probe is a diagnostic test technique that utilizes a probe connected to the ATE to aid in isolating failures on the UUT. The information required to generate a probe diagnostic test includes a detailed description of the circuit that would help direct the test operator where to probe, and data to determine whether or not a response monitored by the probe matched the good circuit response.

The required set of DTIF files that would enable the generation of probe diagnostics for a digital UUT using static patterns are

HEADER (header.tap)
STIMULUS (stimulus.tap)
PO_RESPONSE (response.tap)
PI_NAMES (pinames.tap)
PO_NAMES (ponames.tap)
MAIN_MODEL (mainmodel.tap)
COMPONENT_TYPE (types.tap)
USER_NODE (usernodes.tap)
INPUT_PIN_NAMES (inputpins.tap)
OUTPUT_PIN_NAMES (outputpin.tap)
NEAR_FROMS_POINTERS (nerfrmptr.tap)
NEAR_FROMS (nearfroms.tap)
EVENT (events.tap)
SETTLED_STATE_ONLY (setldonly.tap)
NODE_SOURCE (nodsource.tap)
STEPS (steps.tap)
TRISTATE_FROMS_POINTERS (zfromptrs.tap)
TRISTATE_FROMS (zfroms.tap)
PSEUDOPI_NAMES (psupinams.tap)
TIMING_PER_PATTERN (timperpat.tap)
AUXILIARY_PIN_NAMES (auxpins.tap)
PROBETAG_DEFINITIONS (probetag.tap)
PROBETAG_ASSIGNMENTS (probeasgn.tap)
BURSTS (bursts.tap)
STIMULUS_TEXT (stimtext.tap)
NODE_NAMES (nodenames.tap)
EVENTS_INIT (eventsinit.tap)
PROBE_DETECTION (probedet.tap)

The required set of DTIF files that would enable the generation of probe diagnostics for a digital UUT using dynamic patterns are

HEADER (header.tap)
STIMULUS (stimulus.tap)
PO_RESPONSE (response.tap)
PI_NAMES (pinames.tap)
PO_NAMES (ponames.tap)
MAIN_MODEL (mainmodel.tap)

COMPONENT_TYPE (types.tap)
USER_NODE (usernodes.tap)
INPUT_PIN_NAMES (inputpins.tap)
OUTPUT_PIN_NAMES (outputpin.tap)
NEAR_FROMS_POINTERS (nerfrmptr.tap)
NEAR_FROMS (nearfroms.tap)
EVENT (events.tap)
SETTLED_STATE_&_PULSES (setldpuls.tap)
NODE_SOURCE (nodsource.tap)
STEPS (steps.tap)
TRISTATE_FROMS_POINTERS (zfromptrs.tap)
TRISTATE_FROMS (zfroms.tap)
PSEUDOPI_NAMES (psupinams.tap)
TIMING_SETS (timesets.tap)
TIMING_PER_PATTERN (timperpat.tap)
PHASE_CONNECTIONS (phaseconn.tap)
AUXILIARY_PIN_NAMES (auxpins.tap)
PI_FORMATS (piformats.tap)
FORMAT_ATTRIBUTES (formattrs.tap)
PROBETAG_DEFINITIONS (probetag.tap)
PROBETAG_ASSIGNMENTS (probeasgn.tap)
BURSTS (bursts.tap)
STIMULUS_TEXT (stimtext.tap)
NODE_NAMES (nodenames.tap)
EVENTS_INIT (eventsinit.tap)
PROBE_DETECTION (probedet.tap)

Table 1 summarizes the required set of DTIF files for the implementation of various digital test strategies.

Table 1—Conformance matrix

DTIF file name	File no.	End-to-end	End-to-end	Fault Dictionary	Fault Dictionary	Probe	Probe
		Static	Dynamic	Static	Dynamic	Static	Dynamic
header.tap	1	X	X	X	X	X	X
stimulus.tap	2	X	X	X	X	X	X
response.tap	3	X	X	X	X	X	X
pinames.tap	4	X	X	X	X	X	X
ponames.tap	5	X	X	X	X	X	X
mainmodel.tap	6			X	X	X	X
types.tap	7			X	X	X	X
usernodes.tap	8			X	X	X	X
inputpins.tap	9			X	X	X	X
outputpin.tap	10			X	X	X	X
nerfrmptr.tap	11					X	X
nearfroms.tap	12					X	X
events.tap	13					X	X
setldonly.tap	14					X	
setldpuls.tap	15						X
nodsource.tap	16					X	X
steps.tap	17					X	X
fdpopats.tap	18			X			
fdfltsig.tap	19			X			
fdprint.tap	20			X			
zfromptrs.tap	21					X	X
zfroms.tap	22					X	X
psupinams.tap	23			X		X	X
timesets.tap	24		X				X
timperpat.tap	25	X	X	X		X	X
phaseconn.tap	26		X		X		X
auxpins.tap	27			X	X	X	X
piformats.tap	28		X		X		X
formattrs.tap	29		X		X		X
fdxref.tap	30			X	X		
probetag.tap	31					X	X
probeasgn.tap	32					X	X
bursts.tap	33	X	X	X	X	X	X
stimtext.tap	34	X	X	X	X	X	X
nodenames.tap	35			X	X	X	X
eventsinit.tap	36					X	X
equivflts.tap	37			X	X		
probedet.tap	38					X	X
fdeqvs.tap	39			X	X		

Annex A

(informative)

Implementation overview

Test data generated must be linked to a target tester in order to perform a test on a given UUT. This annex describes various software elements, which are utilized in an application development environment to meet the target tester input data requirements. There are two environments, the application development environment and the ATE environment, which are shown in Figure A.1.

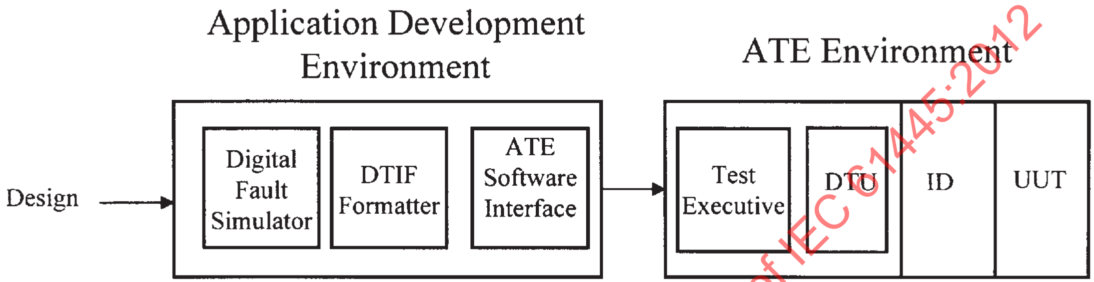

Figure A.1—Software elements of a test environment

A.1 Application development environment

A.1.1 Digital fault simulator

A system that predicts the behavior of a digital circuit under good and faulty conditions via a set of software programs. It utilizes component software models to create a circuit connectivity model. A set of input stimuli and a corresponding set of responses are generated for good circuit and faulty circuit performance. Fault detection and isolation data are established using the simulator's internal format.

A.1.2 DTIF formatter

The DTIF formatter is the software that converts non-standard test data, usually from the DATPG, into the DTIF format.

A.1.3 ATE Software interface

Software required to link the DTIF data to the target tester. It may include the following:

a) A tester peculiar language which describes test strategy for a given UUT and to which any other required data can be linked.
b) A way to create a tester environment (e.g., UUT/ID I/O pin mapping, UUT/ATE timing data considerations, etc.).
c) A processor to convert the DTIF data into source data for the target tester.

A.2 ATE environment

A.2.1 Test executive

The test executive is the set of core software routines that access the ATE hardware.

A.2.2 Digital test unit (DTU)

The DTU is the specific hardware for applying and measuring digital test patterns.

A.2.3 Interface device (ID)

The ID is the hardware necessary to interlace the ATE interface to the UUT interface.

A.2.4 Unit under test (UUT)

The UUT is the device being tested.

Annex B

(informative)

DTIF dependency diagrams

The purpose of the dependency diagrams is to show the interrelationship and dependency of the files. There are four dependency diagrams (Figures B.2 through B.5) representing the following groups of files:

a) Model;
b) Stimulus and Response;
c) Fault dictionary;
d) Probe.

In all diagrams, the arrows represent pointers or dependency of files. A file that "points to" another file, "depends on" the information contained in that file. The dashed lines and dashed boxes represent a dependency to a file in another functional group. The only connection of lines is when the same file points to a series of other files. Therefore, there are no lines from two or more different files interconnecting to point to one file (see Figure B.1).

Figure B.1 — File connection description

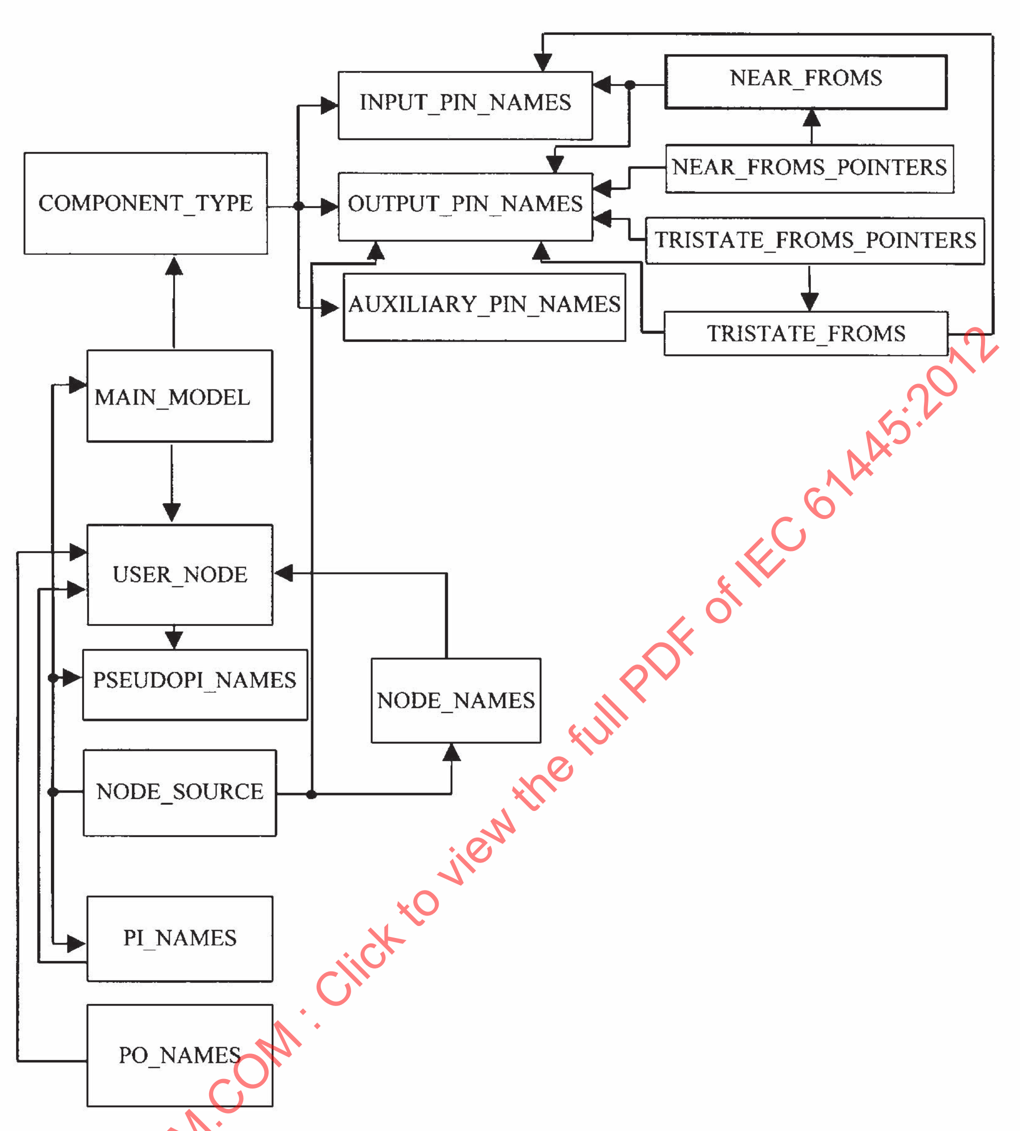

Figure B.2—UUT Model Group dependency diagram

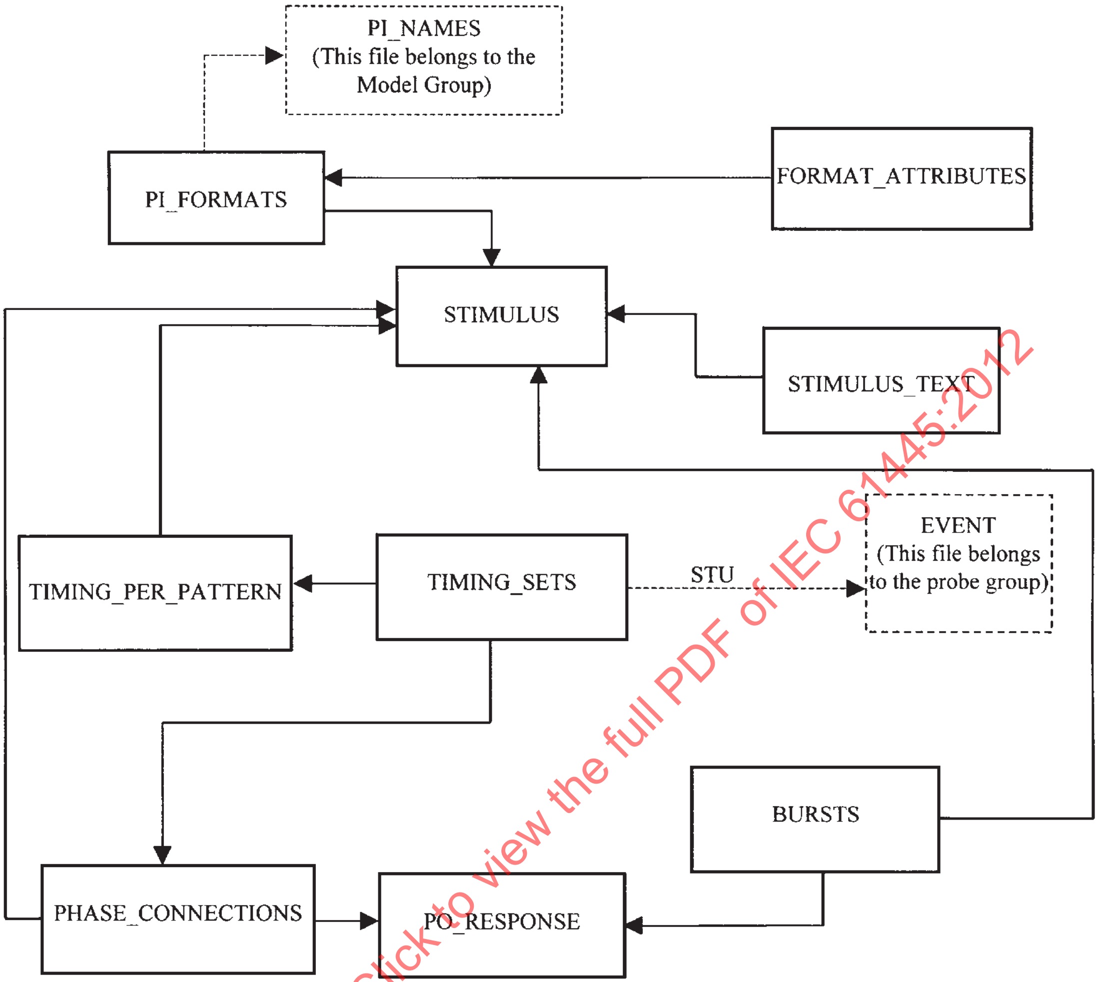

Figure B.3—Stimulus and Response Group dependency diagram

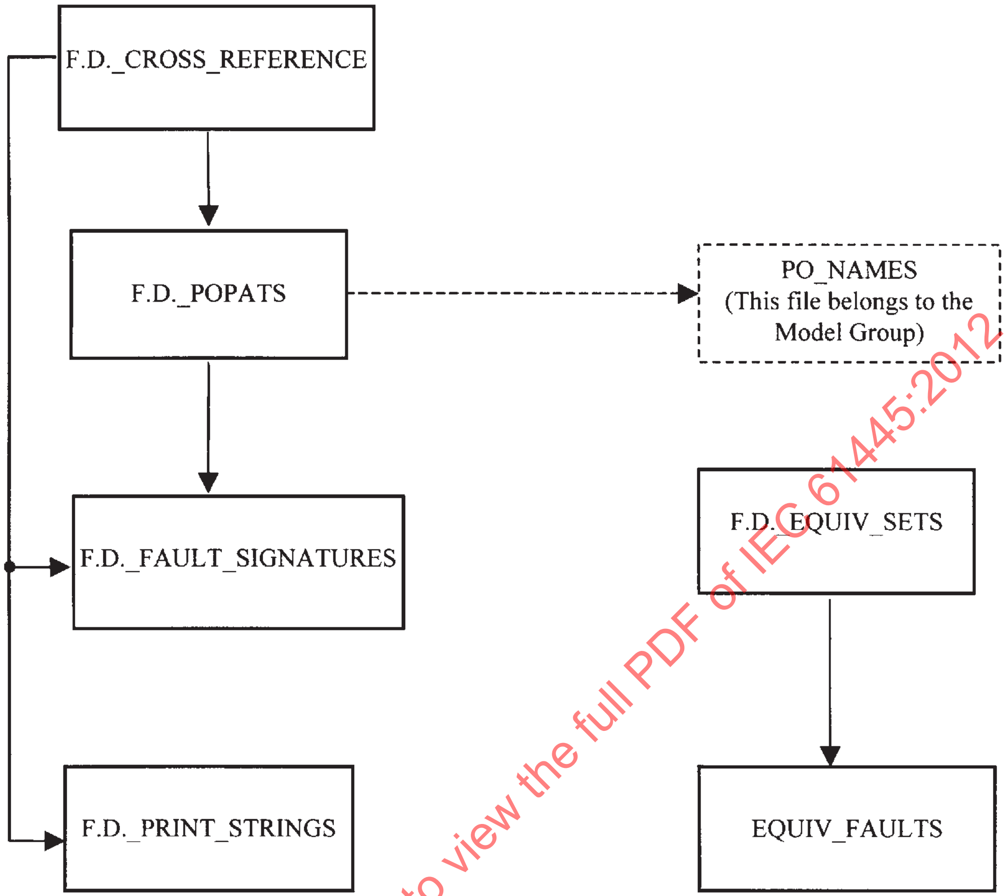

Figure B.4—Fault Dictionary Group dependency diagram

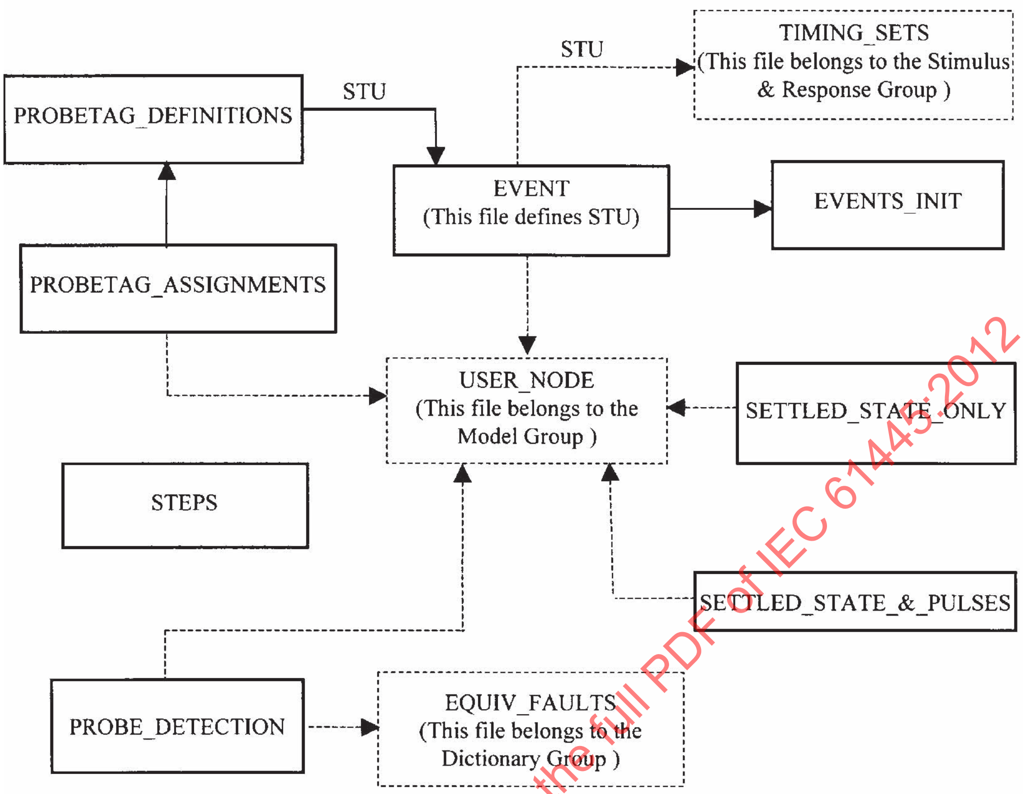

Figure B.5—Probe Group dependency diagram

Annex C

(informative)

Example circuit

Figure C.1 is the schematic of the circuit used in a test simulation. A complete set of input source files and the resultant DTIF compliant output file listings are provided for this circuit. The DTIF compliant output files may be generated by a variety of DATPGs. This example employed LASAR as the DATPG in producing these files. The LASAR simulation input files necessary to produce all 39 of the DTIF files are listed in C.1. The DTIF files are listed in C.2.

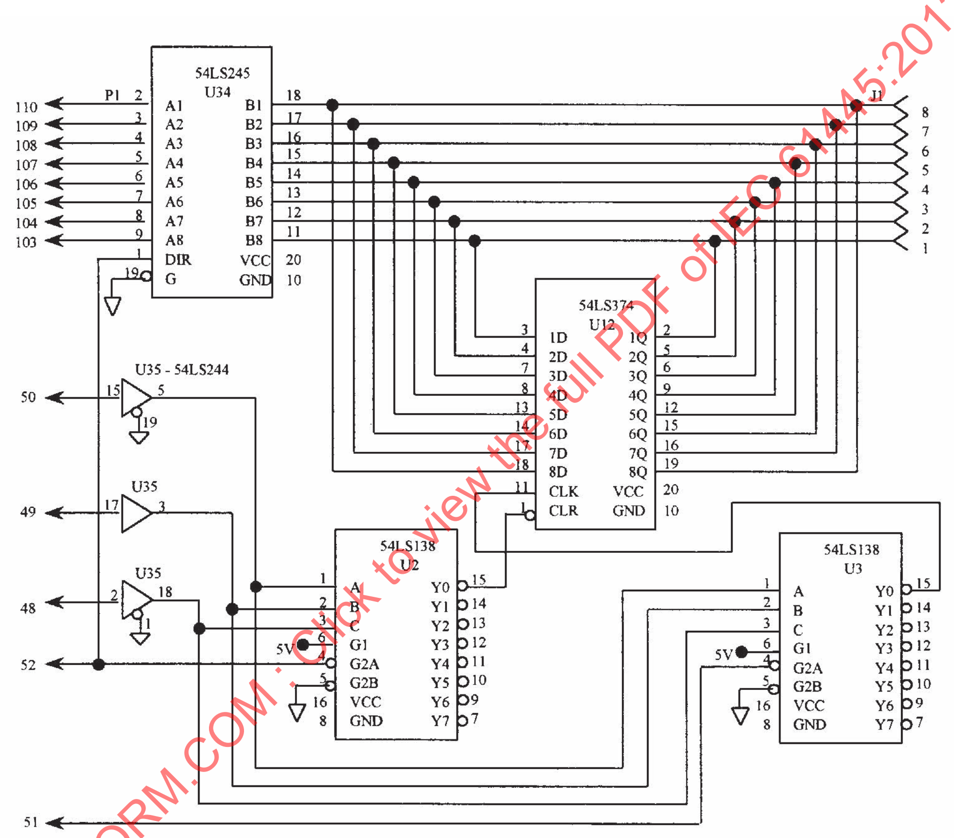

Figure C.1—Example circuit used in a test simulation

C.1 Input source files

C.1.1 Batch.com

This file runs all the LASAR commands in a batch mode.

```
$ LASAR
  COMPILE MODEL EXAMPLE.NET
  COMPILE FAULTLIST FAULTPIN.FLT
  COMPILE STIMULUS STIM.PAT
  SIMULATE
  JUDGE/NODROP_POSSIBLE_DETECTS/PROBE_DATA/FAULT_DICTIONARY
```

```
   GENERATE FAULT_DICTIONARY
   COMPILE PROBESPECS
   POSTPROCESS LSRTAP/DIAGNOSE=TR,FA/PROBE_FILES/EQUIVALENT_FAULTS/WIZARD
$  EXIT
```

C.1.2 Example.net

This file is the netlist for the example circuit.

```
MODULE NAME = 'EXAMPLE';
!
DECLARATION SECTION;
!
DECLARE AS $LOG_0 NODE 'GND';
DECLARE AS $LOG_1 NODE 'VCC';
!
END_DECLARATIONS;
!
COMPONENT SECTION;
!
U2 = SN74LS138;
U3 = SN74LS138;
U12 =SN54LS374;
U34 = SN54LS245;
U35 = SN54LS244;
!
END_COMPONENT;
!
NODE SECTION;
!
@GND, U34-19, U35-1, U35-19, U2-5, U3-5;
!
@VCC, U2-6, U3-6;
! PRIMARY INPUTS
PI P1_50, U35-15;
PI P1_49, U35-17;
PI P1_48, U35-2;
PI P1_52, U2-4, U34-1;
PI P1_51, U3-4;
! BIDIRECTIONALS
BI J1_8, U34-18, U12-19, U12-18;
BI J1_7, U34-17, U12-16, U12-17;
BI J1_6, U34-16, U12-15, U12-14;
BI J1_5, U34-15, U12-12, U12-13;
BI J1_4, U34-14, U12-9, U12-8;
BI J1_3, U34-13, U12-6, U12-7;
BI J1_2, U34-12, U12-5, U12-4;
BI J1_1, U34-11, U12-2, U12-3;
BI P1_110, U34-2;
BI P1_109, U34-3;
BI P1_108, U34-4;
BI P1_107, U34-5;
BI P1_106, U34-6;
BI P1_105, U34-7;
BI P1_104, U34-8;
BI P1_103, U34-9;
! INTERNAL SIGNALS
U35-5, U2-1, U3-1;
U35-3, U2-2, U3-2;
U35-18, U2-3, U3-3;
U2-15, U12-1;
U3-15, U12-11;
!
```

```
END_NODE;
!
END_MODULE;
```

C.1.3 Faultpin.flt

This file lists all the fault types to be observed.

```
SWITCHES;
         EQUIVALENCING=INTERNAL;
         PREDETECTION=ON;
         SAMPLING=OFF;
         REPORT=NONE;
ENDSWITCHES;
INSERT NONPRIMITIVE OUTPUTS STUCK, INPUTS OPEN;
INSERT PIs STUCK;
INSERT POs STUCK;
```

C.1.4 Stim.pat

This file is the static stimulus for the example circuit.

```
INCLUDE 'TIMING.PAT'
!
DECLARATIONS
      GROUPNAME DBUS FOR P1_110 P1_109 P1_108 P1_107 P1_106 P1_105 P1_104 P1_103
      GROUPNAME JBUS FOR J1_8 J1_7 J1_6 J1_5 J1_4 J1_3 J1_2 J1_1
ENDDECLARATIONS
!
.NOBFR
H INPUTS L P1_51 P1_50 P1_49 P1_48 OFF JBUS !INITIALIZE THE UUT
MSG '           Verify DBUS through J1_8 - J1_1'
L P1_110 P1_108 P1_106 P1_104    !VERIFY DBUS THROUGH JBUS - NO SHORTS
H P1_110 P1_108 P1_106 P1_104    !VERIFY DBUS THROUGH JBUS - NO SHORTS
L P1_109 P1_107 P1_105 P1_103    !VERIFY DBUS THROUGH JBUS - NO SHORTS
H P1_109 P1_107 P1_105 P1_103    !VERIFY DBUS THROUGH JBUS - NO SHORTS
OFF DBUS !GET U34 READY TO READ
H JBUS !PLACE A HI ON THE JBUS TO CLOCK INTO U12
L P1_52 !CHANGE DIRECTION ON U34 (B->A) AND ENABLE THE OUTPUT ON U12
MSG '           Clock U12'
H P1_51 !CLOCK U12
L P1_51 !CLOCK U12
OFF JBUS !READ DATA FROM U12 OUT THROUGH DBUS
H P1_52 !CHANGE DIRECTION ON U34 (A->B) AND DISABLE THE OUTPUT ON U12
L JBUS  !PLACE A LO ON THE JBUS TO CLOCK INTO U12
L P1_52 !CHANGE DIRECTION ON U34 (B->A) AND ENABLE THE OUTPUT ON U12
MSG '           Verify data through the DBUS'
MSG '           Clock U12'
H P1_51 !CLOCK U12
L P1_51 !CLOCK U12
OFF JBUS !READ DATA FROM U12 OUT THROUGH DBUS
H P1_52 !CHANGE DIRECTION ON U34 (A->B) AND DISABLE THE OUTPUT ON U12
H J1_8 J1_6 J1_4 J1_2 L J1_7 J1_5 J1_3 J1_1 !VERIFY NO SHORTS EXIST ON THE JBUS AND U12
L P1_52 !CHANGE DIRECTION ON U34 (B->A) AND ENABLE THE OUTPUT ON U12
MSG '           Verify data through the DBUS'
MSG '           Clock U12'
H P1_51 !CLOCK U12
L P1_51 !CLOCK U12
OFF JBUS !READ DATA FROM U12 OUT THROUGH DBUS
H P1_52 !CHANGE DIRECTION ON U34 (A->B) AND DISABLE THE OUTPUT ON U12
!STIMULUS USED TO GENERATE 'TAP' FILES ASSOCIATED WITH DYNAMIC STIMULUS
.BFR
HIGHSPEED
USE TSET 1
CPP = 1
```

```
L P1_110 P1_109 P1_108 P1_107 P1_106 P1_105 P1_104 P1_103;
H P1_110 P1_109 P1_108 P1_107 P1_106 P1_105 P1_104 P1_103;
!
END
```

C.1.5 Timing.pat

This file lists the dynamic stimulus for the example circuit.

```
SET TSET 1 CLOCK = 300NS
   PHASE 1 ASSERT = 0NS  RETURN = 285NS
   WINDOW 1 OPEN = 200NS CLOSE = 220NS;

SET DIGITAL (P1_110)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_109)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_108)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_107)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_106)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_105)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_104)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_103)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (J1_8)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (J1_7)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (J1_6)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (J1_5)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (J1_4)
   PHASE = 1
   FORMAT = $NRET;
```

```
SET DIGITAL (J1_3)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (J1_2)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (J1_1)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_50)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_49)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_48)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_52)
   PHASE = 1
   FORMAT = $NRET;

SET DIGITAL (P1_51)
   PHASE = 1
   FORMAT =$NRET;

SET PHASE 1 TRIGGER = $T0CLK;

SET DIGITAL (P1_110.)
   WINDOW = 1;

SET DIGITAL (P1_109.)
   WINDOW = 1;

SET DIGITAL (P1_108.)
   WINDOW = 1;

SET DIGITAL (P1_107.)
   WINDOW = 1;

SET DIGITAL (P1_106.)
   WINDOW = 1;

SET DIGITAL (P1_105.)
   WINDOW = 1;

SET DIGITAL (P1_104.)
   WINDOW = 1;

SET DIGITAL (P1_103.)
   WINDOW = 1;

SET DIGITAL (J1_8.)
   WINDOW = 1;

SET DIGITAL (J1_7.)
   WINDOW = 1;
```

```
SET DIGITAL (J1_6.)
   WINDOW = 1;

SET DIGITAL (J1_5.)
   WINDOW = 1;

SET DIGITAL (J1_4.)
   WINDOW = 1;

SET DIGITAL (J1_3.)
   WINDOW = 1;

SET DIGITAL (J1_2.)
   WINDOW = 1;

SET DIGITAL (J1_1.)
   WINDOW = 1;
```

C.2 Resultant DTIF compliant output files

C.2.1 HEADER file

```
HEADER                    1   3EXAMPLE                  5-DEC-1997 10:03
 5-DEC-1997                   MODEL COMPILATION DATE
       21                     NUMBER OF PI'S
       16                     NUMBER OF PO'S
       29                     NUMBER OF PATTERNS
       21                     NUMBER OF PACKAGES
       21                     HIGHEST USER COMPONENT NUMBER
        6                     NUMBER OF COMPONENT TYPES
                              (NO LONGER IN USE)
                              (NO LONGER IN USE)
       16                     NUMBER OF WIRED NETS
       76                     NUMBER OF USER NODES
      105                     HIGHEST USER NODE NUMBER
       38                     NUMBER OF FILES ON THIS TAPE
      159                     NUMBER OF FAULTS CONSIDERED (FD ONLY)
      121                     NUMBER OF FAULTS DEFINITELY DETECTED (FD ONLY)
       32                     NUMBER OF FAULTS POSSIBLY DETECTED (FD ONLY)
       10                     PATTERN DETECTION LIMIT (FD ONLY)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
                              (SPARE)
HEADER                    1
STIMULUS                  2
PO_RESPONSE               3
PI_NAMES                  4
PO_NAMES                  5
MAIN_MODEL                6
COMPONENT_TYPE            7
USER_NODE                 8
INPUT_PIN_NAMES           9
OUTPUT_PIN_NAMES         10
```

```
NEAR_FROMS_POINTERS        11
NEAR_FROMS                 12
EVENT                      13
SETTLED_STATE_ONLY         14
SETTLED_STATE_&_PULSES     15
NODE_SOURCE                16
STEPS                      17
F.D._POPATS                18
F.D._FAULT_SIGNATURES      19
F.D._PRINT_STRINGS         20
TRISTATE_FROMS_POINTERS    21
TRISTATE_FROMS             22
PSEUDOPI_NAMES             23
TIMING_SETS                24
TIMING_PER_PATTERN         25
PHASE_CONNECTIONS          26
AUXILIARY_PIN_NAMES        27
PI_FORMATS                 28
FORMAT_ATTRIBUTES          29
F.D._CROSS_REFERENCE       30
PROBETAG_DEFINITIONS       31
PROBETAG_ASSIGNMENTS       32
BURSTS                     33
STIMULUS_TEXT              34
NODE_NAMES                 35
EVENTS_INIT                36
EQUIV_FAULTS               37
PROBE_DETECTION            38
F.D._EQUIV_SETS            39
```

C.2.2 STIMULUS file

```
STIMULUS                    2   2EXAMPLE                  5-DEC-1997 10:03
        21         29          1        29
444444444444444444444
333434444444444444444
333432222222244444444
333432222222234343434
333432222222244444444
333432222222243434343
333432222222244444444
333432222222222222222
333434444444422222222
333334444444422222222
333334444444422222222
333334444444422222222
333332222222222222222
333432222222222222222
333433333333322222222
333333333333322222222
333343333333322222222
333333333333322222222
333332222222222222222
333432222222222222222
333434242424222222222
333434343434322222222
333334343434322222222
333344343434322222222
333334343434322222222
333332222222222222222
333432222222222222222
333432222222233333333
333432222222244444444
```

C.2.3 PO_RESPONSE file

```
PO_RESPONSE              3    2EXAMPLE                 5-DEC-1997  10:03
        16         29         1         29
4444444444444444
4444444444444444
4444444444444444
3434343434343434
4444444444444444
4343434343434343
4444444444444444
1111111122222222
4444444422222222
4444444444444444
4444444444444444
4444444444444444
4444444444444444
1111111122222222
3333333322222222
3333333333333333
3333333333333333
3333333333333333
3333333333333333
1111111122222222
4141414122222222
4343434322222222
4343434343434343
4343434343434343
4343434343434343
4343434343434343
1111111122222222
3333333333333333
4444444444444444
```

C.2.4 PI_NAMES file

```
PI_NAMES          4   2EXAMPLE        5-DEC-1997 10:03
    21    17
P1_50             1    0
P1_49             2    0
P1_48             3    0
P1_52             4    0
P1_51             5    0
J1_8              6    1
J1_7              7    2
J1_6              8    3
J1_5              9    4
J1_4             10    5
J1_3             11    6
J1_2             12    7
J1_1             13    8
P1_110           14    9
P1_109           15   10
P1_108           16   11
P1_107           17   12
P1_106           18   13
P1_105           19   14
P1_104           20   15
P1_103           21   16
```

C.2.5 PO_NAMES file

```
PO_NAMES                  5   2EXAMPLE                 5-DEC-1997 10:03
        16     17
J1_8.                      80     1
J1_7.                      81     2
J1_6.                      82     3
J1_5.                      83     4
J1_4.                      84     5
J1_3.                      85     6
J1_2.                      86     7
J1_1.                      87     8
P1_110.                    88     9
P1_109.                    89    10
P1_108.                    90    11
P1_107.                    91    12
P1_106.                    92    13
P1_105.                    93    14
P1_104.                    94    15
P1_103.                    95    16
```

C.2.6 MAIN_MODEL file

```
MAIN_MODEL                6   2EXAMPLE                 5-DEC-1997 10:03
        21
U2                         1    6    8      1     7          1
U3                         1    6    8     15    21          2
U12                        2   10    8     29    39          3
U34                        3    2   16     47    49          4
U35                        4   10    8     65    75          5
WN$_1                      5    3    1     83    86          6
WN$_2                      5    3    1     87    90          7
WN$_3                      5    3    1     91    94          8
WN$_4                      5    3    1     95    98          9
WN$_5                      5    3    1     99   102         10
WN$_6                      5    3    1    103   106         11
WN$_7                      5    3    1    107   110         12
WN$_8                      5    3    1    111   114         13
WN$_9                      6    2    1    115   117         14
WN$_10                     6    2    1    118   120         15
WN$_11                     6    2    1    121   123         16
WN$_12                     6    2    1    124   126         17
WN$_13                     6    2    1    127   129         18
WN$_14                     6    2    1    130   132         19
WN$_15                     6    2    1    133   135         20
WN$_16                     6    2    1    136   138         21
```

C.2.7 COMPONENT_TYPE file

```
COMPONENT_TYPE            7   4EXAMPLE                 5-DEC-1997 10:03
         6
SN74LS138                  6    8     1     1      1
SN54LS374                 10    8     7     9      2
SN54LS245                  2   16    17    17      3
SN54LS244                 10    8    19    33      4
WN_3                       3    1    29    41      5W
WN_2                       2    1    32    42      6W
```

C.2.8 USER_NODE file

```
USER_NODE                 8   2EXAMPLE                  5-DEC-1997  10:03
       105        18
        78        79        72       105         4       104        32        33
        34        35        36        37        38        39        78        79
        72       105         5       104        40        41        42        43
        44        45        46        47        32        87        86        85
        84        83        82        81        80        40        48        49
        50        51        52        53        54        55         4       104
        56        57        58        59        60        61        62        63
        64        65        66        67        68        69        70        71
       104         3       100       100       100       104       100       100
         1         2        72        73        74        75        76        77
        78        79         6        71        55        80         7        70
        54        81         8        69        53        82         9        68
        52        83        10        67        51        84        11        66
        50        85        12        65        49        86        13        64
        48        87        14        56        88        15        57        89
        16        58        90        17        59        91        18        60
        92        19        61        93        20        62        94        21
        63        95
```

C.2.9 INPUT_PIN_NAMES file

```
INPUT_PIN_NAMES           9   2EXAMPLE                  5-DEC-1997  10:03
        33         7
               1               2               3               6               4
               5               1               3               4               7
               8              13              14              17              18
              11               1              19               1               2
               4               6               8              19              11
              13              15              17              $1              $2
              $3              $1              $2
```

C.2.10 OUTPUT_PIN_NAMES file

```
OUTPUT_PIN_NAMES         10   2EXAMPLE                  5-DEC-1997  10:03
        42         9
              15              14              13              12              11
              10               9               7               2               5
               6               9              12              15              16
              19               2               3               4               5
               6               7               8               9              11
              12              13              14              15              16
              17              18              18              16              14
              12               9               7               5               3
             OUT             OUT
```

C.2.11 NEAR_FROMS_POINTERS file

```
NEAR_FROMS_POINTERS      11   2EXAMPLE                  5-DEC-1997  10:03
   6     1   6     7   6    13   6    19   6    25   6    31   6    37   6    43
   3    49   3    52   3    55   3    58   3    61   3    64   3    67   3    70
   3    73   3    76   3    79   3    82   3    85   3    88   3    91   3    94
   3    97   3   100   3   103   3   106   3   109   3   112   3   115   3   118
   2   121   2   123   2   125   2   127   2   129   2   131   2   133   2   135
   3   137   2   140
```

C.2.12 NEAR_FROMS file

```
NEAR_FROMS                  12    2EXAMPLE                   5-DEC-1997  10:03
    1    2    3    5    6    4    2    3    5    6    1    4    1    3    5    6    2    4    3    5
    6    1    2    4    1    2    5    6    3    4    2    5    6    3    1    4    1    5    6    3
    2    4    5    6    4    2    1    3    1    2   10    1    3   10    1    4   10    1    5   10
    1    6   10    1    7   10    1    8   10    1    9   10  -16    2    1  -15    2    1  -14    2
    1  -13    2    1  -12    2    1  -11    2    1  -10    2    1   -9    2    1    1   -8    2    1
   -7    2    1   -6    2    1   -5    2    1   -4    2    1   -3    2    1   -2    2    1   -1    2
    2    1    3    1    4    1    5    1    7    6    8    6    9    6   10    6    1    2    3    1
    2
```

C.2.13 EVENT file

```
EVENT                       13    2EXAMPLE                   5-DEC-1997  10:03
           1  -12
0C1H2H3H4H5H6H7H8H9H10H11H12H13H14H15H16H17H18H19H20H21H10C88H89H90H91H92H93H94H
95H7528A32H33H34H35H36H37H38H39H40H41H42H43H44H45H46H47H462A72H78H79H10A64H65H66
H67H68H69H70H71H10A80H81H82H83H84H85H86H87H980A56K57K58K59K60K61K62K63K6574A48K4
9K50K51K52K53K54K55K2426C72H78H79H10C64H65H66H67H68H69H70H71H10C80H81H82H83H84H8
5H86H87H7980C32H33H34H35H36H37H38H39H40H41H42H43H44H45H46H47H2000C56K57K58K59K60
K61K62K63K18000C48K49K50K51K52K53K54K55K46001@1P0C1E2E3E5E4447A47E3553A72E78E79E
1261C47N8739C72E78E79E260A40E41N42N43N44N45N46N6070A47HB41H42H43H44H45H46H13670D
47E6999D41N42N43N44N45N46N1C41H42H43H44H45H46H47H12000C40E57001@2P0C6K7K8K9K10K1
1K12K13K1@3P0C14E16E18E20E10C88E90E92E94E8000A65E67E69E71E10A80E82E84E86E9990C65
E67E69E71E10C80E82E84E86E18021@4P0C14H16H18H20H10C88H90H92H94H8000A65H67H69H71H1
0A80H82H84H86H9990C65H67H69H71H10C80H82H84H86H18021@5P0C15E17E19E21E10C89E91E93E
95E8000A64E66E68E70E10A81E83E85E87E9990C64E66E68E70E10C81E83E85E87E18021@6P0C15H
17H19H21H10C89H91H93H95H8000A64H66H68H70H10A81H83H85H87H9990C64H66H68H70H10C81H8
3H85H87H18021@7P0C14K15K16K17K18K19K20K21K10C88K89K90K91K92K93K94K95K8000A64N65N
66N67N68N69N70N71N10A80N81N82N83N84N85N86N87N9990C64N65N66N67N68N69N70N71N10C80N
81N82N83N84N85N86N87N18021@8P0C6H7H8H9H10H11H12H13H10C80H81H82H83H84H85H86H87H11
@9P0C4E4447A32E4553A64K65K66K67K68K69K70K71K6630A56H57H58H59H60H61H62H63H10A88H8
9H90H91H92H93H94H95H4197A48N49N50N51N52N53N54N55N8163C64K65K66K67K68K69K70K71K10
000C32E2000C56H57H58H59H60H61H62H63H10C88H89H90H91H92H93H94H95H25990C48N49N50N51
N52N53N54N55N66001@10P0C5H7538A40H8036A48H49H50H51H52H53H54H55H10426C40H28000C48
H49H50H51H52H53H54H55H54001@11P0C5E4447A40E33553C40E38001@12P0C6K7K8K9K10K11K12K
13K1@13P0C4H7538A32H498A56K57K58K59K60K61K62K63K10A88K89K90K91K92K93K94K95K7528A
48K49K50K51K52K53K54K55K10A80K81K82K83K84K85K86K87K46A64N65N66N67N68N69N70N71N10
B80N81N82N83N84N85N86N87N10360C32H2000C56K57K58K59K60K61K62K63K10C88K89K90K91K92
K93K94K95K11990C64N65N66N67N68N69N70N71N10C80N81N82N83N84N85N86N87N13990C48K49K5
0K51K52K53K54K55K54001@14P0C6E7E8E9E10E11E12E13E10C80E81E82E83E84E85E86E87E11@15
P0C4E4447A32E4553A64K65K66K67K68K69K70K71K9230A56E57E58E59E60E61E62E63E10A88E89E
90E91E92E93E94E95E1597A48H49H50H51H52H53H54H55H8163C64K65K66K67K68K69K70K71K1000
0C32E2000C56E57E58E59E60E61E62E63E10C88E89E90E91E92E93E94E95E23990C48H49H50H51H5
2H53H54H55H64001@16P0C5H7538A40H12890A48E49E50E51E52E53E54E55E5572C40H28000C48E4
9E50E51E52E53E54E55E54001@17P0C5E4447A40E33553C40E38001@18P0C6K7K8K9K10K11K12K13
K1@19P0C4H7538A32H1462A56K57K58K59K60K61K62K63K10A88K89K90K91K92K93K94K95K5728A4
8K49K50K51K52K53K54K55K10A80K81K82K83K84K85K86K87K2262A64N65N66N67N68N69N70N71N1
0B80N81N82N83N84N85N86N87N7980C56K57K58K59K60K61K62K63K10C88K89K90K91K92K93K94K9
5K990C32H17010C64N65N66N67N68N69N70N71N10C80N81N82N83N84N85N86N87N2980C48K49K50K
51K52K53K54K55K46001@20P0C6H8H10H12H10C80H82H84H86H11@21P0C7E9E11E13E10C81E83E85
E87E11@22P0C4E4447A32E4553A64K65K66K67K68K69K70K71K6630A56H58H60H62H10A88H90H92H
94H2590A57E59E61E63E10A89E91E93E95E1957A48E49E50E51E52E53E54E55E7803C64K65K66K67
K68K69K70K71K10000C32E2000C56H57E58H59E60H61E62H63E10C88H89E90H91E92H93E94H95E25
990C48E49E50E51E52E53E54E55E66001@23P0C5H7538A40H8036A49H51H53H55H10426C40H28000
C49H51H53H55H54001@24P0C5E4447A40E33553C40E38001@25P0C6K7K8K9K10K11K12K13K1@26P0
C4H7538A32H498A56K58K60K62K10A88K90K92K94K954A57K59K61K63K10A89K91K93K95K5728A48
K50K52K54K10A81K83K85K87K826A49K51K53K55K10A80K82K84K86K46A65N67N69N71N10B80N82N
84N86N1370A64N66N68N70N10B81N83N85N87N7980C57K59K61K63K10C89K91K93K95K990C32H200
0C56K58K60K62K10C88K90K92K94K11990C65N67N69N71N10C80N82N84N86N3000C64N66N68N70N1
0C81N83N85N87N2980C48K50K52K54K8000C49K51K53K55K54001@27P1Q0C14E15E16E17E18E19E2
0E21E10C88E89E90E91E92E93E94E95E8000A64E65E66E67E68E69E70E71E10A80E81E82E83E84E8
```

```
5E86E87E9990C64E65E66E67E68E69E70E71E10C80E81E82E83E84E85E86E87E300000@28P0C14H1
5H16H17H18H19H20H21H10C88H89H90H91H92H93H94H95H8000A64H65H66H67H68H69H70H71H10A8
0H81H82H83H84H85H86H87H9990C64H65H66H67H68H69H70H71H10C80H81H82H83H84H85H86H87H3
00000@29P2QR
```

C.2.14 SETTLED_STATE_ONLY file

```
SETTLED_STATE_ONLY        14    2EXAMPLE                  5-DEC-1997 10:03
*9*2.16 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63*4.69 1 2 3 4 5 6 7 8 9 1
0 11 12 13 14 15 16 17 18 19 20 21 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
47 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
 90 91 92 93 94 95*8*3.8 1 2 3 5 40 72 78 79*8*2.8 6 7 8 9 10 11 12 13*8*3.16 14
 16 18 20 65 67 69 71 80 82 84 86 88 90 92 94*8*4.16 14 16 18 20 65 67 69 71 80
82 84 86 88 90 92 94*8*3.16 15 17 19 21 64 66 68 70 81 83 85 87 89 91 93 95*8*4.
16 15 17 19 21 64 66 68 70 81 83 85 87 89 91 93 95*8*1.16 64 65 66 67 68 69 70 7
1 80 81 82 83 84 85 86 87*2.16 14 15 16 17 18 19 20 21 88 89 90 91 92 93 94 95*8
*4.16 6 7 8 9 10 11 12 13 80 81 82 83 84 85 86 87*8*1.8 48 49 50 51 52 53 54 55*
2.8 64 65 66 67 68 69 70 71*3.2 4 32*4.16 56 57 58 59 60 61 62 63 88 89 90 91 92
 93 94 95*8*4.10 5 40 48 49 50 51 52 53 54 55*8*3.2 5 40*8*2.8 6 7 8 9 10 11 12
13*8*1.16 64 65 66 67 68 69 70 71 80 81 82 83 84 85 86 87*2.24 48 49 50 51 52 53
 54 55 56 57 58 59 60 61 62 63 88 89 90 91 92 93 94 95*4.2 4 32*8*3.16 6 7 8 9 1
0 11 12 13 80 81 82 83 84 85 86 87*8*2.8 64 65 66 67 68 69 70 71*3.18 4 32 56 57
 58 59 60 61 62 63 88 89 90 91 92 93 94 95*4.8 48 49 50 51 52 53 54 55*8*3.8 48
49 50 51 52 53 54 55*4.2 5 40*8*3.2 5 40*8*2.8 6 7 8 9 10 11 12 13*8*1.16 64 65
66 67 68 69 70 71 80 81 82 83 84 85 86 87*2.24 48 49 50 51 52 53 54 55 56 57 58
59 60 61 62 63 88 89 90 91 92 93 94 95*4.2 4 32*8*4.8 6 8 10 12 80 82 84 86*8*3.
8 7 9 11 13 81 83 85 87*8*2.8 64 65 66 67 68 69 70 71*3.18 4 32 48 49 50 51 52 5
3 54 55 57 59 61 63 89 91 93 95*4.8 56 58 60 62 88 90 92 94*8*4.6 5 40 49 51 53
55*8*3.2 5 40*8*2.8 6 7 8 9 10 11 12 13*8*1.16 64 65 66 67 68 69 70 71 80 81 82
83 84 85 86 87*2.24 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 88 89 90 91
92 93 94 95*4.2 4 32*8*10*9*3.32 14 15 16 17 18 19 20 21 64 65 66 67 68 69 70 71
 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95*8*4.32 14 15 16 17 18 19 20 21
64 65 66 67 68 69 70 71 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95*8*10*11
```

C.2.15 SETTLED_STATE_&_PULSES file

```
SETTLED_STATE_&_PULSES    15    2EXAMPLE                  5-DEC-1997 10:03
        10
*9*4.26 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 73 74 75 76 77*17.
16 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63*19.43 32 33 34 35 36 37 38 39
 40 41 42 43 44 45 46 47 64 65 66 67 68 69 70 71 72 78 79 80 81 82 83 84 85 86 8
7 88 89 90 91 92 93 94 95*8*3.8 1 2 3 5 40 72 78 79*4.32 32 33 34 35 36 37 38 39
 64 65 66 67 68 69 70 71 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95*8*4.7 4
1 42 43 44 45 46 47*17.8 6 7 8 9 10 11 12 13*8*3.16 14 16 18 20 65 67 69 71 80 8
2 84 86 88 90 92 94*8*4.16 14 16 18 20 65 67 69 71 80 82 84 86 88 90 92 94*8*3.1
6 15 17 19 21 64 66 68 70 81 83 85 87 89 91 93 95*8*4.16 15 17 19 21 64 66 68 70
 81 83 85 87 89 91 93 95*8*16.16 64 65 66 67 68 69 70 71 80 81 82 83 84 85 86 87
*17.16 14 15 16 17 18 19 20 21 88 89 90 91 92 93 94 95*8*4.8 6 7 8 9 10 11 12 13
*19.8 80 81 82 83 84 85 86 87*8*3.2 4 32*4.24 56 57 58 59 60 61 62 63 80 81 82 8
3 84 85 86 87 88 89 90 91 92 93 94 95*16.8 48 49 50 51 52 53 54 55*17.8 64 65 66
 67 68 69 70 71*8*4.2 5 40*19.8 48 49 50 51 52 53 54 55*8*3.2 5 40*4.8 48 49 50
51 52 53 54 55*8*17.8 6 7 8 9 10 11 12 13*8*4.2 4 32*16.16 64 65 66 67 68 69 70
71 80 81 82 83 84 85 86 87*17.24 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 88 89 90 91 92 93 94 95*8*3.8 6 7 8 9 10 11 12 13*18.8 80 81 82 83 84 85 86 87*
8*3.26 4 32 56 57 58 59 60 61 62 63 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
 95*4.8 48 49 50 51 52 53 54 55*17.8 64 65 66 67 68 69 70 71*8*3.8 48 49 50 51 5
2 53 54 55*4.2 5 40*8*3.2 5 40*8*17.8 6 7 8 9 10 11 12 13*8*4.2 4 32*16.16 64 65
 66 67 68 69 70 71 80 81 82 83 84 85 86 87*17.24 48 49 50 51 52 53 54 55 56 57 5
8 59 60 61 62 63 88 89 90 91 92 93 94 95*8*4.4 6 8 10 12*19.4 80 82 84 86*8*3.4
7 9 11 13*4.4 80 82 84 86*18.4 81 83 85 87*8*3.22 4 32 48 49 50 51 52 53 54 55 5
7 59 61 63 81 83 85 87 89 91 93 95*4.8 56 58 60 62 88 90 92 94*17.8 64 65 66 67
68 69 70 71*8*4.6 5 40 49 51 53 55*8*3.2 5 40*8*17.8 6 7 8 9 10 11 12 13*8*4.2 4
```

```
 32*16.16 64 65 66 67 68 69 70 71 80 81 82 83 84 85 86 87*17.24 48 49 50 51 52 5
3 54 55 56 57 58 59 60 61 62 63 88 89 90 91 92 93 94 95*8*10*9*18.32 14 15 16 17
 18 19 20 21 64 65 66 67 68 69 70 71 80 81 82 83 84 85 86 87 88 89 90 91 92 93 9
4 95*8*4.16 64 65 66 67 68 69 70 71 80 81 82 83 84 85 86 87*19.16 14 15 16 17 18
 19 20 21 88 89 90 91 92 93 94 95*8*10*11
```

C.2.16 NODE_SOURCE file

```
NODE_SOURCE                 16    2EXAMPLE                  5-DEC-1997  10:03
       105
    0  1     0  2     0  3     0  4     0  5     0  6     0  7     0  8     0  9     0 10
    0 11     0 12     0 13     0 14     0 15     0 16     0 17     0 18     0 19     0 20
    0 21     0  0     0  0     0  0     0  0     0  0     0  0     0  0     0  0     0  0
    0  0     1  1     1  2     1  3     1  4     1  5     1  6     1  7     1  8     2  1
    2  2     2  3     2  4     2  5     2  6     2  7     2  8     3  1     3  2     3  3
    3  4     3  5     3  6     3  7     3  8     4  1     4  2     4  3     4  4     4  5
    4  6     4  7     4  8     4  9     4 10     4 11     4 12     4 13     4 14     4 15
    4 16     5  1     5  2     5  3     5  4     5  5     5  6     5  7     5  8     6  1
    7  1     8  1     9  1    10  1    11  1    12  1    13  1    14  1    15  1    16  1
   17  1    18  1    19  1    20  1    21  1     0 23     0 25     0 26     0 27     0 28
    0 29     0 30     0 31     0 24     0 22
```

C.2.17 STEPS file

```
STEPS                       17    2EXAMPLE                  5-DEC-1997  10:03
        10
         1         -2          3          4          6         10        -11         13
        16         19
```

C.2.18 F.D._POPATS file

```
F.D._POPATS                 18    2EXAMPLE                  5-DEC-1997  10:03
   82
   1      1    2      1    3      1    4      1    5      1    6      1    7      1    8      1
   9      1   10      1   11      1   12      1   13      1   14      1   15      1   16      1
   1      3    2      3    3      3    4      3    5      3    6      3    7      3    8      3
   1      4    3      4    5      4    7      4    9      4   11      4   13      4   15      4
   2      6    4      6    6      6    8      6   10      6   12      6   14      6   16      6
   9     10   10     10   11     10   12     10   13     10   14     10   15     10   16     10
   1     13    2     13    3     13    4     13    5     13    6     13    7     13    8     13
   1     15    2     15    3     15    4     15    5     15    6     15    7     15    8     15
   9     16   10     16   11     16   12     16   13     16   14     16   15     16   16     16
   1     19    2     19    3     19    4     19    5     19    6     19    7     19    8     19
   1     23    1     28
```

C.2.19 F.D._FAULT_SIGNATURES file

```
F.D._FAULT_SIGNATURES       19    2EXAMPLE                  5-DEC-1997  10:03
    94     33
    48      3      1     17     41
    48      3      2     18     42
    48      3      3     19     43
    48      3      4     20     44
    48      3      5     21     45
    48      3      6     22     46
    48      3      7     23     47
    48      3      8     24     48
    56      3      9     17     41
    56      3     10     18     42
```

56	3	11	19	43		
56	3	12	20	44		
56	3	13	21	45		
56	3	14	22	46		
56	3	15	23	47		
56	3	16	24	48		
-1	4	25	29	65	82	
-1	3	26	30	67		
-1	3	27	31	69		
-1	3	28	32	71		
82	3	33	37	66		
82	3	34	38	68		
82	3	35	39	70		
82	3	36	40	72		
-1	4	17	-41	-49	-81	
-1	3	18	-42	-50		
-1	3	19	-43	-51		
-1	3	20	-44	-52		
-1	3	21	-45	-53		
-1	3	22	-46	-54		
-1	3	23	-47	-55		
-1	3	24	-48	-56		
-1	4	25	-65	-73	82	
-1	3	26	-67	-75		
-1	3	27	-69	-77		
-1	3	28	-71	-79		
-1	3	33	-66	-74		
-1	3	34	-68	-76		
-1	3	35	-70	-78		
-1	3	36	-72	-80		
-1	1	41				
-1	1	42				
-1	1	43				
-1	1	44				
-1	1	45				
-1	1	46				
-1	1	47				
-1	1	48				
-1	2	-17	49			
-1	2	-18	50			
-1	2	-19	51			
-1	2	-20	52			
-1	2	-21	53			
-1	2	-22	54			
-1	2	-23	55			
-1	2	-24	56			
-1	5	25	57	65	73	82
81	4	33	58	66	74	
-1	4	26	59	67	75	
81	4	34	60	68	76	
-1	4	27	61	69	77	
81	4	35	62	70	78	
-1	4	28	63	71	79	
81	4	36	64	72	80	
-1	1	65				
-1	1	66				
-1	1	67				
-1	1	68				
-1	1	69				
-1	1	70				
-1	1	71				
-1	1	72				
-1	1	73				
-1	1	74				

```
   -1     1    75
   -1     1    76
   -1     1    77
   -1     1    78
   -1     1    79
   -1     1    80
   -1    17   -17   -18   -19   -20   -21   -22   -23   -24   -25   -26   -27
  -28   -33   -34   -35   -36    82
   -1    25   -17   -18   -19   -20   -21   -22   -23   -24   -25   -26   -27
  -28   -33   -34   -35   -36   -57   -58   -59   -60   -61   -62   -63   -64
  -82
   -1     4   -25   -57    73   -82
   -1     3   -26   -59    75
   -1     3   -27   -61    77
   -1     3   -28   -63    79
   -1     3   -33   -58    74
   -1     3   -34   -60    76
   -1     3   -35   -62    78
   -1     3   -36   -64    80
   -1    32   -41   -42   -43   -44   -45   -46   -47   -48   -49   -50   -51
  -52   -53   -54   -55   -56   -65   -66   -67   -68   -69   -70   -71   -72
  -73   -74   -75   -76   -77   -78   -79   -80
   -1    33   -17   -18   -19   -20   -21   -22   -23   -24   -25   -26   -27
  -28   -33   -34   -35   -36   -41   -42   -43   -44   -45   -46   -47   -48
  -65   -66   -67   -68   -69   -70   -71   -72   -82
   -1    16   -49   -50   -51   -52   -53   -54   -55   -56   -73   -74   -75
  -76   -77   -78   -79   -80
   -1    33   -41   -42   -43   -44   -45   -46   -47   -48   -49   -50   -51
  -52   -53   -54   -55   -56   -65   -66   -67   -68   -69   -70   -71   -72
  -73   -74   -75   -76   -77   -78   -79   -80   -81
```

C.2.20 F.D._PRINT_STRINGS file

```
F.D._PRINT_STRINGS       20    3EXAMPLE                  5-DEC-1997  10:03
    94   12    28
     2      2
     9<^>J1_8@0
    10<^>J1_8.@0
     2      2
     9<^>J1_7@0
    10<^>J1_7.@0
     2      2
     9<^>J1_6@0
    10<^>J1_6.@0
     2      2
     9<^>J1_5@0
    10<^>J1_5.@0
     2      2
     9<^>J1_4@0
    10<^>J1_4.@0
     2      2
     9<^>J1_3@0
    10<^>J1_3.@0
     2      2
     9<^>J1_2@0
    10<^>J1_2.@0
     2      2
     9<^>J1_1@0
    10<^>J1_1.@0
     2      2
    11<^>P1_110@0
    12<^>P1_110.@0
     2      2
```