

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

Copyright © 1999 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 10 March 1999. Printed in the United States of America.

Print: ISBN 0-7381-1553-3 SH94715
PDF: ISBN 0-7381-1554-1 SS94715

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS**.”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter 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. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

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.

Participants

The ATPG Subcommittee consisted of the following members who provided significant input toward the development of this standard:

Mukund U. Modi, *ATPG US Co-chair*
Jose De La Cruz, *Working Group Chair*
Harold E. Davis, *Secretary*

Guy Adam
Jean-Yves Barbier
Ann F. Bautista
Aldo Bernal
Michael Blair
Joanne K. Buckley
Antonius Bunsen
Kenneth M. Butler
Larry D. Carpenter
Alexander Cherfas
Richard Churcher
Tim Coggins
Ronald Cox
David Dalbacka
Danny C. Davis
Bernd Dinklage
Bernard Dugas
Michael T. Ellis
William Brit Frank

Arnold M. Greenspan
Joel Guerra
Arthur Hann
Gary L. Hardenburg
Robert G. Hayes
Raymond L. Heather
Ashley M. B. Hulme
Patrick Johnson
Al Karalius
Larry V. Kirkland
Robert Madison
Gregory A. Maston
Mark McCooley
Harry McGuckin
William P. Neblett
James Nicholson
Hans Obermeier
Leslie A. Orlidge

Gaspare Pantano
Edward G. Perkins
Jean Pouilly
Narayanan Ramachandran
William C. Rodriguez
David Roggendorff
David E. Rolince
Eric Sacher
Howard Savage
John W. Sheppard
William R. Simpson*
Butch Sneade
Joseph J. Stanco
William John Taylor
Janette Tuttle
Christine M. West**
William A. Whitaker
William W. Young
Jehuda Ziegler

* Voting IEEE Standards Board Liaison

** Contributing Non-Voting Member

¹IEEE 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.

The following individuals are non-voting IEEE-SA Standards Board Liaisons:

William Brenner

Jim Isaak
Robert C. Rassa

Oscar Sepp

The following members of the balloting committee voted on this standard:

Guy Adam
Jean-Yves Barbier
Anne F. Bautista
Aldo Bernal
Joanne K. Buckley
Antonius Bunsen
Kenneth M. Butler
Larry D. Carpenter
Alexander Cherfas
Tim Coggins
Ronald Cox
David Dalbacka
Danny C. Davis
Harold E. Davis
Jose De La Cruz
Bernd Dinklage

Bernard Dugas
William Brit Frank
Arnold M. Greenspan
Arthur Hann
Robert G. Hayes
Ashley M. B. Hulme
Al Karalius
Larry V. Kirkland
Gregory A. Maston
Mukund U. Modi
William P. Neblett
James Nicholson
Hans Obermeier
Leslie A. Orlidge
Gaspard Pantano

Edward G. Perkins
Jean Pouilly
Narayanan Ramachandran
William C. Rodriguez
David E. Rolince
Eric Sacher
Howard Savage
John W. Sheppard
William R. Simpson
Butch Sneade
Joseph J. Stanco
William John Taylor
Janette Tuttle
William A. Whitaker
William W. Young
Jehuda Ziegler

When the IEEE-SA Standards Board approved this standard on 8 December 1998, it had the following membership:

Richard J. Holleman, *Chair*

Donald N. Heirman, *Vice Chair*

Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon

James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry

L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Valerie E. Zelenty
IEEE Standards Project Editor

Contents

1.	Overview	1
1.1	Scope.....	1
1.2	Purpose.....	1
1.3	Application.....	1
2.	References.....	2
3.	Definitions and acronyms	2
3.1	Definitions.....	2
3.2	Acronyms	4
4.	Data organization overview of the DTIF standard environment.....	4
4.1	UUT Model Group.....	5
4.2	Stimulus and Response Group.....	5
4.3	Fault Dictionary Group	5
4.4	Probe Group	5
5.	File specifications.....	6
5.1	HEADER file	7
5.2	STIMULUS file	9
5.3	PO_RESPONSE file	10
5.4	PI_NAMES file	11
5.5	PO_NAMES file	12
5.6	MAIN_MODEL file.....	13
5.7	COMPONENT_TYPE file.....	14
5.8	USER_NODE file	15
5.9	INPUT_PIN_NAMES file	16
5.10	OUTPUT_PIN_NAMES file	17
5.11	NEAR_FROMS_POINTERS file	18
5.12	NEAR_FROMS file	19
5.13	EVENT file	20
5.14	SETTLED_STATE_ONLY file.....	22
5.15	SETTLED_STATE_&_PULSES file.....	23
5.16	NODE_SOURCE file.....	25
5.17	STEPS file.....	26
5.18	F.D._POPATS file.....	27
5.19	F.D._FAULT_SIGNATURES file	28
5.20	F.D._PRINT_STRINGS file	30
5.21	TRISTATE_FROMS_POINTERS file.....	32
5.22	TRISTATE_FROMS file	33
5.23	PSEUDOPI_NAMES file	34
5.24	TIMING_SETS file.....	35
5.25	TIMING_PER_PATTERN file	37
5.26	PHASE_CONNECTIONS file.....	38
5.27	AUXILIARY_PIN_NAMES file	39
5.28	PI_FORMATS file	40
5.29	FORMAT_ATTRIBUTES file	41

5.30 F.D. _CROSS_REFERENCE file	42
5.31 PROBETAG_DEFINITIONS file	43
5.32 PROBETAG_ASSIGNMENTS file	45
5.33 BURSTS file	46
5.34 STIMULUS_TEXT file	47
5.35 NODE_NAMES file	48
5.36 EVENTS_INIT file	49
5.37 EQUIV_FAULTS file	51
5.38 PROBE_DETECTION file	52
5.39 F.D._EQUIV_SETS file	53
 6. Conformance	 54
6.1 End-to-end test	54
6.2 Diagnostic test using fault dictionary	54
6.3 Diagnostic test using probe	56
 Annex A (informative) Implementation overview	 59
 Annex B (informative) DTIF dependency diagrams	 61
 Annex C (informative) Example circuit	 66

IEEE Standard for 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.**

¹ANSI 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/>).

²IEEE 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).

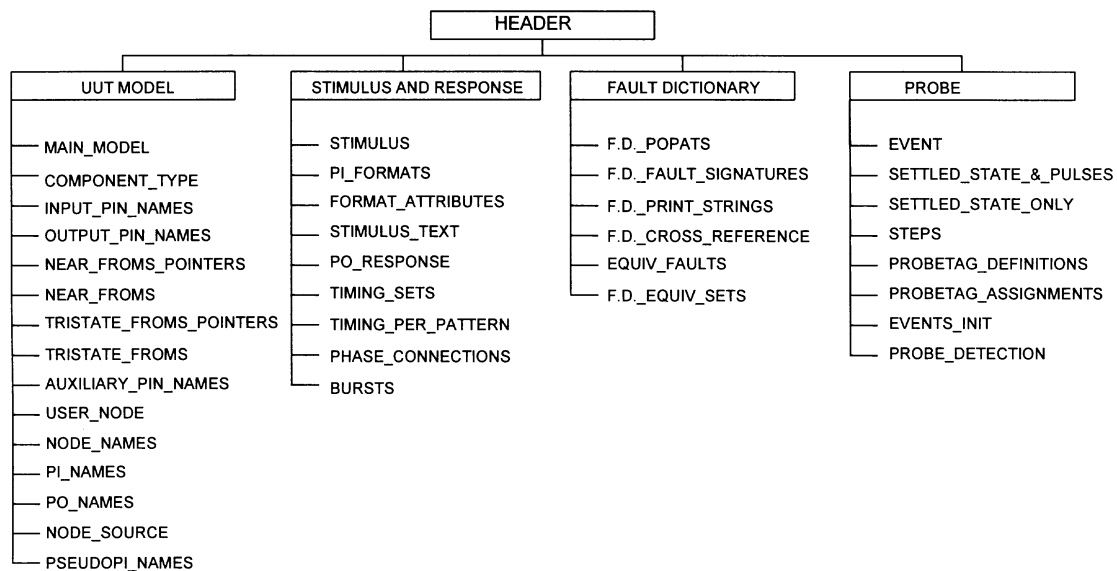


Figure 1—DTIF data files

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

```

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

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
1		1	2	3
5		8	9	1
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
3		6	8	11
4		6	4	5
				9
				2

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	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	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:

<u>op code</u>	<u>Meaning</u>
@	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 1 -12 13 2UnitUnderTestAM 10-JAN-2000 12:00
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 <i>but</i> it <i>definitely</i> pulsed)	
12	X
13	Z
14	0
15	1
(State that the node settled <i>and</i> it <i>possibly</i> 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: nodsourc.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