IEC 62525

Edition 1.0 2007-11

IEEE 1450™

INTERNATIONAL STANDARD

Standard Test Interface Language (STIL) for Digital Test Vector Data

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XG

ICS 25.040;19.080

ISBN 2-8318-9337-2

CONTENTS

FOREWORD 6

IEEE Introduction 9

1. Overview 10

1.1 Scope 12
1.2 Purpose 13

2. References 13

3. Definitions, acronyms, and abbreviations 13

3.1 Definitions 13
3.2 Acronyms and abbreviations 16

4. Structure of this standard 16

5. STIL orientation and capabilities tutorial (informative) 17

5.1 Hello Tester 17
5.2 Basic LS245 22
5.3 STIL timing expressions/"Spec" information 26
5.4 Structural test (scan) 31
5.5 Advanced scan 35
5.6 IEEE Std 1149.1-1990 scan 41
5.7 Multiple data elements per test cycle 46
5.8 Pattern reuse/direct access test 50
5.9 Event data/non-cyclized STIL information 54

6. STIL syntax description 64

6.1 Case sensitivity 64
6.2 Whitespace 64
6.3 Reserved words 64
6.4 Reserved characters 66
6.5 Comments 67
6.6 Token length 67
6.7 Character strings 67
6.8 User-defined name characteristics 68
6.9 Domain names 68
6.10 Signal and group name characteristics 69
6.11 Timing name constructs 69
6.12 Number characteristics 69
6.13 Timing expressions and units (time_expr) 70
6.14 Signal expressions (sigref_expr) 72
6.15 WaveformChar characteristics 73
6.16 STIL name spaces and name resolution 74

7. Statement structure and organization of STIL information 76

7.1 Top-level statements and required ordering 68
7.2 Optional top-level statements 70
7.3 STIL files 70

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

8. STIL statement 79

8.1 STIL syntax 79
8.2 STIL example 79

9. Header block 80

9.1 Header block syntax 80
9.2 Header example 80

10. Include statement 80

10.1 Include statement syntax 81
10.2 Include example 81
10.3 File path resolution with absolute path notation 81
10.4 File path resolution with relative path notation 81

11. UserKeywords statement 82

11.1 UserKeywords statement syntax 82
11.2 UserKeywords example 82

12. UserFunctions statement 82

12.1 UserFunctions statement syntax 83
12.2 UserFunctions example 83

13. Ann statement 83

13.1 Annotations statement syntax 83
13.2 Annotations example 83

14. Signals block 83

14.1 Signals block syntax 84
14.2 Signals block example 86

15. SignalGroups block 86

15.1 SignalGroups block syntax 86
15.2 SignalGroups block example 87
15.3 Default attribute values 87
15.4 Translation of based data into WaveformChar characters 88

16. PatternExec block 89

16.1 PatternExec block syntax 90
16.2 PatternExec block example 90

17. PatternBurst block 90

17.1 PatternBurst block syntax 91
17.2 PatternBurst block example 92

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

18. Timing block and WaveformTable block 92

18.1 Timing and WaveformTable syntax 93
18.2 Waveform event definitions 96
18.3 Timing and WaveformTable example 98
18.4 Rules for timed event ordering and waveform creation 99
18.5 Rules for waveform inheritance 102

19. Spec and Selector blocks 103

19.1 Spec and Selector block syntax 103
19.2 Spec and Selector block example 105

20. ScanStructures block 106

20.1 ScanStructures block syntax 107
20.2 ScanStructures block example 108

21. STIL Pattern data 109

21.1 Cyclized data 109
21.2 Multiple-bit cyclized data 110
21.3 Non-cyclized data 111
21.4 Scan data 111
21.5 Pattern labels 112

22. STIL Pattern statements 112

22.1 Vector (V) statement 112
22.2 WaveformTable (W) statement 113
22.3 Condition (C) statement 113
22.4 Call statement 114
22.5 Macro statement 114
22.6 Loop statement 115
22.7 MatchLoop statement 115
22.8 Goto statement 116
22.9 BreakPoint statements 116
22.10 IDDQTestPoint statement 116
22.11 Stop statement 117
22.12 ScanChain statement 117

23. Pattern block 117

23.1 Pattern block syntax 117
23.2 Pattern initialization 118
23.3 Pattern examples 118

24. Procedures and MacroDefs blocks 118

24.1 Procedures block 119
24.2 Procedures example 120
24.3 MacroDefs block 120
24.4 Scan testing 120
24.5 Procedure and Macro Data substitution 121

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

Annex A (informative) Glossary 125
Annex B (informative) STIL data model 126
Annex C (informative) GNU GZIP reference 131
Annex D (informative) Binary STIL data format 132
Annex E (informative) LS245 design description 136
Annex F (informative) STIL FAQs and language design decisions 138
Annex G (informative) List of participants 142

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

STANDARD TEST INTERFACE LANGUAGE (STIL) FOR DIGITAL TEST VECTOR DATA

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. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements 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.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC 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 transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC/IEEE 62525 has been processed through Technical Committee 93: Design automation.

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

IEEE Std	FDIS	Report on voting
1450(1999)	93/247/FDIS	93/258/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 committee has decided that the contents of this publication will remain unchanged until the maintenance result 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.

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

IEC/IEEE Dual Logo International Standards

This Dual Logo International Standard is the result of an agreement between the IEC and the Institute of Electrical and Electronics Engineers, Inc. (IEEE). The original IEEE Standard was submitted to the IEC for consideration under the agreement, and the resulting IEC/IEEE Dual Logo International Standard has been published in accordance with the ISO/IEC Directives.

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 IEC/IEEE Dual Logo International Standard is wholly voluntary. The IEC and IEEE disclaim 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 IEC or IEEE Standard document.

The IEC and IEEE do not warrant or represent the accuracy or content of the material contained herein, and expressly disclaim 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. IEC/IEEE Dual Logo International Standards documents are supplied “AS IS”.

The existence of an IEC/IEEE Dual Logo International 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 IEC/IEEE Dual Logo International 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 IEC and IEEE are not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Neither the IEC nor IEEE is undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEC/IEEE Dual Logo International Standards or 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 IEC/IEEE Dual Logo International Standards are welcome from any interested party, regardless of membership affiliation with the IEC or 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, P.O. Box 1331, Piscataway, NJ 08855-1331, USA and/or General Secretary, IEC, 3, rue de Varembé, PO Box 131, 1211 Geneva 20, Switzerland.

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; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

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.

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

IEEE Standard Test Interface Language (STIL) for Digital Test Vector Data

Sponsor

Test Technology Standards Committee of the IEEE Computer Society

Approved 18 March 1999

IEEE-SA Standards Board

Abstract: Standard Test Interface Language (STIL) provides an interface between digital test generation tools and test equipment. A test description language is defined that: (a) facilitates the transfer of digital test vector data from CAE to ATE environments; (b) specifies pattern, format, and timing information sufficient to define the application of digital test vectors to a DUT; and (c) supports the volume of test vector data generated from structured tests.
Keywords: automatic test pattern generator (ATPG), built-in self-test (BIST), computer-aided engineering (CAE), cyclize, device under test (DUT), digital test vectors, event, functional vectors, pattern, scan vectors, signal, structural vectors, timed event, waveform, waveshape

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

IEEE Introduction

Standard Test Interface Language (STIL) was initially developed by an ad-hoc consortium of test equipment vendors, computer-aided engineering (CAE) and computer-aided design (CAD) vendors, and integrated circuit (IC) manufacturers, to address the lack of a common solution for transferring digital test data from the generation environment to the test equipment.

The need for a common interchange format for large volumes of digital test data was identified as an overriding priority for the work; as such, the scope of the work was constrained to those aspects of the test environment that contribute significantly to the volume issue, or are necessary to support the comprehension of that data. Binary representations of data were a key consideration in these efforts, resulting in a proposal to incorporate the compression of files as part of this standard.

Limiting the scope of any standards project is a difficult thing to do, especially for a room full of engineers. However, issues that did not impact the scope as identified were dropped from consideration in this version of the standard. Subclause 1.1 covers, specifically, the capabilities that are not intended to be part of this first standard.

Early work in this consortium consisted of identifying the requirements necessary to address this problem and reviewing existing options and languages in the industry. All options proposed fell short of addressing the requirements, and the consortium started to define a new language. This work was executed with heavy leverage from some existing languages and environments, and STIL owes much to the groundwork established by these other languages.

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

STANDARD TEST INTERFACE LANGUAGE (STIL) FOR DIGITAL TEST VECTOR DATA

1. Overview

Standard Test Interface Language (STIL) is a standard language that provides an interface between digital test generation tools and test equipment. STIL may be directly generated as an output language of a test generation tool, or it may be used as an intermediate format for subsequent processing. Figure 1 shows STIL usage in a “pipe” format. This is meant solely as a visual analogy to emphasize the high-volume/high-throughput requirements. It is not meant to represent physical structures or implementation requirements.

STIL is a representation of information needed to define digital test operations in manufacturing tests. STIL is not intended to define how the tester implements that information. While the purpose of STIL is to pass test data into the test environment, the overall STIL language is inherently more flexible than any particular tester. Constructs may be used in a STIL file that exceed the capability of a particular tester. In some circumstances, a translator for a particular type of test equipment may be capable of restructuring the data to support that capability on the tester; in other circumstances, separate tools may operate on that data to provide that restructuring. In all circumstances, it is desirable to provide the capability to check the data against the constraints of a tester. This capability is referred to as Tester Rules Checking and is the domain of tools that operate on STIL data. As such, Tester Rules Checking operations are outside the scope of this standard.

Figure 2 shows how STIL fits into the data flow between computer-aided engineering (CAE)/simulation and the test environment. In this figure, STIL is shown as both the input and output of “STIL Manipulation Tools.” STIL represents patterns as a series of cyclized waveforms that are executed sequentially. The waveform representation can be as simple as a “print-on-change” set of events, or a complex set of parameterized events. Hence, tools may be required to manipulate the data according to the requirements of a particular class of device, simulation, or tester. The output of that manipulation is still represented in STIL.

Another issue presented in Figure 2 is the need for data from the tester to be transmitted back to the CAE/simulation environment for the purpose of correlating simulation data to tester data. Although this is recognized as an important aspect of testing digital devices, it does not represent the data volume that the patterns themselves do, and is not specifically supported in this version of the standard.

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

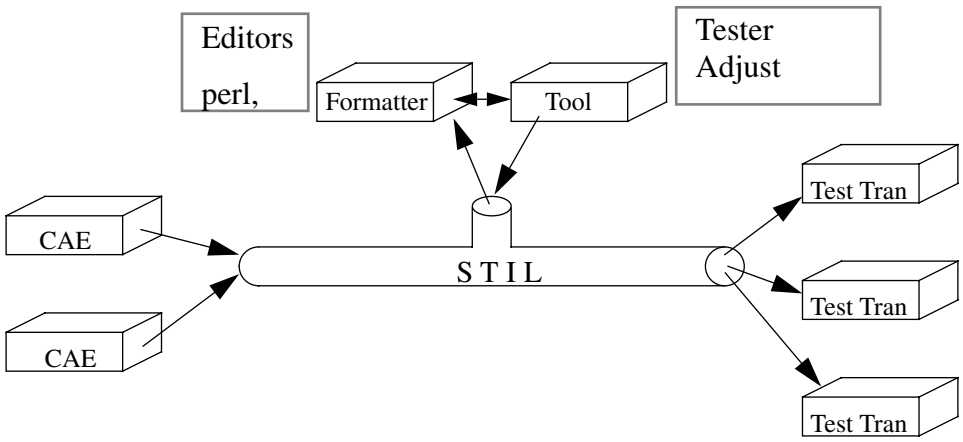


Figure 1—A conduit for transporting data from CAE to ATE

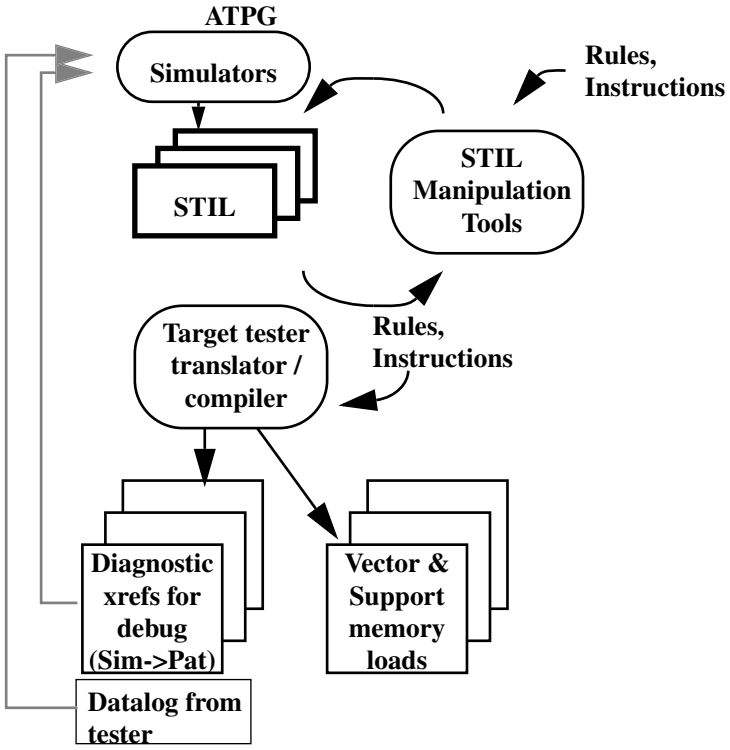


Figure 2—STIL usage model

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

1.1 Scope

This standard defines a test description language that:

a) Facilitates the transfer of large volumes of digital test vector data from CAE environments to automated test equipment (ATE) environments;

b) Specifies pattern, format, and timing information sufficient to define the application of digital test vectors to a device under test (DUT);

c) Supports the volume of test vector data generated from structured tests such as scan/automatic test pattern generation (ATPG), integral test techniques such as built-in self test (BIST), and functional test specifications for IC designs and their assemblies, in a format optimized for application in ATE environments.

In setting the scope for any standard, some issues are defined to not be pertinent to the initial project. The following is a partial list of issues that were dropped from the scope of this initial project:

— Levels: A key aspect of a digital test program is the ability to establish voltage and current parameters (levels) for signals under test. Level handling is not explicitly defined in the current standard, as this information is both compact (not presenting a transportation issue) and commonly established independently of digital test data, requiring different support mechanisms outside the current scope of this standard. Termination values may affect levels.

— Diagnostic/fault-tracing information: The goal of this standard is to optimally present data that needs to be moved onto ATE. While diagnostic data, fault identification data, and macro/design element correspondence data can fall into this category (and is often fairly large), this standard is also focused on integrated circuit and assemblies test, and most debug/failure analysis occurs separately from the ATE for these structures. Note that return of failure information (for off-ATE analysis) is also not part of the standard as currently defined.

— Datalogging mechanisms, formatting, and control usually are not defined as part of this current standard.

— Parametric tests are not defined as an integral part of this standard, except for optional pattern labels that identify potential locations for parametric tests, such as I_{DDQ} tests or alternating current (AC) timing tests.

— Program flow: Test sequencing and ordering are not defined as part of the current standard except as necessary to define collections of digital patterns meant to execute as a unit.

— Binning constructs are not part of the current standard.

— Analog or mixed-signal test: While this is an area of concern for many participants, at this point transfer of analog test data does not contribute to the same transportation issue seen with digital data.

— Algorithmic pattern constructs (such as sequences commonly used for memory test) are not currently defined as part of the standard.

— Parallel test/multisite test constructs are not an integral part of the current environment.

— User input and user control/options are not part of the current standard.

— Characterization tools, such as shmoo plots, are not defined as part of the current standard.

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.

1.2 Purpose

This standard addresses a need in the integrated circuit (IC)[1] test industry to define a standard mechanism for transferring the large volumes of digital test data from the generation environment through to test. The environment today contains unique output formats of existing CAE tools, individual test environments of IC manufacturers, and proprietary IC ATE input interfaces. As each of these three arenas solves individual problems, together they have created a morass of interfaces, translators, and software environments that provide no opportunity to leverage common goals and result in much wasted efforts re-engineering solutions. As device density increases, the magnitude of test data threatens to shift the test bottleneck from the generation process to the processes necessary solely to maintain and transport this data. These two factors threaten to eliminate any productive work performed in this area unless a viable standard is defined.

With a common standard for CAE and IC ATE environments, the generation, movement, and processing of this test data is greatly facilitated. This standard also allows for immediate access to test equipment supporting this standard, which benefits both ATE and IC vendors reviewing this equipment.

This standard also serves as a catalyst for the development of a set of standard third party interface tools to both test and design aspects of IC device generation.

2. References

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

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

IEEE Std 260.1-1993, American National Standard Letter Symbols for Units of Measurement (SI Units, Customary Inch-Pound Units, and Certain Other Units).

ISO 2955:1983, Information processing—Representation of SI and other units in systems with limited character sets.[3]

ISO/IEC 9899:1999, Programming languages—C.[4]

Published by IEC under licence from IEEE. © 1999 IEEE. All rights reserved.