

IEEE Standard for Test Access Port and Boundary-Scan Architecture

IEEE Computer Society

Sponsored by the
Test Technology Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1149.1™-2013
(Revision of
IEEE Std 1149.1-2001)

13 May 2013

Currently in preview, click buy full version

IEEE Std 1149.1™-2013

(Revision of
IEEE Std 1149.1-2001)

IEEE Standard for Test Access Port and Boundary-Scan Architecture

Sponsor

**Test Technology Standards Committee
of the
IEEE Computer Society**

Approved 6 February 2013

IEEE-SA Standards Board

Currently in preview, click buy full version

Abstract: Circuitry that may be built into an integrated circuit to assist in the test, maintenance and support of assembled printed circuit boards and the test of internal circuits is defined. The circuitry includes a standard interface through which instructions and test data are communicated. A set of test features is defined, including a boundary-scan register, such that the component is able to respond to a minimum set of instructions designed to assist with testing of assembled printed circuit boards. Also, a language is defined that allows rigorous structural description of the component-specific aspects of such testability features, and a second language is defined that allows rigorous procedural description of how the testability features may be used.

Keywords: boundary scan, boundary-scan architecture, Boundary-Scan Description Language (BSDL), boundary-scan register, circuit boards, circuitry, IEEE 1149.1™, integrated circuit, printed circuit boards, Procedural Description Language (PDL), test, test access port (TAP), very high speed integrated circuit (VHSIC), VHSIC Hardware Description Language (VHDL)

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2013 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 13 May 2013. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-0-7381-8263-6 STD98160
Print: ISBN 978-0-7381-8264-3 STDPD98160

IEEE prohibits discrimination, harassment and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.
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.

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. 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 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 or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. 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 any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, 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 in its standards 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 ten years. When a document is more than ten years old and has not undergone a revision process, 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 its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important to ensure that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. A person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at <http://standards.ieee.org/develop/wg/>.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854-4141
USA

Photocopies: 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.

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at <http://standards.ieee.org/index.html> or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit IEEE-SA Website at <http://standards.ieee.org/index.html>.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not 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. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the P1149.1 Working Group had the following membership:

C. J. Clark, *Chair*
Carol Pyron, *Vice-Chair*
Carl F. Barnhart, *Editor*
Bill Tuthill, *Secretary*

John Braden
Bill Bruce
Richard Cornejo
Adam Cron
Wim Driessen
David Dubberke
Ted Eaton
Heiko Ehrenberg

William Eklow
Peter Elias
Joshua Ferry
Jeff Halnon
Dharma Konda
Roland R. Latvala
Adam W. Ley
Sankaran Menon

Kent Ng
Kenneth P. Parker
Francisco Russi
John Seibold
Roger Sowada
Craig Stephan
Brian Turmell
Hugh Wallace

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Gobinathan Athimolom
Carl F. Barnhart
Hugh Barrass
William Borroz
John Braden
Dennis Brophy
Susan Burgess
Gunnar Carlsson
Vivek Chickermane
C. J. Clark
Richard Cornejo
Adam Cron
Alfred Crouch
Frans G De Jong
Jason Doege
Wim Driessen
David Dubberke
Sourav Dutta
Heiko Ehrenberg
William Eklow
Peter Elias
Joshua Ferry
Chris Goringe
Prashant Goteti
Robert Gottlieb
J. Grealish

Randall Groves
Jeff Halnon
Peter Harrod
Neil Glenn Jacobson
Rohit Kapur
Dharma Konda
Roland R. Latvala
Philippe LeBourg
Adam W. Ley
Teresa Lopes
Greg Luri
Wayne Manges
Colin Mander
Ian Mantos
Harrison Miles Jr.
Jeffrey Moore
Benoit Nadeau-Dostie
Ion Neag
Kenneth P. Parker
Steve Poehlman
Ulrich Pohl
Irith Pomeranz
John Potter
Carol Pyron
Mike Ricchetti
Gordon Robinson

Andrzej Rucinski
Francisco Russi
Bartien Sayogo
John Seibold
Ozgur Sinanoglu
Roger Sowada
Craig Stephan
Cees Stork
Walter Struppler
Stephen Sunter
Bambang Suparjo
Anthony Suto
Efren Taboada
David Thompson
Brian Turmelle
Bill Tuthill
Louis Ungar
Dmitri Varsanofiev
Srinivasa Vemuru
John Vergis
Tom Waayers
Douglas D. Way
Thomas Williams
Henk Wit
Oren Yuen
Janusz Zalewski

When the IEEE-SA Standards Board approved this standard on 6 February 2013, it had the following membership:

Richard H. Hulett, *Chair*
Robert Grow, *Past Chair*
Konstantinos Karachalios, *Secretary*

Satish Aggarwal
Masayuki Ariyoshi
Peter Balma
William Bartley
Ted Burse
Clint Chaplin
Wael William Diab
Jean-Philippe Faure

Alexander Gelman
Paul Houzé
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling

Oleg Logvinov
Ted Olsen
Gary Robinson
Jon Walter Rosdahl
Mike Seavey
Yatin Trivedi
Phil Winston
Yu Yuan

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Kathryn Bennett
IEEE Standards Program Manager, Technical Program Development

Introduction

This introduction is not part of IEEE Std 1149.1-2013, IEEE Standard for Test Access Port and Boundary-Scan Architecture.

This standard defines a test access port and boundary-scan architecture for digital integrated circuits and for the digital portions of mixed analog/digital integrated circuits. The facilities defined by the standard seek to provide a solution to the problem of testing assembled printed circuit boards and other products based on highly complex digital integrated circuits and high-density surface-mounting assembly techniques. They also provide a means of accessing and controlling design-for-test features built into the digital integrated circuits themselves. Such features might, for example, include internal scan paths and self-test functions as well as other features intended to support service applications in the assembled product.

In addition, two languages are provided to describe both the structure of the test logic and the procedures needed to use the test logic.

History of the development of this standard

The process of developing this standard began in 1985 when the Joint European Test Action Group (JETAG) was formed in Europe. During 1986, this group expanded to include members from both Europe and North America and, as a result, was renamed the Joint Test Action Group (JTAG). Between 1986 and 1988, the JTAG Technical Subcommittee developed and published a series of proposals for a standardized form of boundary scan. In 1988, the last of these proposals, JTAG Version 2.0, was offered to the IEEE Testability Bus Standards Committee (P1149) for inclusion in the standard then under development. The Testability Bus Standards Committee accepted this approach. It decided that the JTAG proposal should become the basis of a standard within the Testability Bus family, with the result that the P1149.1 project was initiated. Following these decisions, the JTAG Technical Subcommittee became the core of the IEEE Working Group that developed this standard.

After the initial approval of this standard in February 1990 and its subsequent publication, the Working Group immediately began efforts to develop a supplement for the purpose of correction, clarification, and enhancement. This effort, spurred and guided by interaction between developers and users of the original standard, culminated in IEEE Std 1149.1a™-1993, which was approved in June 1993.

The major changes to this standard introduced in IEEE Std 1149.1a-1993 were as follows:

- The addition of two optional instructions, *CLAMP* and *HIGHZ*, which standardized the names and specifications of features often implemented as design-specific features.
- The addition of an optional facility to switch a component from a mode in which it complies to this standard into one in which it supports another design-for-test approach.

Furthermore, starting with a proposal made by Kenneth P. Parker and Stig Oresjo in 1990, an effort was undertaken to develop a language to describe components that conform to this standard. This effort concluded in the approval of IEEE Std 1149.1b™-1994 in September 1994.

The major change introduced to this standard by IEEE Std 1149.1b-1994 was the addition of Annex B, which defines the Boundary-Scan Description Language. All other changes were minor and were strictly for clarification.

The 2003 revision was primarily a housekeeping update, designed to incorporate learning from the first 10 years of the standard's use into the standard document. The principal changes introduced were as follows:

- To reduce the risk of accidental entry into test mode, the requirement that a binary code for the *EXTEST* instruction be {000...0} was removed and use of this binary code for other instructions that result in entry to test mode was deprecated.

- To increase the flexibility with which instructions may be implemented and merged, the implicitly merged *SAMPLE/PRELOAD* instruction was redefined as two separate instructions: *SAMPLE* and *PRELOAD*. *These instructions can continue to share a single binary code*, effectively resulting in a merged *SAMPLE/PRELOAD* instruction, but alternatively, they may now share binary codes with other instructions, provided that no rules applying to any of the merged instructions are violated.
- To enable more efficient implementation of boundary-scan register cells provided at system logic outputs, the source of data to be captured in such cells in response to the *SAMPLE* instruction was allowed to be at the connected system pin. Additionally, three new cell types based on this implementation (**BC_8**, **BC_9**, and **BC_10**) were added to the standard Boundary-Scan Description Language (BSDL) Package and Package Body.
- To permit more flexible boundary-scan register cell implementations, sharing of circuitry between the boundary-scan register and other elements of the test and/or system logic were allowed in limited cases.
- To support more complete description of IC pin drivers with bus keeper circuits, a new value for <disable result> was defined (**KEEPER**).
- To track the widespread acceptance of BSDL, the language was made a normative part of this standard and its use for documentation was mandated.

Additionally, a number of minor changes were made to correct and clarify the language of this standard.

Changes introduced by this revision

First, this version of the standard affirms what had been required in the previous (2001) version. There are only minor clarifications or relaxations to the rules that are already established. It is expected that components currently compliant with the previous version of this standard will remain compliant with this one. The one exception is that the previously deprecated **BC_6** boundary-scan cell is no longer supported or defined, and the component supplier must provide a user-supplied BSDL package defining the **BC_6** cell for any component using the **STD_1149_1_2013** standard Package and still using that cell.

Second, while this is a major revision, items introduced in this version are optional and intended to provide test improvements for the complex components being created today and in the foreseeable future. There are also significant improvements in documentation capability, including the introduction of a new language to document test procedures unique to the component.

The major changes, listed in the order in which they appear in this standard, are as follows:

In the standard body:

- A new, optional, test mode persistence controller that can maintain the IEEE 1149.1 test logic in test mode even if the active instruction does not force test mode. Clause 6 is now split into 6.1 for the TAP controller and 6.2 for the test mode persistence controller. In support of this new controller, there are three new instructions: *CLAMP_HOLD* and *TMP_STATUS* in 8.20, with the new TMP status test data register in Clause 16; and *CLAMP_RELEASE* in 8.20.
- A new, optional *ECIDCODE* instruction in 8.15 and its electronic chip identification test data register in Clause 13 to supplement the existing *IDCODE* and *USERCODE* instructions and allow for the recovery of an Electronic Chip Identification value used to identify and track individual integrated circuits.
- A new, optional, component initialization mechanism to provide more flexibility in preparing the component for test. The *INIT_SETUP*, *INIT_SETUP_CLAMP*, and *INIT_RUN* instructions in 8.17, 8.18, and 8.19, and their new initialization data and initialization status test data registers in Clause 14 and Clause 15, respectively. This will allow programmable input/output (I/O) to be set up prior to board or system testing, as well as any tasks required to put the system logic into a safe state for test.
- A new, optional, *IC_RESET* instruction in 8.21 and its reset_select test data register in Clause 17 to provide control of component reset functions through the TAP.

- In 9.2, an optional standard TAP to test data register interface is recommended, and examples of different types of test data register cells using this interface are shown. In addition, the concept of register segments is expanded to allow for segments that may be excluded or included. This is introduced to support power domains that may be powered down, and yet may have a segment of a test data register within that domain. However, the capability was kept general.
- In the new 9.4, the rules for defining and controlling the new excludable and selectable segments are established.
- Boundary-scan register description in Clause 11 has been updated to support:
 - i) Optional excludable (but not selectable) boundary-scan register segments
 - ii) Optional observe-only boundary-scan register cells to redundantly capture the signal value on all digital pins except the TAP pins
 - iii) Optional observe-only boundary-scan register cells to capture a fault condition on all pins, including nondigital pins, except the TAP pins
- Documentation requirements in Clause 18 have been updated for the new capabilities.

Note that where rules were removed or moved in this version of this standard, a placeholder was left behind (“Removed in this version” or “Moved to Permission”) in order to preserve the rule numbering from previous versions. This is intended to simplify the transition for both users and tool vendors in supporting what is a significant change.

In Annex B (Boundary Scan Description Language):

- The entire annex was rewritten for:
 - i) Increased clarity of what was normative versus descriptive text
 - ii) Increased consistency in presentation
- BSDL is no longer a “proper subset” of VHDL, but it is now “based on” VHDL. See B.4. In particular, new pin type keywords were introduced in B.8.3 that are not needed in VHDL but give a more accurate description of each port in BSDL.
- Formal definitions of language elements are included in B.5 instead of reliance on inheritance from VHDL.
- Some changes to the BNF notation used, including definition of all the special character tokens, are in B.6.
- Pin mapping in B.8.7 now allows for documenting that a port is not connected to any device package pin in a specific mapped device package.
- The boundary-scan register description in B.8.14 introduces new attributes for defining boundary-scan register segments, and introduces a requirement for documenting the behavior of an undriven input.
- New capabilities are introduced for documenting the structural details of test data registers:
 - i) Subclause B.8.18 introduces the definition of mnemonics that may be associated with register fields.
 - ii) Subclause B.8.19 introduces the ability to name fields within a register or segment.
 - iii) Subclause B.8.20 introduces the ability to define the types of cells used in a test data register (TDR) field.
 - iv) Subclause B.8.21 introduces the ability to hierarchically assemble segments into larger segments or whole registers.
 - v) Subclause B.8.22 introduces the ability to define constraints on the values to be loaded in a register or register field.
 - vi) Subclause B.8.23 introduces the ability to associate a register field or bit to specific ports and other information, and to associate a power port to other ports.
- The role of a User Defined Package defined in B.10 has been expanded to support logic IP providers who may need to document test data register segments contained within their IP.

A new annex, Annex C, codifies the Procedural Description Language (PDL), a new language for documenting the procedural and data requirements for some of the new instructions. As mentioned, this version of the standard introduces new instructions for configuring complex I/Os prior to entering the *EXTTEST* instruction. As the data required for initialization could vary for each use of the component on each distinct board or system design, this

created the need for a new language for setting internal TDR register fields in order to configure the I/O. It was decided to adopt PDL and tailor it to the BSDL register descriptions and IEEE 1149.1 needs.

A new informative annex, Annex D, shows extended examples of BSDL and PDL used together to describe the structure and the procedures for use of new capabilities.

A new informative annex, Annex E, shows example pseudo-code for the execution of the PDL iApply command, the most complex of the new commands in PDL.

Contents

1. Overview	1
1.1 Scope	1
1.2 Purpose	1
1.3 Document outline	5
1.4 Text conventions.....	6
1.5 Logic diagram conventions.....	6
2. Normative references.....	7
3. Definitions, abbreviations, acronyms, and special terms.....	8
3.1 Definitions	8
3.2 Abbreviations and acronyms	11
3.3 Special terms.....	12
4. Test access port (TAP)	13
4.1 Connections that form the TAP	13
4.2 Test clock input (TCK).....	13
4.3 Test mode select (TMS) input	14
4.4 Test data input (TDI).....	15
4.5 Test data output (TDO).....	15
4.6 Test reset input (TRST*)	16
4.7 Interconnection of components compatible with this standard.....	17
4.8 Subordination of this standard within a higher level test strategy	20
5. Test logic architecture	22
5.1 Test logic design.....	22
5.2 Test logic realization	23
6. Test logic controllers	24
6.1 TAP controller	24
6.2 Test mode persistence (TMP) controller.....	39
7. Instruction register.....	46
7.1 Design and construction of the instruction register	46
7.2 Instruction register operation.....	47
8. Instructions	50
8.1 Response of the test logic to instructions.....	50
8.2 Public instructions	51
8.3 Private instructions	53
8.4 <i>BYPASS</i> instruction.....	53
8.5 Boundary-scan register instructions.....	54
8.6 <i>SAMPLE</i> instruction	57
8.7 <i>PRELOAD</i> instruction	58
8.8 <i>EXTEST</i> instruction	60
8.9 <i>INTEST</i> instruction	62
8.10 <i>RUNBIST</i> instruction	66
8.11 <i>CLAMP</i> instruction.....	68
8.12 Device identification register instructions	70
8.13 <i>IDCODE</i> instruction.....	70
8.14 <i>USERCODE</i> instruction.....	71
8.15 <i>ECIDCODE</i> instruction.....	73
8.16 <i>HIGHZ</i> instruction.....	74

8.17 Component initialization instructions and procedures.....	76
8.18 <i>INIT_SETUP</i> and <i>INIT_SETUP_CLAMP</i> instructions.....	81
8.19 <i>INIT_RUN</i> instruction.....	82
8.20 <i>CLAMP_HOLD</i> , <i>CLAMP_RELEASE</i> , and <i>TMP_STATUS</i> instructions.....	84
8.21 <i>IC_RESET</i> instruction.....	88
9. Test data registers.....	91
9.1 Provision of test data registers.....	91
9.2 Design and construction of test data registers.....	94
9.3 Operation of test data registers.....	106
9.4 Design and control of test data register segments.....	108
10. Bypass register.....	116
10.1 Design and operation of the bypass register.....	116
11. Boundary-scan register.....	118
11.1 Introduction.....	118
11.2 Register design.....	122
11.3 Register operation.....	124
11.4 General rules regarding cell provision.....	126
11.5 Provision and operation of cells at system logic inputs.....	131
11.6 Provision and operation of cells at system logic outputs.....	138
11.7 Provision and operation of cells at bidirectional system logic pins.....	154
11.8 Redundant cells.....	161
11.9 Special cases.....	163
12. Device identification register.....	166
12.1 Design and operation of the device identification register.....	166
12.2 Manufacturer identity code.....	168
12.3 Part-number code.....	169
12.4 Version code.....	170
13. Electronic chip identification (ECID) register.....	171
13.1 Design and operation of the ECID register.....	171
14. Initialization data register.....	172
14.1 Design and operation of the initialization data register.....	172
15. Initialization status register.....	175
15.1 Design and operation of the initialization status register.....	175
16. TMP status register.....	176
16.1 Design and operation of the TMP status register.....	176
17. Reset selection register.....	178
17.1 Design and operation of the reset selection register.....	178
18. Conformance and documentation requirements.....	183
18.1 Claiming conformance to this standard.....	183
18.2 Prime and second source components.....	184
18.3 Documentation requirements.....	184
Annex A (informative) Example implementation using level-sensitive design techniques.....	188

Annex B (normative) Boundary Scan Description Language (BSDL).....	189
B.1 General information.....	189
B.1.1 Document outline.....	189
B.1.2 Conventions.....	189
B.1.3 BSDL history.....	189
B.2 Purpose of BSDL.....	190
B.3 Scope of BSDL.....	190
B.4 Relationship of BSDL to VHDL.....	191
B.4.1 Specifications.....	191
B.5 Lexical elements of BSDL.....	192
B.5.1 Character set.....	192
B.5.2 BSDL reserved words.....	193
B.5.3 VHDL reserved and predefined words.....	194
B.5.4 Identifiers.....	195
B.5.5 Numeric literals.....	196
B.5.6 Strings.....	197
B.5.7 Information tag.....	198
B.5.8 Comments.....	199
B.6 Syntax definition.....	199
B.6.1 BNF conventions.....	199
B.6.2 Commonly used syntactic elements.....	200
B.7 Components of a BSDL description.....	202
B.7.1 Specifications.....	202
B.7.2 Description.....	203
B.8 Entity description.....	203
B.8.1 Overall syntax of the entity description.....	203
B.8.2 Generic parameter statement.....	204
B.8.3 Logical port description statement.....	205
B.8.4 Standard use statement.....	208
B.8.5 Use statement.....	211
B.8.6 Component conformance statement.....	212
B.8.7 Device package pin mappings.....	213
B.8.8 Grouped port identification.....	216
B.8.9 Scan port identification.....	219
B.8.10 Compliance-enable description.....	220
B.8.11 Instruction register description.....	221
B.8.12 Optional device register description.....	224
B.8.13 Register access description.....	227
B.8.14 Boundary-scan register description.....	229
B.8.15 RUNBIST description.....	245
B.8.16 INTEST description.....	247
B.8.17 System clock requirements attribute.....	249
B.8.18 Register mnemonics description.....	250
B.8.19 Register fields description.....	254
B.8.20 Register field assignment description.....	261
B.8.21 Register assembly description.....	271
B.8.22 Register constraint description.....	288
B.8.23 Register and power port association attributes.....	291
B.8.24 User extensions to BSDL.....	295
B.8.25 Design warning.....	297
B.9 Standard BSDL Package STD_1149_1_2013.....	298
B.10 User-supplied BSDL packages.....	302
B.10.1 Specifications.....	302
B.10.2 Description.....	306
B.10.3 Examples.....	307

B.11	BSDL example applications	308
B.11.1	Typical application of BSDL	308
B.11.2	Boundary-scan register description	311
B.12	1990 version of BSDL	315
B.12.1	1990 Standard VHDL Package STD_1149_1_1990	316
B.12.2	Typical application of BSDL, 1990 version	319
B.12.3	Obsolete syntax	320
B.12.4	Miscellaneous points on 1990 version	321
B.13	1994 version of BSDL	321
B.13.1	Standard VHDL Package STD_1149_1_1994	321
B.14	2001 version of BSDL	325
B.14.1	Standard VHDL Package STD_1149_1_2001	325
Annex C (normative) Procedural Description Language (PDL)		329
C.1	General information	329
C.1.1	Purpose	329
C.1.2	Dependence on Tool Command Language (Tcl)	330
C.1.3	Dependence on Boundary Scan Description Language (BSDL)	330
C.2	PDL concepts and use model	330
C.2.1	Use model introduction	330
C.2.2	PDL levels	332
C.2.3	PDL procedures	333
C.2.4	Read and write with capture-shift-update sequence	334
C.2.5	Register state definition	334
C.2.6	Level-0 PDL commands	336
C.2.7	Specification of names and values	339
C.2.8	Retargeting	340
C.2.9	Simple PDL Example	341
C.3	PDL Level 0 command reference	343
C.3.1	Understanding a PDL “string”	343
C.3.2	BNF conventions	344
C.3.3	PDL lexical elements and common syntax	345
C.3.4	PDL File	350
C.3.5	Procedure definition commands	351
C.3.6	Test setup commands	356
C.3.7	Test execution commands	360
C.3.8	Flow-control commands	367
C.3.9	Optimization commands	374
C.3.10	Miscellaneous commands	378
C.3.11	Low-level commands	379
C.4	PDL Level 1 command reference	382
C.4.1	Level-1 PDL operation	383
C.4.2	iGet command	383
C.4.3	iGetStatus command	388
C.5	Example BSDL and PDL for the use model	388
C.5.1	BSDL Packages for IP	389
C.5.2	BSDL files for components	390
C.5.3	PDL files supplied by IP supplier	393
C.5.4	PDL files supplied by component supplier	394
C.5.5	PDL files coded by test engineer	395
Annex D (informative) Integrated examples of BSDL and PDL		398
D.1	Initialization example structure and procedures	398
D.1.1	Initialization example using register description attributes	398
D.1.2	Example PDL for INIT example	405

D.2 Multiple wrapper serial port structure and procedures	408
D.2.1 Wrapper serial port structural description.....	408
D.2.2 Wrapper serial port example.....	417
Annex E (informative) Example iApply execution flow	420

Figures

Figure 1-1—Boundary-scan register cell.....	3
Figure 1-2—Boundary-scannable board design	4
Figure 1-3—Logic symbology used in this standard.....	6
Figure 4-1—Serial connection using one TMS signal.....	18
Figure 7-2—Instruction register with decoder between shift and update stages.....	49
Figure 8-1—Simplified view of the boundary-scan register	55
Figure 8-2—Example boundary-scan register cell design.....	56
Figure 8-3—Figure used to illustrate boundary-scan instructions.....	56
Figure 8-4—Data flow for the <i>SAMPLE</i> instruction	58
Figure 8-5—Data flow for the <i>PRELOAD</i> instruction.....	60
Figure 8-6—Data flow for the <i>EXTEST</i> instruction	62
Figure 8-7—Data flow for the <i>INTEST</i> instruction	64
Figure 8-8—Control of applied system clock during <i>INTEST</i>	65
Figure 8-9—Use of TCK as clock for on-chip system logic during <i>INTEST</i>	65
Figure 8-10—Use of the <i>HIGHZ</i> instruction.....	75
Figure 8-11—Provision of <i>HIGHZ</i> at a two-state pin.....	76
Figure 8-12—Boundary-scan register control cell with a reset on the update flip-flop R2.....	87
Figure 9-1—Implementation of the group of test data registers	92
Figure 9-2—Construction of multiple test data registers from shared circuitry	96
Figure 9-3—Gated-clock boundary-scan register gating.....	97
Figure 9-4—Test data register control gating.....	97
Figure 9-5—Capture-update TDR cell using gated clocks.....	98
Figure 9-6—Capture-update TDR cell with nongated clock and optional reset.....	100
Figure 9-7—Update TDR cell without capture and with nongated clock and optional reset	101
Figure 9-8—Capture TDR cell with nongated clock and without update stage	102
Figure 9-9—Shift-only TDR cell with nongated clock and without update stage.....	102
Figure 9-10—Self-monitoring TDR cell with update stage and nongated clocks.....	104
Figure 9-11—Self-resetting and self-monitoring TDR cell with nongated clocks.....	105
Figure 9-12—Timing of a self-resetting and self-monitoring TDR cell at <i>Update-DR</i>	106
Figure 9-13—Example design containing two optional test data registers.....	107
Figure 9-14—Scan control of excludable test data register segments	112
Figure 9-15—Scan control of excludable test data register segments with domain control	113
Figure 9-16—Domain POR reset of nested segment-select fields.....	114
Figure 9-17—Hierarchical reset of nested segment-select fields	114
Figure 9-18—Selectable segments and selection field	115
Figure 9-19—Example segment-select or selection-field cell with ungated clocks	115
Figure 10-1—Bypass register gated-clock implementation.....	116
Figure 11-1—Component without boundary scan.....	119
Figure 11-2—Input connections.....	120
Figure 11-3—Connection of an observe-only boundary-scan register cell	121
Figure 11-4—Insertion of a control-and-observe boundary-scan register cell	121
Figure 11-5—Conceptual view of a control-and-observe boundary-scan register cell.....	122
Figure 11-6—Boundary-scan shift-register design.....	124
Figure 11-7—Component that contains analog circuitry.....	126
Figure 11-8—Placement of boundary-scan register cells.....	128
Figure 11-9—Component with differential inputs and outputs	129
Figure 11-10—Conceptual schematic of redundant observe-only cells on differential pins	130
Figure 11-11—Provision of a boundary-scan register cell at a system input	131
Figure 11-12—Provision of multiple boundary-scan register cells at one input	132
Figure 11-13—Noninversion of data between pin and TDO.....	134

Figure 11-14—Noninversion of data between TDI and the system logic.....	134
Figure 11-15—Input cell with parallel output register [BC_2]	136
Figure 11-16—Input cell without parallel output register [BC_3]	136
Figure 11-17—Cell that forces the system logic input to 1 during <i>EXTEST</i> [BC_4]	137
Figure 11-18—Observe-only input cell without control [BC_4].....	137
Figure 11-19—Input cell that supports all instructions [BC_1].....	138
Figure 11-20—Provision of a boundary-scan register cell at a digital system output pin	139
Figure 11-21—Provision of boundary-scan register cells at system logic outputs.....	140
Figure 11-22—Provision of cells when one output is used both as control and data	140
Figure 11-23—Noninversion of data between the system logic and TDO	142
Figure 11-24—Noninversion of data between TDI and a system output pin	143
Figure 11-25—Noninversion of control signal values between the system logic and TDO.....	144
Figure 11-26—Control of multiple three-state outputs from one signal.....	145
Figure 11-27—Testing board-level bus lines	146
Figure 11-28—Testing external logic via the boundary-scan register.....	147
Figure 11-29—Primitive noncompliant output cell design with potential problems.....	147
Figure 11-30—Circuit illustrating potential boundary-scan test problem	148
Figure 11-31—Output cell that supports all instructions [BC_1].....	149
Figure 11-32—Output cell that does not support <i>INTEST</i> [BC_2].....	149
Figure 11-33—Self-monitoring output cell that supports <i>INTEST</i> [BC_9]	151
Figure 11-34—Self-monitoring output cell that does not support <i>INTEST</i> [BC_10]	151
Figure 11-35—Boundary-scan register cells at a three-state output—Example 1 [BC_1, control and data]	152
Figure 11-36—Boundary-scan register cells at a three-state output—Example 2 [BC_2, control and data]	153
Figure 11-37—Boundary-scan register cells at a bidirectional pin—Example 1 [BC_1, control]	155
Figure 11-38—Boundary-scan register cells at a bidirectional pin—Example 2 [BC_2 control; BC_7 data]	156
Figure 11-39—Deprecated boundary-scan register cells at a bidirectional pin [BC_2 control; BC_6 data]	158
Figure 11-40—Boundary-scan register cells at an open-collector bidirectional pin [BC_4, input; BC_2, output] ..	159
Figure 11-41—Boundary-scan register cell at an open-collector bidirectional pin [BC_8]	160
Figure 11-42—Boundary-scan register cells for use at a bidirectional pin where <i>INTEST</i> is not provided [BC_2, control; BC_8, data].....	161
Figure 11-43—Cell that should not be included in the boundary-scan register.....	163
Figure 11-44—Input pins used only to control output pins—Case A	164
Figure 11-45—Input pins used only to control output pins—Case B.....	164
Figure 11-46—Noncompliant use of a single cell for output control and data.....	165
Figure 11-47—Boundary-scan register cells at a three-state pin where output control is from a system pin [BC_5, control; BC_1, data].....	165
Figure 12-1—Structure of the device identification code.....	166
Figure 12-2—Device identification register gated-clock cell design.....	167
Figure 16-1—Example TMP status register (nongated clocks).....	177
Figure 17-1—Reset selection register overview.....	178
Figure 17-2—Minimal reset selection register example.....	181
Figure 18-1—Measuring setup and hold timing.....	187
Figure 18-2—Measuring propagation delay.....	187
Figure B-1—Components of a BSDL description.....	202
Figure B-2—Example use of nonboundary-scan port-types.....	208
Figure B-3—Example of unconnected pin types.....	216
Figure B-4—Cell design corresponding to Figure 11-19 and Figure 11-31	230
Figure B-5—Symbolic representation of a boundary-scan register cell.....	230
Figure B-6—Symbolic representation of a boundary-scan register cell without an update stage	231
Figure B-7—Cell on an input, which pulls to a logic 1	243
Figure B-8—Illustration of use of <input spec> for an IC	243
Figure B-9—Illustration of use of fault detection boundary cells for an IC	244
Figure B-10—Simple local reset structure	270
Figure B-11—Simple selectable register segment structure.....	280

Figure B-12—Illustrative component power control structure	283
Figure B-13—Simple wrapper serial port	285
Figure B-14—Three wrappers in parallel	287
Figure B-15—Boundary-scan register cell showing possible capture sources	306
Figure B-16—Texas Instruments SN74BCT8374	309
Figure B-17—Component that illustrates several OBSERVE_ONLY and INTERNAL cells	312
Figure B-18—Component that illustrates several merged cells	314
Figure C-1—PDL example board	331
Figure C-2—PDL example detail	332
Figure C-3—PDL scan frame	335
Figure C-4—Data flow during an iApply command	336
Figure C-5—iMerge example	375
Figure C-6—Example circuit board	389
Figure D-1—Hard SerDes IP defined in a package	399
Figure D-2—Simple wrapper serial port	408
Figure D-3—Three wrappers with WSC gating logic	411
Figure D-4—WSP example for interconnect testing	412
Figure D-5—Three wrappers with WSC gating logic and SEGSEL	415

Tables

Table 6-1—Use of controller states for different test types.....	29
Table 6-2—Test logic operation in each controller state.....	33
Table 6-3—State assignments for example TAP controller.....	36
Table 7-1—Instruction register operation in each controller state.....	47
Table 8-1—Typical initialization sequence, deferred test mode.....	79
Table 8-2—Including boundary-scan segments in mission mode.....	80
Table 8-3—Typical initialization sequence, immediate test mode.....	80
Table 8-4—Including boundary-scan segments in test mode.....	81
Table 8-5—I/O pin behavior for TMP controller states.....	86
Table 9-1—Recommended TDR interface for design specific TDRs.....	95
Table 9-2—Naming of test data registers that share circuitry.....	96
Table 11-1—Routing of signals in cells at system logic inputs.....	132
Table 11-2—Mode signal generation for the example cells in Figure 11-15 and Figure 11-16.....	136
Table 11-3—Mode signal generation for the example cell in Figure 11-19.....	138
Table 11-4—Routing of signals in cells at system logic outputs.....	142
Table 11-5—Test for driver B.....	146
Table 11-6—Mode signal generation for the example cells in Figure 11-31, Figure 11-35, Figure 11-37, and Figure 11-47.....	149
Table 11-7—Mode signal generation for the example cells in Figure 11-32, Figure 11-34, and Figure 11-40.....	150
Table 11-8—Mode signal generation for the example cell in Figure 11-33.....	151
Table 11-9—Mode signal generation for the example cell in Figure 11-36.....	154
Table 11-10—Mode signal generation for the example cells in Figure 11-38.....	157
Table 11-11—Mode signal generation for the deprecated example cells in Figure 11-39.....	159
Table 11-12—Mode signal generation for the example cells in Figure 11-41 and Figure 11-42.....	160
Table 17-1—Logic hazards of dual transitions of reset-enable and reset-control pairs.....	182
Table 18-1—Public instructions.....	183
Table B-1—Scope of BSDL.....	191
Table B-2—Pin types.....	206
Table B-3—List of cells defined in the Standard BSDL Package and relevant figure numbers.....	229
Table B-4—Function element values and meanings.....	238
Table B-5—Constraint expression operators.....	290
Table B-6—Unit value definitions.....	293
Table B-7—Cell context element values and meanings.....	303
Table B-8—Data source element values and meanings.....	304
Table B-9—Compliant capture sources for <cell context> of INPUT, CLOCK, and BIDIR_IN.....	304
Table B-10—Compliant capture sources for <cell context> of OUTPUT2, OUTPUT3, and BIDIR_OUT.....	304
Table B-11—Compliant capture sources for <cell context> of CONTROL and CONTROLR.....	304
Table B-12—Compliant capture sources for <cell context> of INTERNAL.....	305
Table B-13—Compliant capture sources for <cell context> of OBSERVE_ONLY.....	305
Table C-1—PDL Level-0 commands.....	337
Table C-2—Handling PDL procedure hierarchy.....	341
Table C-3—PDL Level-1 commands.....	383

IEEE Standard for Test Access Port and Boundary-Scan Architecture

IMPORTANT NOTICE: IEEE standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

1. Overview

1.1 Scope

This standard defines test logic that can be included in an integrated circuit to provide standardized approaches to:

- Testing the interconnections between integrated circuits once they have been assembled onto a printed circuit board or other substrate
- Testing the integrated circuit itself
- Observing or modifying circuit activity during the component’s normal operation

The test logic consists of a boundary-scan register and other building blocks and is accessed through a test access port (TAP).

1.2 Purpose

1.2.1 Overview of the operation of this standard

This subclause provides a general overview of the operation of a component compatible with this standard and provides a background to the detailed discussion in later clauses.

The circuitry defined by this standard allows test instructions (which take control of the component outputs and observe the component inputs) and associated test data to be fed into a component and, subsequently, allows the results of execution of such instructions to be read out. All information (instructions, test data, and test results) is communicated in a serial format.

The sequence of operations would be controlled by a bus master, which could be either an automatic test equipment (ATE) or a component that interfaces to a higher level test bus as a part of a complete system maintenance architecture. Control is achieved through signals applied to the test mode select (TMS) and test clock (TCK) inputs of the various components connected to the bus master. Starting from an initial state in which the test circuitry defined by this standard is inactive, a typical sequence of operations would be as follows.