



IEC 62530

Edition 2.0 2011-05

# INTERNATIONAL STANDARD

IEEE Std 1800™



SystemVerilog – iTeh STANDARD PREVIEW  
Unified Hardware Design, Specification, and Verification Language  
(standards.iteh.ai)

[IEC 62530:2011](#)

[https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-4f1d-a83a-  
caa9634a3161/iec-62530-2011](https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-4f1d-a83a-caa9634a3161/iec-62530-2011)



**THIS PUBLICATION IS COPYRIGHT PROTECTED**  
**Copyright © 2009 IEEE**

All rights reserved. IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc.

Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the IEC Central Office.

Any questions about IEEE copyright should be addressed to the IEEE. Enquiries about obtaining additional rights to this publication and other information requests should be addressed to the IEC or your local IEC member National Committee.

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland  
Email: [inmail@iec.ch](mailto:inmail@iec.ch)  
Web: [www.iec.ch](http://www.iec.ch)

The Institute of Electrical and Electronics Engineers, Inc  
3 Park Avenue  
US-New York, NY10016-5997  
USA  
Email: [stds-info@ieee.org](mailto:stds-info@ieee.org)  
Web: [www.ieee.org](http://www.ieee.org)

### About the IEC

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

### About IEC publications

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

- Catalogue of IEC publications: [www.iec.ch/searchpub](http://www.iec.ch/searchpub)

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: [www.iec.ch/online\\_news/justpub](http://www.iec.ch/online_news/justpub)

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

- Electropedia: [www.electropedia.org](http://www.electropedia.org)

The world's leading online dictionary of electronic and electrical terms containing more than 20.000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: [www.iec.ch/webstore/custserv](http://www.iec.ch/webstore/custserv)

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: [csc@iec.ch](mailto:csc@iec.ch)

Tel.: +41 22 919 02 11



IEC 62530

Edition 2.0 2011-05

# INTERNATIONAL STANDARD

IEEE Std 1800™



---

**SystemVerilog – iTeh STANDARD PREVIEW**  
**Unified Hardware Design, Specification, and Verification Language**  
(standards.itih.ai)

[IEC 62530:2011](https://standards.itih.ai/catalog/standards/sist/e0c60393-16c7-4f1d-a83a-caa9634a3161/iec-62530-2011)

[https://standards.itih.ai/catalog/standards/sist/e0c60393-16c7-4f1d-a83a-  
caa9634a3161/iec-62530-2011](https://standards.itih.ai/catalog/standards/sist/e0c60393-16c7-4f1d-a83a-caa9634a3161/iec-62530-2011)

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

PRICE CODE **XX**

---

ICS 25.040

ISBN 978-2-88912-450-3

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

[IEC 62530:2011](#)

[https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-4f1d-a83a-  
caa9634a3161/iec-62530-2011](https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-4f1d-a83a-<br/>caa9634a3161/iec-62530-2011)

# Contents

## Part One:

### Design and Verification Constructs

1. Overview.....	2
1.1 Scope.....	2
1.2 Purpose.....	2
1.3 Merger of IEEE Std 1364-2005 and IEEE Std 1800-2005.....	3
1.4 Special terms.....	3
1.5 Conventions used in this standard.....	3
1.6 Syntactic description.....	4
1.7 Use of color in this standard.....	5
1.8 Contents of this standard.....	5
1.9 Deprecated clauses.....	8
1.10 Examples.....	8
1.11 Prerequisites.....	8
2. Normative references.....	9
3. Design and verification building blocks.....	11
3.1 General.....	11
3.2 Design elements.....	11
3.3 Modules.....	11
3.4 Programs.....	12
3.5 Interfaces.....	13
3.6 Checkers.....	14
3.7 Primitives.....	14
3.8 Subroutines.....	14
3.9 Packages.....	14
3.10 Configurations.....	15
3.11 Overview of hierarchy.....	15
3.12 Compilation and elaboration.....	16
3.13 Name spaces.....	18
3.14 Simulation time units and precision.....	19
4. Scheduling semantics.....	23
4.1 General.....	23
4.2 Execution of a hardware model and its verification environment.....	23
4.3 Event simulation.....	23
4.4 The stratified event scheduler.....	24
4.5 The SystemVerilog simulation reference algorithm.....	29
4.6 Determinism.....	29
4.7 Nondeterminism.....	30
4.8 Race conditions.....	30
4.9 Scheduling implication of assignments.....	30
4.10 The PLI callback control points.....	32
5. Lexical conventions.....	33
5.1 General.....	33
5.2 Lexical tokens.....	33
5.3 White space.....	33
5.4 Comments.....	33
5.5 Operators.....	33
5.6 Identifiers, keywords, and system names.....	34
5.7 Numbers.....	35
5.8 Time literals.....	40

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

[IEC 62530:2011](https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-41fd-a83a-<br/>caa9634a3161/iec-62530-2011)

[https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-41fd-a83a-  
caa9634a3161/iec-62530-2011](https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-41fd-a83a-<br/>caa9634a3161/iec-62530-2011)

5.9	String literals .....	40
5.10	Structure literals .....	42
5.11	Array literals .....	43
5.12	Attributes .....	43
5.13	Built-in methods .....	45
6.	Data types .....	47
6.1	General .....	47
6.2	Data types and data objects .....	47
6.3	Value set .....	47
6.4	Singular and aggregate types .....	48
6.5	Nets and variables .....	49
6.6	Net types .....	50
6.7	Net declarations .....	56
6.8	Variable declarations .....	58
6.9	Vector declarations .....	60
6.10	Implicit declarations .....	61
6.11	Integer data types .....	62
6.12	Real, shortreal and realtime data types .....	63
6.13	Void data type .....	63
6.14	Handle data type .....	63
6.15	Class .....	64
6.16	String data type .....	64
6.17	Event data type .....	69
6.18	User-defined types .....	70
6.19	Enumerations .....	71
6.20	Constants .....	77
6.21	Scope and lifetime .....	84
6.22	Type compatibility .....	86
6.23	Type operator .....	89
6.24	Casting .....	90
7.	Aggregate data types .....	97
7.1	General .....	97
7.2	Structures .....	97
7.3	Unions .....	99
7.4	Packed and unpacked arrays .....	102
7.5	Dynamic arrays .....	106
7.6	Array assignments .....	109
7.7	Arrays as arguments to subroutines .....	110
7.8	Associative arrays .....	111
7.9	Associative array methods .....	114
7.10	Queues .....	117
7.11	Array querying functions .....	121
7.12	Array manipulation methods .....	121
8.	Classes .....	127
8.1	General .....	127
8.2	Overview .....	127
8.3	Syntax .....	128
8.4	Objects (class instance) .....	129
8.5	Object properties and object parameter data .....	130
8.6	Object methods .....	130
8.7	Constructors .....	131
8.8	Static class properties .....	132
8.9	Static methods .....	133

ITeCh STANDARD PREVIEW  
(standards.iteh.ai)

[IEC 62530:2011](https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-41fd-a83a-<br/>caa9634a3161/iec-62530-2011)

[https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-41fd-a83a-  
caa9634a3161/iec-62530-2011](https://standards.iteh.ai/catalog/standards/sist/e0c60393-16c7-41fd-a83a-<br/>caa9634a3161/iec-62530-2011)

8.10	This .....	133
8.11	Assignment, renaming, and copying .....	134
8.12	Inheritance and subclasses .....	135
8.13	Overridden members .....	136
8.14	Super .....	137
8.15	Casting .....	137
8.16	Chaining constructors .....	138
8.17	Data hiding and encapsulation .....	138
8.18	Constant class properties .....	139
8.19	Virtual methods .....	140
8.20	Abstract classes and pure virtual methods .....	141
8.21	Polymorphism: dynamic method lookup .....	141
8.22	Class scope resolution operator :: .....	142
8.23	Out-of-block declarations .....	144
8.24	Parameterized classes .....	145
8.25	Typedef class .....	148
8.26	Classes and structures .....	149
8.27	Memory management .....	149
9.	Processes .....	151
9.1	General .....	151
9.2	Structured procedures .....	151
9.3	Block statements .....	155
9.4	Procedural timing controls .....	161
9.5	Process execution threads .....	170
9.6	Process control .....	171
9.7	Fine-grain process control .....	175
10.	Assignment statements .....	177
10.1	General .....	177
10.2	Overview .....	177
10.3	Continuous assignments .....	178
10.4	Procedural assignments .....	181
10.5	Variable declaration assignment (variable initialization) .....	186
10.6	Procedural continuous assignments .....	186
10.7	Assignment extension and truncation .....	188
10.8	Assignment-like contexts .....	189
10.9	Assignment patterns .....	190
10.10	Unpacked array concatenation .....	194
10.11	Net aliasing .....	197
11.	Operators and expressions .....	199
11.1	General .....	199
11.2	Overview .....	199
11.3	Operators .....	200
11.4	Operator descriptions .....	204
11.5	Operands .....	224
11.6	Expression bit lengths .....	227
11.7	Signed expressions .....	230
11.8	Expression evaluation rules .....	231
11.9	Tagged union expressions and member access .....	232
11.10	String literal expressions .....	234
11.11	Operator overloading .....	235
11.12	Minimum, typical, and maximum delay expressions .....	237
11.13	Let construct .....	238
12.	Procedural programming statements .....	245

ITIH STANDARD PREVIEW  
(standards.iteh.ai)

12.1	General	245
12.2	Overview	245
12.3	Syntax	245
12.4	Conditional if–else statement	246
12.5	Case statement	251
12.6	Pattern matching conditional statements	256
12.7	Loop statements	260
12.8	Jump statements	264
13.	Tasks and functions (subroutines)	267
13.1	General	267
13.2	Overview	267
13.3	Tasks	267
13.4	Functions	271
13.5	Subroutine calls and argument passing	277
13.6	Import and export functions	282
13.7	Task and function names	282
14.	Clocking blocks	283
14.1	General	283
14.2	Overview	283
14.3	Clocking block declaration	283
14.4	Input and output skews	285
14.5	Hierarchical expressions	286
14.6	Signals in multiple clocking blocks	287
14.7	Clocking block scope and lifetime	287
14.8	Multiple clocking blocks example	287
14.9	Interfaces and clocking blocks	288
14.10	Clocking block events	289
14.11	Cycle delay: ##	289
14.12	Default clocking	290
14.13	Input sampling	291
14.14	Global clocking	292
14.15	Synchronous events	293
14.16	Synchronous drives	293
15.	Interprocess synchronization and communication	299
15.1	General	299
15.2	Overview	299
15.3	Semaphores	299
15.4	Mailboxes	301
15.5	Named events	304
16.	Assertions	309
16.1	General	309
16.2	Overview	309
16.3	Immediate assertions	309
16.4	Deferred assertions	312
16.5	Concurrent assertions overview	316
16.6	Boolean expressions	318
16.7	Sequences	320
16.8	Declaring sequences	323
16.9	Sequence operations	331
16.10	Local variables	353
16.11	Calling subroutines on match of a sequence	359
16.12	System functions	360
16.13	Declaring properties	360



16.14 Multiclock support .....	385
16.15 Concurrent assertions .....	393
16.16 Disable iff resolution .....	410
16.17 Clock resolution .....	412
16.18 Expect statement .....	417
16.19 Clocking blocks and concurrent assertions .....	419
17. Checkers .....	421
17.1 Overview .....	421
17.2 Checker declaration .....	421
17.3 Checker instantiation .....	424
17.4 Context inference .....	427
17.5 Checker procedures .....	427
17.6 Covergroups in checkers .....	428
17.7 Checker variables .....	429
17.8 Functions in checkers .....	435
17.9 Complex checker example .....	435
18. Constrained random value generation .....	437
18.1 General .....	437
18.2 Overview .....	437
18.3 Concepts and usage .....	437
18.4 Random variables .....	440
18.5 Constraint blocks .....	442
18.6 Randomization methods .....	457
18.7 In-line constraints—randomize() with .....	459
18.8 Disabling random variables with rand_mode() .....	461
18.9 Controlling constraints with constrain_mode() .....	463
18.10 Dynamic constraint modification .....	464
18.11 In-line random variable control .....	464
18.12 Randomization of scope variables—std::randomize() .....	465
18.13 Random number system functions and methods .....	467
18.14 Random stability .....	468
18.15 Manually seeding randomize .....	471
18.16 Random weighted case—randcase .....	471
18.17 Random sequence generation—randsequence .....	472
19. Functional coverage .....	483
19.1 General .....	483
19.2 Overview .....	483
19.3 Defining the coverage model: covergroup .....	484
19.4 Using covergroup in classes .....	486
19.5 Defining coverage points .....	488
19.6 Defining cross coverage .....	498
19.7 Specifying coverage options .....	503
19.8 Predefined coverage methods .....	507
19.9 Predefined coverage system tasks and system functions .....	509
19.10 Organization of option and type_option members .....	509
19.11 Coverage computation .....	510
20. Utility system tasks and system functions .....	515
20.1 General .....	515
20.2 Simulation control system tasks .....	516
20.3 Simulation time system functions .....	516
20.4 Timescale system tasks .....	518
20.5 Conversion functions .....	521
20.6 Data query functions .....	522

20.7	Array querying functions .....	524
20.8	Math functions .....	526
20.9	Severity tasks .....	528
20.10	Elaboration system tasks .....	528
20.11	Assertion control system tasks .....	530
20.12	Assertion action control system tasks .....	531
20.13	Assertion system functions .....	533
20.14	Coverage system functions .....	534
20.15	Probabilistic distribution functions .....	534
20.16	Stochastic analysis tasks and functions .....	536
20.17	Programmable logic array (PLA) modeling system tasks .....	538
20.18	Miscellaneous tasks and functions .....	542
21.	I/O system tasks and system functions .....	543
21.1	General .....	543
21.2	Display system tasks .....	543
21.3	File input-output system tasks and system functions .....	554
21.4	Loading memory array data from a file .....	565
21.5	Writing memory array data to a file .....	568
21.6	Command line input .....	569
21.7	Value change dump (VCD) files .....	572
22.	Compiler directives .....	593
22.1	General .....	593
22.2	Overview .....	593
22.3	<code>`resetall</code> .....	593
22.4	<code>`include</code> .....	594
22.5	<code>`define</code> , <code>`undef</code> and <code>`undefineall</code> .....	594
22.6	<code>`ifdef</code> , <code>`else</code> , <code>`elsif</code> , <code>`endif</code> , <code>`ifndef</code> .....	600
22.7	<code>`timescale</code> .....	603
22.8	<code>`default_nettype</code> .....	604
22.9	<code>`unconnected_drive</code> and <code>`nounconnected_drive</code> .....	605
22.10	<code>`celldefine</code> and <code>`endcelldefine</code> .....	605
22.11	<code>`pragma</code> .....	605
22.12	<code>`line</code> .....	606
22.13	<code>`__FILE__</code> and <code>`__LINE__</code> .....	607
22.14	<code>`begin_keywords</code> , <code>`end_keywords</code> .....	608

**Part Two:**

**Hierarchy Constructs**

23.	Modules and hierarchy .....	614
23.1	General .....	614
23.2	Module definitions .....	614
23.3	Module instances (hierarchy) .....	626
23.4	Nested modules .....	636
23.5	Extern modules .....	637
23.6	Hierarchical names .....	638
23.7	Member selects and hierarchical names .....	641
23.8	Upwards name referencing .....	642
23.9	Scope rules .....	644
23.10	Overriding module parameters .....	646
23.11	Binding auxiliary code to scopes or instances .....	654
24.	Programs .....	659
24.1	General .....	659
24.2	Overview .....	659

24.3	The program construct .....	659
24.4	Eliminating testbench races .....	663
24.5	Blocking tasks in cycle/event mode.....	663
24.6	Programwide space and anonymous programs.....	664
24.7	Program control tasks .....	664
25.	Interfaces.....	665
25.1	General.....	665
25.2	Overview.....	665
25.3	Interface syntax.....	666
25.4	Ports in interfaces.....	670
25.5	Modports.....	671
25.6	Interfaces and specify blocks.....	677
25.7	Tasks and functions in interfaces.....	678
25.8	Parameterized interfaces.....	684
25.9	Virtual interfaces.....	686
25.10	Access to interface objects.....	691
26.	Packages.....	693
26.1	General.....	693
26.2	Package declarations.....	693
26.3	Referencing data in packages.....	694
26.4	Using packages in module headers.....	698
26.5	Search order rules.....	699
26.6	Exporting imported names from packages.....	701
26.7	The std built-in package.....	702
27.	Generate constructs.....	705
27.1	General.....	705
27.2	Overview.....	705
27.3	Generate construct syntax.....	705
27.4	Loop generate constructs.....	707
27.5	Conditional generate constructs.....	711
27.6	External names for unnamed generate blocks.....	714
28.	Gate-level and switch-level modeling.....	717
28.1	General.....	717
28.2	Overview.....	717
28.3	Gate and switch declaration syntax.....	717
28.4	and, nand, nor, or, xor, and xnor gates.....	723
28.5	buf and not gates.....	724
28.6	bufif1, bufif0, notif1, and notif0 gates.....	725
28.7	MOS switches.....	726
28.8	Bidirectional pass switches.....	727
28.9	CMOS switches.....	728
28.10	pullup and pulldown sources.....	729
28.11	Logic strength modeling.....	729
28.12	Strengths and values of combined signals.....	731
28.13	Strength reduction by nonresistive devices.....	744
28.14	Strength reduction by resistive devices.....	744
28.15	Strengths of net types.....	744
28.16	Gate and net delays.....	745
29.	User defined primitives (UDPs).....	749
29.1	General.....	749
29.2	Overview.....	749
29.3	UDP definition.....	749
29.4	Combinational UDPs.....	753

ITeH STANDARD PREVIEW  
(standards.iteh.ai)

29.5	Level-sensitive sequential UDPs .....	754
29.6	Edge-sensitive sequential UDPs .....	754
29.7	Sequential UDP initialization .....	755
29.8	UDP instances.....	757
29.9	Mixing level-sensitive and edge-sensitive descriptions.....	758
29.10	Level-sensitive dominance.....	759
30.	Specify blocks.....	761
30.1	General.....	761
30.2	Overview.....	761
30.3	Specify block declaration.....	761
30.4	Module path declarations.....	762
30.5	Assigning delays to module paths.....	771
30.6	Mixing module path delays and distributed delays.....	775
30.7	Detailed control of pulse filtering behavior.....	776
31.	Timing checks.....	785
31.1	General.....	785
31.2	Overview.....	785
31.3	Timing checks using a stability window.....	788
31.4	Timing checks for clock and control signals.....	795
31.5	Edge-control specifiers.....	804
31.6	Notifiers: user-defined responses to timing violations.....	805
31.7	Enabling timing checks with conditioned events.....	807
31.8	Vector signals in timing checks.....	808
31.9	Negative timing checks.....	809
32.	Backannotation using the standard delay format (SDF).....	815
32.1	General.....	815
32.2	Overview.....	815
32.3	The SDF annotator.....	815
32.4	Mapping of SDF constructs to SystemVerilog.....	815
32.5	Multiple annotations.....	820
32.6	Multiple SDF files.....	821
32.7	Pulse limit annotation.....	821
32.8	SDF to SystemVerilog delay value mapping.....	822
32.9	Loading timing data from an SDF file.....	822
33.	Configuring the contents of a design.....	825
33.1	General.....	825
33.2	Overview.....	825
33.3	Libraries.....	826
33.4	Configurations.....	828
33.5	Using libraries and configs.....	834
33.6	Configuration examples.....	835
33.7	Displaying library binding information.....	837
33.8	Library mapping examples.....	837
34.	Protected envelopes.....	841
34.1	General.....	841
34.2	Overview.....	841
34.3	Processing protected envelopes.....	841
34.4	Protect pragma directives.....	843
34.5	Protect pragma keywords.....	845

**Part Three:**  
**Application Programming Interfaces**

35. Direct programming interface (DPI).....	862
35.1 General.....	862
35.2 Overview.....	862
35.3 Two layers of the DPI.....	863
35.4 Global name space of imported and exported functions.....	864
35.5 Imported tasks and functions.....	865
35.6 Calling imported functions.....	872
35.7 Exported functions.....	874
35.8 Exported tasks.....	875
35.9 Disabling DPI tasks and functions.....	875
36. Programming language interface (PLI/VPI) overview.....	877
36.1 General.....	877
36.2 PLI purpose and history.....	877
36.3 User-defined system task and system function names.....	878
36.4 User-defined system task and system function arguments.....	879
36.5 User-defined system task and system function types.....	879
36.6 User-supplied PLI applications.....	879
36.7 PLI include files.....	879
36.8 VPI sizetf, compiletf and calltf routines.....	879
36.9 PLI mechanism.....	880
36.10 VPI access to SystemVerilog objects and simulation objects.....	882
36.11 List of VPI routines by functional category.....	883
36.12 VPI backwards compatibility features and limitations.....	885
37. VPI object model diagrams.....	891
37.1 General.....	891
37.2 VPI Handles.....	891
37.3 VPI object classifications.....	892
37.4 Key to data model diagrams.....	898
37.5 Module.....	901
37.6 Interface.....	902
37.7 Modport.....	902
37.8 Interface task or function declaration.....	902
37.9 Program.....	903
37.10 Instance.....	904
37.11 Instance arrays.....	906
37.12 Scope.....	907
37.13 IO declaration.....	908
37.14 Ports.....	909
37.15 Reference objects.....	910
37.16 Nets.....	913
37.17 Variables.....	917
37.18 Packed array variables.....	920
37.19 Variable select.....	921
37.20 Memory.....	922
37.21 Variable drivers and loads.....	922
37.22 Object Range.....	923
37.23 Typespec.....	924
37.24 Structures and unions.....	926
37.25 Named events.....	927
37.26 Parameter, spec param, def param, param assign.....	928
37.27 Class definition.....	929

37.28 Class typespec .....	930
37.29 Class variables and class objects.....	932
37.30 Constraint, constraint ordering, distribution .....	934
37.31 Primitive, prim term.....	935
37.32 UDP.....	936
37.33 Intermodule path .....	936
37.34 Constraint expression .....	937
37.35 Module path, path term .....	937
37.36 Timing check .....	938
37.37 Task and function declaration .....	939
37.38 Task and function call .....	940
37.39 Frames .....	942
37.40 Threads .....	943
37.41 Delay terminals .....	943
37.42 Net drivers and loads .....	944
37.43 Continuous assignment .....	945
37.44 Clocking block .....	946
37.45 Assertion .....	947
37.46 Concurrent assertions .....	948
37.47 Property declaration .....	949
37.48 Property specification .....	950
37.49 Sequence declaration .....	951
37.50 Sequence expression .....	952
37.51 Immediate assertions .....	953
37.52 Multiclock sequence expression .....	954
37.53 Let .....	954
37.54 Simple expressions .....	955
37.55 Expressions .....	956
37.56 Atomic statement .....	959
37.57 Event statement .....	960
37.58 Process .....	960
37.59 Assignment .....	961
37.60 Event control .....	961
37.61 While, repeat .....	962
37.62 Waits .....	962
37.63 Delay control.....	962
37.64 Repeat control .....	963
37.65 Forever .....	963
37.66 If, if-else .....	963
37.67 Case, pattern .....	964
37.68 Expect .....	965
37.69 For .....	965
37.70 Do-while, foreach .....	965
37.71 Alias statement .....	966
37.72 Disables.....	967
37.73 Return statement .....	967
37.74 Assign statement, deassign, force, release.....	967
37.75 Callback .....	968
37.76 Time queue.....	968
37.77 Active time format.....	969
37.78 Attribute .....	970
37.79 Iterator.....	971
37.80 Generates .....	972
38. VPI routine definitions.....	975

38.1	General	975
38.2	vpi_chk_error()	975
38.3	vpi_compare_objects()	976
38.4	vpi_control()	978
38.5	vpi_flush()	979
38.6	vpi_get()	979
38.7	vpi_get64()	980
38.8	vpi_get_cb_info()	980
38.9	vpi_get_data()	981
38.10	vpi_get_delays()	982
38.11	vpi_get_str()	984
38.12	vpi_get_systf_info()	985
38.13	vpi_get_time()	986
38.14	vpi_get_userdata()	987
38.15	vpi_get_value()	987
38.16	vpi_get_value_array()	993
38.17	vpi_get_vlog_info()	997
38.18	vpi_handle()	998
38.19	vpi_handle_by_index()	999
38.20	vpi_handle_by_multi_index()	999
38.21	vpi_handle_by_name()	1000
38.22	vpi_handle_multi()	1001
38.23	vpi_iterate()	1001
38.24	vpi_mcd_close()	1002
38.25	vpi_mcd_flush()	1003
38.26	vpi_mcd_name()	1003
38.27	vpi_mcd_open()	1004
38.28	vpi_mcd_printf()	1005
38.29	vpi_mcd_vprintf()	1006
38.30	vpi_printf()	1006
38.31	vpi_put_data()	1007
38.32	vpi_put_delays()	1009
38.33	vpi_put_userdata()	1012
38.34	vpi_put_value()	1012
38.35	vpi_put_value_array()	1015
38.36	vpi_register_cb()	1019
38.37	vpi_register_systf()	1027
38.38	vpi_release_handle()	1030
38.39	vpi_remove_cb()	1031
38.40	vpi_scan()	1031
38.41	vpi_vprintf()	1032
39.	Assertion API	1033
39.1	General	1033
39.2	Overview	1033
39.3	Static information	1033
39.4	Dynamic information	1034
39.5	Control functions	1038
40.	Code coverage control and API	1043
40.1	General	1043
40.2	Overview	1043
40.3	SystemVerilog real-time coverage access	1044
40.4	FSM recognition	1049
40.5	VPI coverage extensions	1051
41.	Data read API	1057