IEC 62530

Edition 2.0 2011-05

INTERNATIONAL STANDARD

IEEE Std 1800™

colour inside

SystemVerilog –
Unified Hardware Design, Specification, and Verification Language

IEC 62530:2011(E) IEEE Std 1800-2009

Documents
Document Sharing Hub

Full document details: 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 Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: 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
Web: 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
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 It also gives information on projects, withdrawn and replaced publications.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
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
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
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
Tel.: +41 22 919 02 11

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

IEC 62530

Edition 2.0 2011-05

INTERNATIONAL STANDARD

IEEE Std 1800™

colour inside

SystemVerilog –
Unified Hardware Design, Specification, and Verification Language

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XX

ICS 25.040

ISBN 978-2-88912-450-3

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Documents
Document Sharing Hub

Full document details: 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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: 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 Chandle 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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: 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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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 constraint_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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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 `resetall ... 593
22.4 `include ... 594
22.5 `define, `undef and `undefineall ... 594
22.6 `ifdef, `else, `elsif, `endif, `ifndef ... 600
22.7 `timescale ... 603
22.8 `default_nettype ... 604
22.9 `unconnected_drive and `nounconnected_drive ... 605
22.10 `celldefine and `endcelldefine ... 605
22.11 `pragma ... 605
22.12 `line ... 606
22.13 `__FILE__ and `__LINE__ ... 607
22.14 `begin_keywords, `end_keywords ... 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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Part Four:
Annexes

Annex A (normative) Formal syntax 1060
A.1 Source text 1060
A.2 Declarations 1068
A.3 Primitive instances 1079
A.4 Instantiations 1081
A.5 UDP declaration and instantiation 1082
A.6 Behavioral statements 1084
A.7 Specify section 1090
A.8 Expressions 1094
A.9 General 1100
A.10 Footnotes (normative) 1102
Annex B (normative) Keywords 1105
Annex C (normative) Deprecation 1107
C.1 General 1107
C.2 Constructs that have been deprecated 1107
C.3 Accellera SystemVerilog 3.1a-compatible access to packed data 1108
C.4 Constructs identified for deprecation 1108
Annex D (informative) Optional system tasks and system functions 1111
D.1 General 1111
D.2 $countdrivers 1111
D.3 $getpattern 1112
D.4 $input 1113
D.5 $key and $nokey 1113
D.6 $list 1113
D.7 $log and $nolog 1114
D.8 $reset, $reset_count, and $reset_value 1114
D.9 $save, $restart, and $incsave 1115
D.10 $scale 1116
D.11 $scope 1116
D.12 $showscopes 1116
D.13 $showvars 1116
D.14 $sreadmemb and $sreadmemh 1117
Annex E (informative) Optional compiler directives 1119
E.1 General 1119
E.2 `default_decay_time 1119
E.3 `default_trireg_strength 1119
E.4 `delay_mode_distributed 1120
E.5 `delay_mode_path 1120
E.6 `delay_mode_unit 1120
E.7 `delay_mode_zero 1120
Annex F (normative) Formal semantics of concurrent assertions 1121
F.1 General 1121
F.2 Overview 1121
F.3 Abstract syntax 1122
F.4 Rewriting algorithms 1128
F.5 Semantics 1132
F.6 Extended expressions 1141
F.7 Recursive properties 1141
Annex G (normative) Std package 1145
G.1 General 1145
G.2 Overview 1145

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

G.3 Semaphore ... 1145
G.4 Mailbox ... 1145
G.5 Randomize ... 1146
G.6 Process ... 1146
Annex H (normative) DPI C layer ... 1147
H.1 General ... 1147
H.2 Overview ... 1147
H.3 Naming conventions ... 1148
H.4 Portability ... 1148
H.5 svdpi.h include file ... 1148
H.6 Semantic constraints ... 1149
H.7 Data types ... 1152
H.8 Argument passing modes ... 1155
H.9 Context tasks and functions ... 1158
H.10 Include files ... 1163
H.11 Arrays ... 1166
H.12 Open arrays ... 1168
H.13 SV3.1a-compatible access to packed data (deprecated functionality) ... 1174
Annex I (normative) svdpi.h ... 1181
I.1 General ... 1181
I.2 Overview ... 1181
I.3 Source code ... 1181
Annex J (normative) Inclusion of foreign language code ... 1191
J.1 General ... 1191
J.2 Overview ... 1191
J.3 Location independence ... 1192
J.4 Object code inclusion ... 1192
Annex K (normative) vpi_user.h ... 1195
K.1 General ... 1195
K.2 Source code ... 1195
Annex L (normative) vpi_compatibility.h ... 1213
L.1 General ... 1213
L.2 Source code ... 1213
Annex M (normative) sv_vpi_user.h ... 1217
M.1 General ... 1217
M.2 Source code ... 1217
Annex N (normative) Algorithm for probabilistic distribution functions ... 1227
N.1 General ... 1227
N.2 Source code ... 1227
Annex O (informative) Encryption/decryption flow ... 1235
O.1 General ... 1235
O.2 Overview ... 1235
O.3 Tool vendor secret key encryption system ... 1235
O.4 IP author secret key encryption system ... 1236
O.5 Digital envelopes ... 1237
Annex P (informative) Glossary ... 1239
Annex Q (informative) Mapping of IEEE Std 1364-2005 and IEEE Std 1800-2005 clauses into IEEE Std 1800-2009 ... 1243
Annex R (informative) Bibliography ... 1247
Annex S (informative) IEEE List of Participants ... 1249

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

List of figures

Figure 4-1—Event scheduling regions ... 28
Figure 6-1—Simulation values of a trireg and its driver ... 53
Figure 6-2—Simulation results of a capacitive network ... 54
Figure 6-3—Simulation results of charge sharing ... 55
Figure 7-1—VInt type with packed qualifier ... 102
Figure 7-2—Instr type with packed qualifier ... 102
Figure 9-1—Intra-assignment repeat event control utilizing a clock edge ... 170
Figure 14-1—Sample and drive times including skew with respect to the positive edge of the clock ... 286
Figure 16-1—Sampling a variable in a simulation time step ... 317
Figure 16-2—Concatenation of sequences ... 322
Figure 16-3—Value change expressions ... 337
Figure 16-4—Future value change ... 341
Figure 16-5—ANDing (and) two sequences ... 343
Figure 16-6—ANDing (and) two sequences, including a time range ... 344
Figure 16-7—ANDing (and) two Boolean expressions ... 344
Figure 16-8—Intersecting two sequences ... 345
Figure 16-9—ORing (or) two Boolean expressions ... 346
Figure 16-10—ORing (or) two sequences ... 347
Figure 16-11—ORing (or) two sequences, including a time range ... 348
Figure 16-12—Match with throughout restriction fails ... 350
Figure 16-13—Match with throughout restriction succeeds ... 351
Figure 16-14—Conditional sequence matching ... 367
Figure 16-15—Conditional sequences ... 368
Figure 16-16—Results without the condition ... 368
Figure 16-17—Clocking blocks and concurrent assertion ... 419
Figure 17-1—Non-deterministic free checker variable ... 430
Figure 18-1—Example of randc ... 442
Figure 18-2—Global constraints ... 451
Figure 18-3—Truth tables for conjunction, disjunction, and negation rules ... 455
Figure 21-1—Creating the 4-state VCD file ... 572
Figure 21-2—Creating the extended VCD file ... 582
Figure 23-1—Scopes available to upward name referencing ... 646
Figure 28-1—Schematic diagram of interconnections in array of instances ... 723
Figure 28-2—Scale of strengths ... 731
Figure 28-3—Combining unequal strengths ... 731
Figure 28-4—Combination of signals of equal strength and opposite values ... 732
Figure 28-5—Weak x signal strength ... 732
Figure 28-6—Bufifs with control inputs of x ... 733
Figure 28-7—Strong H range of values ... 733

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Figure 28-8—Strong L range of values ... 733
Figure 28-9—Combined signals of ambiguous strength ... 734
Figure 28-10—Range of strengths for an unknown signal ... 734
Figure 28-11—Ambiguous strengths from switch networks ... 735
Figure 28-12—Range of two strengths of a defined value ... 735
Figure 28-13—Range of three strengths of a defined value ... 735
Figure 28-14—Unknown value with a range of strengths ... 736
Figure 28-15—Strong X range ... 736
Figure 28-16—Ambiguous strength from gates ... 736
Figure 28-17—Ambiguous strength signal from a gate ... 737
Figure 28-18—Weak 0 ... 737
Figure 28-19—Ambiguous strength in combined gate signals ... 737
Figure 28-20—Elimination of strength levels ... 738
Figure 28-21—Result showing a range and the elimination of strength levels of two values ... 739
Figure 28-22—Result showing a range and the elimination of strength levels of one value ... 740
Figure 28-23—A range of both values ... 741
Figure 28-24—Wired logic with unambiguous strength signals ... 741
Figure 28-25—Wired logic and ambiguous strengths ... 743
Figure 28-26—Trireg net with capacitance ... 748
Figure 29-1—Module schematic and simulation times of initial value propagation ... 757
Figure 30-1—Module path delays ... 763
Figure 30-2—Difference between parallel and full connection paths ... 769
Figure 30-3—Module path delays longer than distributed delays ... 776
Figure 30-4—Module path delays shorter than distributed delays ... 776
Figure 30-5—Example of pulse filtering ... 777
Figure 30-6—On-detect versus on-event ... 779
Figure 30-7—Current event cancellation problem and correction ... 781
Figure 30-8—NAND gate with nearly simultaneous input switching where one event is scheduled prior to another that has not matured ... 782
Figure 30-9—NAND gate with nearly simultaneous input switching with output event scheduled at same time ... 783
Figure 31-1—Sample $timeskew ... 797
Figure 31-2—Sample $timeskew with remain_active_flag set ... 798
Figure 31-3—Sample $fullskew ... 800
Figure 31-4—Data constraint interval, positive setup/hold ... 809
Figure 31-5—Data constraint interval, negative setup/hold ... 810
Figure 31-6—Timing check violation windows ... 813
Figure 37-1—Example of object relationships diagram ... 893
Figure 37-2—Accessing a class of objects using tags ... 894
Figure 38-1—s_vpi_error_info structure definition ... 976
Figure 38-2—s_cb_data structure definition ... 981

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Figure 38-3—s_vpi_delay structure definition 982
Figure 38-4—s_vpi_time structure definition 982
Figure 38-5—s_vpi_systf_data structure definition 985
Figure 38-6—s_vpi_time structure definition 986
Figure 38-7—s_vpi_value structure definition 988
Figure 38-8—s_vpi_vecval structure definition 988
Figure 38-9—s_vpi_strengthval structure definition 988
Figure 38-10—s_vpi_vlog_info structure definition 997
Figure 38-11—s_vpi_delay structure definition 1010
Figure 38-12—s_vpi_time structure definition 1010
Figure 38-13—s_vpi_value structure definition 1014
Figure 38-14—s_vpi_time structure definition 1014
Figure 38-15—s_vpi_vecval structure definition 1015
Figure 38-16—s_vpi_strengthval structure definition 1015
Figure 38-17—s_cb_data structure definition 1019
Figure 38-18—s_vpi_systf_data structure definition 1027
Figure 39-1—Assertions with global clocking future sampled value functions 1038
Figure 40-1—Hierarchical instance example 1046
Figure 40-2—FSM specified with pragmas 1051

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

List of tables

Table 3-1—Time unit strings ... 19
Table 4-1—PLI callbacks ... 32
Table 5-1—Specifying special characters in string literals ... 41
Table 6-1—Net types ... 50
Table 6-2—Truth table for wire and tri nets ... 51
Table 6-3—Truth table for wand and triand nets ... 52
Table 6-4—Truth table for wor and trior nets ... 52
Table 6-5—Truth table for tri0 net ... 56
Table 6-6—Truth table for tri1 net ... 56
Table 6-7—Default values ... 60
Table 6-8—Integer data types ... 62
Table 6-9—String operators ... 66
Table 6-10—Enumeration element ranges ... 73
Table 6-11—Differences between specparams and parameters ... 83
Table 7-1—Value read from a nonexistent associative array entry ... 114
Table 8-1—Comparison of pointer and handle types ... 129
Table 9-1—fork-join control options ... 157
Table 9-2—Detecting posedge and negedge ... 163
Table 9-3—Intra-assignment timing control equivalence ... 169
Table 10-1—Legal left-hand forms in assignment statements ... 177
Table 11-1—Operators and data types ... 201
Table 11-2—Operator precedence and associativity ... 202
Table 11-3—Arithmetic operators defined ... 205
Table 11-4—Power operator rules ... 206
Table 11-5—Unary operators defined ... 206
Table 11-6—Examples of modulus and power operators ... 206
Table 11-7—Data type interpretation by arithmetic operators ... 207
Table 11-8—Definitions of relational operators ... 208
Table 11-9—Definitions of equality operators ... 209
Table 11-10—Wildcard equality and wildcard inequality operators ... 209
Table 11-11—Bitwise binary and operator ... 211
Table 11-12—Bitwise binary or operator ... 211
Table 11-13—Bitwise binary exclusive or operator ... 212
Table 11-14—Bitwise binary exclusive nor operator ... 212
Table 11-15—Bitwise unary negation operator ... 212
Table 11-16—Reduction unary and operator ... 213
Table 11-17—Reduction unary or operator ... 213
Table 11-18—Reduction unary exclusive or operator ... 213
Table 11-19—Results of unary reduction operations ... 214

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Table 11-20—Ambiguous condition results for conditional operator ... 215
Table 11-21—Bit lengths resulting from self-determined expressions ... 228
Table 16-1—Operator precedence and associativity ... 331
Table 16-2—Global clocking future sampled value functions ... 341
Table 16-3—Sequence and property operator precedence and associativity ... 363
Table 18-1—rand_mode argument ... 462
Table 18-2—constraint_mode argument ... 463
Table 19-1—Instance-specific coverage options ... 503
Table 19-2—Coverage options per-syntactic level ... 505
Table 19-3—Coverage group type (static) options ... 505
Table 19-4—Coverage type options ... 507
Table 19-5—Predefined coverage methods ... 507
Table 20-1—Diagnostics for $finish ... 516
Table 20-2—$timeformat units_number arguments ... 519
Table 20-3—$timeformat default value for arguments ... 520
Table 20-4—SystemVerilog to C real math function cross-listing ... 527
Table 20-5—VPI callbacks for assertion control tasks ... 531
Table 20-6—VPI callbacks for assertion action control tasks ... 532
Table 20-7—Types of queues of $q_type values ... 536
Table 20-8—Argument values for $q_exam system task ... 537
Table 20-9—Status code values ... 537
Table 20-10—PLA modeling system tasks ... 538
Table 21-1—Escape sequences for printing special characters ... 544
Table 21-2—Escape sequences for format specifications ... 545
Table 21-3—Format specifications for real numbers ... 547
Table 21-4—Logic value component of strength format ... 550
Table 21-5—Mnemonics for strength levels ... 550
Table 21-6—Explanation of strength formats ... 551
Table 21-7—Types for file descriptors ... 555
Table 21-8—$fscanf input field characters ... 559
Table 21-9—Rules for left-extending vector values ... 578
Table 21-10—How the VCD can shorten values ... 578
Table 21-11—Keyword commands ... 579
Table 21-12—VCD type mapping ... 591
Table 22-1—IEEE Std 1364-1995 reserved keywords ... 610
Table 22-2—IEEE Std 1364-2001 additional reserved keywords ... 611
Table 22-3—IEEE Std 1364-2005 additional reserved keywords ... 611
Table 22-4—IEEE Std 1800-2005 additional reserved keywords ... 612
Table 22-5—IEEE Std 1800-2009 additional reserved keywords ... 612
Table 23-1—Net types resulting from dissimilar port connections ... 635

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Table 26-1—Scoping rules for package importation ... 700
Table 28-1—Built-in gates and switches ... 718
Table 28-2—Valid gate types for strength specifications ... 719
Table 28-3—Truth tables for multiple input logic gates ... 724
Table 28-4—Truth tables for multiple output logic gates ... 725
Table 28-5—Truth tables for three-state logic gates ... 726
Table 28-6—Truth tables for MOS switches ... 727
Table 28-7—Strength levels for scalar net signal values ... 730
Table 28-8—Strength reduction rules ... 744
Table 28-9—Rules for propagation delays ... 745
Table 29-1—UDP table symbols ... 752
Table 29-2—Initial statements in UDPs and modules ... 755
Table 29-3—Mixing of level-sensitive and edge-sensitive entries ... 759
Table 30-1—List of valid operators in state-dependent path delay expression ... 766
Table 30-2—Associating path delay expressions with transitions ... 773
Table 30-3—Calculating delays for x transitions ... 774
Table 31-1—$setup arguments ... 788
Table 31-2—$hold arguments ... 789
Table 31-3—$setuphold arguments ... 790
Table 31-4—$removal arguments ... 792
Table 31-5—$recovery arguments ... 793
Table 31-6—$recrem arguments ... 794
Table 31-7—$skew arguments ... 796
Table 31-8—$timeskew arguments ... 797
Table 31-9—$fullskew arguments ... 799
Table 31-10—$width arguments ... 801
Table 31-11—$period arguments ... 802
Table 31-12—$nochange arguments ... 803
Table 31-13—Notifier value responses to timing violations ... 805
Table 32-1—Mapping of SDF delay constructs to SystemVerilog declarations ... 816
Table 32-2—Mapping of SDF timing check constructs to SystemVerilog ... 817
Table 32-3—SDF annotation of interconnect delays ... 819
Table 32-4—SDF to SystemVerilog delay value mapping ... 822
Table 32-5—mtm_spec argument ... 823
Table 32-6—scale_type argument ... 824
Table 34-1—protect pragma keywords ... 844
Table 34-2—Encoding algorithm identifiers ... 848
Table 34-3—Encryption algorithm identifiers ... 850
Table 34-4—Message digest algorithm identifiers ... 855
Table 36-1—VPI routines for simulation-related callbacks ... 883

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Table 36-2—VPI routines for system task or system function callbacks ... 884
Table 36-3—VPI routines for traversing SystemVerilog hierarchy ... 884
Table 36-4—VPI routines for accessing properties of objects ... 884
Table 36-5—VPI routines for accessing objects from properties ... 884
Table 36-6—VPI routines for delay processing ... 884
Table 36-7—VPI routines for logic and strength value processing ... 884
Table 36-8—VPI routines for simulation time processing ... 885
Table 36-9—VPI routines for miscellaneous utilities ... 885
Table 36-10—Summary of VPI incompatibilities across standard versions ... 886
Table 37-1—Part-select parent expressions ... 958
Table 38-1—Return error constants for vpi_chk_error() ... 976
Table 38-2—Size of the s_vpi_delay->da array ... 983
Table 38-3—Return value field of the s_vpi_value structure union ... 989
Table 38-4—Size of the s_vpi_delay->da array ... 1010
Table 38-5—Value format field of cb_data_p->value->format ... 1021
Table 38-6—cbStmt callbacks ... 1023
Table 40-1—Coverage control return values ... 1045
Table 40-2—Instance coverage permutations ... 1046
Table 40-3—Assertion coverage results ... 1053
Table B.1—Reserved keywords ... 1105
Table D.1—Argument return value for $countdriver function ... 1112
Table H.1—Mapping data types ... 1153
Table N.1—SystemVerilog to C function cross-listing ... 1227
Table Q.1—Mapping of LRM clauses ... 1243

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

List of syntax excerpts

Syntax 5-1—Syntax for system tasks and system functions (excerpt from Annex A) ... 35
Syntax 5-2—Syntax for integer and real numbers (excerpt from Annex A) ... 36
Syntax 5-3—Syntax for attributes (excerpt from Annex A) ... 44
Syntax 6-1—Syntax for net declarations (excerpt from Annex A) ... 57
Syntax 6-2—Syntax for variable declarations (excerpt from Annex A) ... 59
Syntax 6-3—User-defined types (excerpt from Annex A) ... 70
Syntax 6-4—Enumerated types (excerpt from Annex A) ... 72
Syntax 6-5—Parameter declaration syntax (excerpt from Annex A) ... 78
Syntax 6-6—Casting (excerpt from Annex A) ... 90
Syntax 7-1—Structure declaration syntax (excerpt from Annex A) ... 97
Syntax 7-2—Union declaration syntax (excerpt from Annex A) ... 99
Syntax 7-3—Dynamic array new constructor syntax (excerpt from Annex A) ... 107
Syntax 7-4—Declaration of queue dimension (excerpt from Annex A) ... 118
Syntax 7-5—Array method call syntax (not in Annex A) ... 121
Syntax 8-1—Class syntax (excerpt from Annex A) ... 129
Syntax 9-1—Syntax for structured procedures (excerpt from Annex A) ... 151
Syntax 9-2—Syntax for sequential block (excerpt from Annex A) ... 155
Syntax 9-3—Syntax for parallel block (excerpt from Annex A) ... 156
Syntax 9-4—Delay and event control syntax (excerpt from Annex A) ... 162
Syntax 9-5—Syntax for wait statement (excerpt from Annex A) ... 167
Syntax 9-6—Syntax for intra-assignment delay and event control (excerpt from Annex A) ... 168
Syntax 9-7—Syntax for process control statements (excerpt from Annex A) ... 171
Syntax 10-1—Syntax for continuous assignment (excerpt from Annex A) ... 178
Syntax 10-2—Blocking assignment syntax (excerpt from Annex A) ... 182
Syntax 10-3—Nonblocking assignment syntax (excerpt from Annex A) ... 183
Syntax 10-4—Syntax for procedural continuous assignments (excerpt from Annex A) ... 186
Syntax 10-5—Assignment patterns syntax (excerpt from Annex A) ... 191
Syntax 10-6—Syntax for net aliasing (excerpt from Annex A) ... 197
Syntax 11-1—Operator syntax (excerpt from Annex A) ... 200
Syntax 11-2—Conditional operator syntax (excerpt from Annex A) ... 215
Syntax 11-3—Inside expression syntax (excerpt from Annex A) ... 218
Syntax 11-4—Streaming concatenation syntax (excerpt from Annex A) ... 220
Syntax 11-5—With expression syntax (excerpt from Annex A) ... 222
Syntax 11-6—Tagged union syntax (excerpt from Annex A) ... 232
Syntax 11-7—Operator overloading syntax (excerpt from Annex A) ... 235
Syntax 11-8—Syntax for min:typ:max expression (excerpt from Annex A) ... 238
Syntax 11-9—Let syntax (excerpt from Annex A) ... 239
Syntax 12-1—Procedural statement syntax (excerpt from Annex A) ... 246
Syntax 12-2—Syntax for if–else statement (excerpt from Annex A) ... 246

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Syntax 12-3—Syntax for case statements (excerpt from Annex A) ... 251
Syntax 12-4—Pattern syntax (excerpt from Annex A) ... 256
Syntax 12-5—Loop statement syntax (excerpt from Annex A) ... 260
Syntax 12-6—Jump statement syntax (excerpt from Annex A) ... 264
Syntax 13-1—Task syntax (excerpt from Annex A) ... 268
Syntax 13-2—Function syntax (excerpt from Annex A) ... 272
Syntax 13-3—Task or function call syntax (excerpt from Annex A) ... 278
Syntax 14-1—Clocking block syntax (excerpt from Annex A) ... 284
Syntax 14-2—Cycle delay syntax (excerpt from Annex A) ... 289
Syntax 14-3—Default clocking syntax (excerpt from Annex A) ... 290
Syntax 14-4—Global clocking syntax (excerpt from Annex A) ... 292
Syntax 14-5—Synchronous drive syntax (excerpt from Annex A) ... 294
Syntax 15-1—Event trigger syntax (excerpt from Annex A) ... 305
Syntax 15-2—Wait_order event sequencing syntax (excerpt from Annex A) ... 306
Syntax 16-1—Immediate assertion syntax (excerpt from Annex A) ... 310
Syntax 16-2—Deferred immediate assertion syntax (excerpt from Annex A) ... 312
Syntax 16-3—Sequence syntax (excerpt from Annex A) ... 320
Syntax 16-4—Sequence concatenation syntax (excerpt from Annex A) ... 321
Syntax 16-5—Declaring sequence syntax (excerpt from Annex A) ... 324
Syntax 16-6—Sequence repetition syntax (excerpt from Annex A) ... 332
Syntax 16-7—And operator syntax (excerpt from Annex A) ... 342
Syntax 16-8—Intersect operator syntax (excerpt from Annex A) ... 344
Syntax 16-9—Or operator syntax (excerpt from Annex A) ... 345
Syntax 16-10—First_match operator syntax (excerpt from Annex A) ... 348
Syntax 16-11—Throughout construct syntax (excerpt from Annex A) ... 350
Syntax 16-12—Within construct syntax (excerpt from Annex A) ... 351
Syntax 16-13—Assertion variable declaration syntax (excerpt from Annex A) ... 353
Syntax 16-14—Variable assignment syntax (excerpt from Annex A) ... 354
Syntax 16-15—Subroutine call in sequence syntax (excerpt from Annex A) ... 359
Syntax 16-16—Property construct syntax (excerpt from Annex A) ... 362
Syntax 16-17—Implication syntax (excerpt from Annex A) ... 366
Syntax 16-18—Followed-by syntax (excerpt from Annex A) ... 370
Syntax 16-19—Property statement case syntax (excerpt from Annex A) ... 378
Syntax 16-20—Concurrent assert construct syntax (excerpt from Annex A) ... 394
Syntax 16-21—Default clocking and default disable syntax (excerpt from Annex A) ... 410
Syntax 16-22—Expect statement syntax (excerpt from Annex A) ... 418
Syntax 17-1—Checker declaration syntax (excerpt from Annex A) ... 422
Syntax 17-2—Checker instantiation syntax (excerpt from Annex A) ... 424
Syntax 18-1—Random variable declaration syntax (excerpt from Annex A) ... 440
Syntax 18-2—Constraint syntax (excerpt from Annex A) ... 443

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Syntax 18-3—Constraint distribution syntax (excerpt from Annex A) ... 446
Syntax 18-4—Constraint implication syntax (excerpt from Annex A) ... 447
Syntax 18-5—If–else constraint syntax (excerpt from Annex A) ... 448
Syntax 18-6—Foreach iterative constraint syntax (excerpt from Annex A) ... 449
Syntax 18-7—Solve...before constraint ordering syntax (excerpt from Annex A) ... 452
Syntax 18-8—Static constraint syntax (excerpt from Annex A) ... 453
Syntax 18-9—In-line constraint syntax (excerpt from Annex A) ... 459
Syntax 18-10—Scope randomize function syntax (not in Annex A) ... 465
Syntax 18-11—Randcase syntax (excerpt from Annex A) ... 471
Syntax 18-12—Randsequence syntax (excerpt from Annex A) ... 473
Syntax 18-13—Random production weights syntax (excerpt from Annex A) ... 474
Syntax 18-14—If–else conditional random production syntax (excerpt from Annex A) ... 475
Syntax 18-15—Case random production syntax (excerpt from Annex A) ... 475
Syntax 18-16—Repeat random production syntax (excerpt from Annex A) ... 476
Syntax 18-17—Rand join random production syntax (excerpt from Annex A) ... 476
Syntax 18-18—Random production syntax (excerpt from Annex A) ... 478
Syntax 19-1—Covergroup syntax (excerpt from Annex A) ... 484
Syntax 19-2—Coverage point syntax (excerpt from Annex A) ... 488
Syntax 19-3—Transition bin syntax (excerpt from Annex A) ... 492
Syntax 19-4—Cross coverage syntax (excerpt from Annex A) ... 498
Syntax 20-1—Syntax for simulation control tasks (not in Annex A) ... 516
Syntax 20-2—Syntax for time system functions (not in Annex A) ... 516
Syntax 20-3—Syntax for $printtimescale (not in Annex A) ... 518
Syntax 20-4—Syntax for $timeformat (not in Annex A) ... 519
Syntax 20-5—Type name function syntax (not in Annex A) ... 522
Syntax 20-6—Size function syntax (not in Annex A) ... 523
Syntax 20-7—Range function syntax (not in Annex A) ... 524
Syntax 20-8—Array querying function syntax (not in Annex A) ... 524
Syntax 20-9—Severity system task syntax (not in Annex A) ... 528
Syntax 20-10—Elaboration system task syntax (excerpt from Annex A) ... 529
Syntax 20-11—Assertion control syntax (not in Annex A) ... 530
Syntax 20-12—Assertion action control syntax (not in Annex A) ... 531
Syntax 20-13—Assertion system function syntax (not in Annex A) ... 533
Syntax 20-14—Syntax for $random (not in Annex A) ... 534
Syntax 20-15—Syntax for probabilistic distribution functions (not in Annex A) ... 535
Syntax 20-16—Syntax for PLA modeling system task (not in Annex A) ... 538
Syntax 20-17—System function syntax (not in Annex A) ... 542
Syntax 21-1—Syntax for $display and $write system tasks (not in Annex A) ... 544
Syntax 21-2—Syntax for $strobe system tasks (not in Annex A) ... 553
Syntax 21-3—Syntax for $monitor system tasks (not in Annex A) ... 553

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Syntax 21-4—Syntax for $fopen and $fclose system tasks (not in Annex A) 554
Syntax 21-5—Syntax for file output system tasks (not in Annex A) 556
Syntax 21-6—Syntax for formatting data tasks (not in Annex A) 557
Syntax 21-7—Syntax for file read system functions (not in Annex A)..... 558
Syntax 21-8—Syntax for file positioning system functions (not in Annex A)..... 563
Syntax 21-9—Syntax for file flush system task (not in Annex A) 564
Syntax 21-10—Syntax for file I/O error detection system function (not in Annex A) 564
Syntax 21-11—Syntax for end-of-file file detection system function (not in Annex A) 564
Syntax 21-12—Syntax for memory load system tasks (not in Annex A)..... 565
Syntax 21-13—Writemem system task syntax (not in Annex A)..... 568
Syntax 21-14—Syntax for $dumpfile task (not in Annex A) 573
Syntax 21-15—Syntax for $dumpvars task (not in Annex A)..... 573
Syntax 21-16—Syntax for $dumpoff and $dumpon tasks (not in Annex A) 574
Syntax 21-17—Syntax for $dumpall task (not in Annex A) 575
Syntax 21-18—Syntax for $dumplimit task (not in Annex A)..... 575
Syntax 21-19—Syntax for $dumpflush task (not in Annex A) 575
Syntax 21-20—Syntax for output 4-state VCD file (not in Annex A) 577
Syntax 21-21—Syntax for $dumpports task (not in Annex A) 582
Syntax 21-22—Syntax for $dumpportsoff and $dumpportson system tasks (not in Annex A) 583
Syntax 21-23—Syntax for $dumpportsall system task (not in Annex A) 584
Syntax 21-24—Syntax for $dumpportslimit system task (not in Annex A)..... 584
Syntax 21-25—Syntax for $dumpportsflush system task (not in Annex A) 585
Syntax 21-26—Syntax for $vcdclose keyword (not in Annex A)..... 585
Syntax 21-27—Syntax for output extended VCD file (not in Annex A) 587
Syntax 21-28—Syntax for extended VCD node information (not in Annex A) 587
Syntax 21-29—Syntax for value change section (not in Annex A)..... 589
Syntax 22-1—Syntax for include compiler directive (not in Annex A)..... 594
Syntax 22-2—Syntax for text macro definition (not in Annex A) 595
Syntax 22-3—Syntax for text macro usage (not in Annex A)..... 596
Syntax 22-4—Syntax for undef compiler directive (not in Annex A)..... 600
Syntax 22-5—Syntax for conditional compilation directives (not in Annex A) 600
Syntax 22-6—Syntax for timescale compiler directive (not in Annex A)..... 603
Syntax 22-7—Syntax for default_nettype compiler directive (not in Annex A)..... 605
Syntax 22-8—Syntax for pragma compiler directive (not in Annex A)..... 606
Syntax 22-9—Syntax for line compiler directive (not in Annex A)..... 607
Syntax 22-10—Syntax for begin_keywords and end_keywords compiler directives (not in Annex A)..... 608
Syntax 23-1—Module declaration syntax (excerpt from Annex A)..... 615
Syntax 23-2—Non-ANSI style module header declaration syntax (excerpt from Annex A) 616
Syntax 23-3—Non-ANSI style port declaration syntax (excerpt from Annex A)..... 617
Syntax 23-4—ANSI style list_of_port_declarations syntax (excerpt from Annex A) 620

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Syntax 23-5—Module item syntax (excerpt from Annex A) 626
Syntax 23-6—Module instance syntax (excerpt from Annex A) 627
Syntax 23-7—Syntax for hierarchical path names (excerpt from Annex A)..... 639
Syntax 23-8—Syntax for upward name referencing (not in Annex A) 642
Syntax 23-9—Bind construct syntax (excerpt from Annex A) 655
Syntax 24-1—Program declaration syntax (excerpt from Annex A) 660
Syntax 25-1—Interface syntax (excerpt from Annex A)..... 666
Syntax 25-2—Modport clocking declaration syntax (excerpt from Annex A) 676
Syntax 25-3—Virtual interface declaration syntax (excerpt from Annex A)..... 686
Syntax 26-1—Package declaration syntax (excerpt from Annex A)..... 694
Syntax 26-2—Package import syntax (excerpt from Annex A)..... 695
Syntax 26-3—Package import in header syntax (excerpt from Annex A) 699
Syntax 26-4—Package export syntax (excerpt from Annex A) 701
Syntax 26-5—Std package import syntax (not in Annex A) 703
Syntax 27-1—Syntax for generate constructs (excerpt from Annex A)..... 707
Syntax 28-1—Syntax for gate instantiation (excerpt from Annex A) 718
Syntax 29-1—Syntax for UDPs (excerpt from Annex A)..... 750
Syntax 29-2—Syntax for UDP instances (excerpt from Annex A)..... 757
Syntax 30-1—Syntax for specify block (excerpt from Annex A) 761
Syntax 30-2—Syntax for module path declaration (excerpt from Annex A)..... 762
Syntax 30-3—Syntax for simple module path (excerpt from Annex A)..... 763
Syntax 30-4—Syntax for edge-sensitive path declaration (excerpt from Annex A)..... 764
Syntax 30-5—Syntax for state-dependent paths (excerpt from Annex A)..... 765
Syntax 30-6—Syntax for path delay value (excerpt from Annex A) 772
Syntax 30-7—Syntax for PATHPULSE$ pulse control (excerpt from Annex A) 777
Syntax 30-8—Syntax for pulse style declarations (excerpt from Annex A) 779
Syntax 30-9—Syntax for showcancelled declarations (excerpt from Annex A)..... 780
Syntax 31-1—Syntax for system timing checks (excerpt from Annex A) 786
Syntax 31-2—Syntax for time check conditions and timing check events (excerpt from Annex A)..... 787
Syntax 31-3—Syntax for $setup (excerpt from Annex A) 788
Syntax 31-4—Syntax for $hold (excerpt from Annex A) 789
Syntax 31-5—Syntax for $setuphold (excerpt from Annex A)..... 790
Syntax 31-6—Syntax for $removal (excerpt from Annex A) 792
Syntax 31-7—Syntax for $recovery (excerpt from Annex A)..... 792
Syntax 31-8—Syntax for $recrem (excerpt from Annex A) 793
Syntax 31-9—Syntax for $skew (excerpt from Annex A) 795
Syntax 31-10—Syntax for $timeskew (excerpt from Annex A) 796
Syntax 31-11—Syntax for $fullskew (excerpt from Annex A)..... 799
Syntax 31-12—Syntax for $width (excerpt from Annex A) 801
Syntax 31-13—Syntax for $period (excerpt from Annex A) 802

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Syntax 31-14—Syntax for $nochange (excerpt from Annex A) 803
Syntax 31-15—Syntax for edge-control specifier (excerpt from Annex A)..... 804
Syntax 31-16—Syntax for controlled timing check events (excerpt from Annex A) 807
Syntax 32-1—Syntax for $sdf_annotate system task (not in Annex A)..... 823
Syntax 33-1—Syntax for cell (excerpt from Annex A)..... 826
Syntax 33-2—Syntax for declaring library in library map file (excerpt from Annex A) 827
Syntax 33-3—Syntax for include command (excerpt from Annex A)..... 828
Syntax 33-4—Syntax for configurations (excerpt from Annex A) 829
Syntax 35-1—DPI import declaration syntax (excerpt from Annex A)..... 869
Syntax 35-2—DPI export declaration syntax (excerpt from Annex A) 874

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

INTERNATIONAL ELECTROTECHNICAL COMMISSION

SystemVerilog – Unified Hardware Design, Specification, and Verification Language

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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

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 62530/IEEE Std 1800 has been processed through IEC technical committee 93: *Design automation*.

This second edition cancels and replaces the first edition published in 2007. This edition constitutes a technical revision.

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

IEEE Std	FDIS	Report on voting
1800 (2009)	93/303/FDIS	93/305/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.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub
Full document details: IEC 62530 2011

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

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

IMPORTANT – The 'color inside' logo on the cover page of this publication indicates that it contains colors which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a color printer.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

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, Piscataway, NJ 08854, 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 license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

IEEE Standard for SystemVerilog — Unified Hardware Design, Specification, and Verification Language

Sponsor
Design Automation Standards Committee
of the
IEEE Computer Society

and the
IEEE Standards Association Corporate Advisory Group

Approved 11 November 2009
IEEE-SA Standards Board

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Abstract: This standard represents a merger of two previous standards: IEEE Std 1364™-2005 Verilog hardware description language (HDL) and IEEE Std 1800-2005 SystemVerilog unified hardware design, specification, and verification language. The 2005 SystemVerilog standard defines extensions to the 2005 Verilog standard. These two standards were designed to be used as one language. Merging the base Verilog language and the SystemVerilog extensions into a single standard provides users with all information regarding syntax and semantics in a single document.

Keywords: assertions, design automation, design verification, hardware description language, HDL, HDVL, PLI, programming language interface, SystemVerilog, Verilog, VPI

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

IEEE introduction

The purpose of this standard is to provide the electronic design automation (EDA), semiconductor, and system design communities with a well-defined and official IEEE unified hardware design, specification, and verification standard language. The language is designed to coexist and enhance the hardware description and verification languages (HDVLs) presently used by designers while providing the capabilities lacking in those languages.

SystemVerilog is a unified hardware design, specification, and verification language based on the Accellera SystemVerilog 3.1a extensions to the Verilog HDL [B3][a], published in 2004. Accellera is a consortium of EDA, semiconductor, and system companies. IEEE Std 1800 enables a productivity boost in design and validation and covers design, simulation, validation, and formal assertion-based verification flows.

SystemVerilog enables the use of a unified language for abstract and detailed specification of the design, specification of assertions, coverage, and testbench verification based on manual or automatic methodologies. SystemVerilog offers application programming interfaces (APIs) for coverage and assertions, a vendor-independent API to access proprietary waveform file formats, and a direct programming interface (DPI) to access proprietary functionality. SystemVerilog offers methods that allow designers to continue to use present design languages when necessary to leverage existing designs and intellectual property. This standardization project will provide the VLSI design engineers with a well-defined IEEE standard, which meets their requirements in design and validation, and which enables a step function increase in their productivity. This standardization project will also provide the EDA industry with a standard to which they can adhere and which they can support in order to deliver their solutions in this area.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard 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.

[a]The numbers in brackets correspond to the numbers in the bibliography in Annex R.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Updating of IEEE documents

Users of IEEE standards 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 Standards Association website at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org.

Errata

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

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this amendment may require use of subject matter covered by patent rights. By publication of this amendment, no position is taken with respect to the existence or validity of any patent rights in connection therewith. 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 Patents 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 amendment 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.

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection in all circumstances. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

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.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Part One:
Design and Verification Constructs

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

SystemVerilog — Unified Hardware Design, Specification, and Verification Language

1. Overview

1.1 Scope

This SystemVerilog standard (IEEE Std 1800) is a Unified Hardware Design, Specification, and Verification language. IEEE Std 1364™-2005 Verilog is a design language. Both standards were approved by the IEEE-SASB in November 2005. This standard creates new revisions of the IEEE 1364 Verilog and IEEE 1800 SystemVerilog standards, which include errata fixes and resolutions, enhancements, enhanced assertion language, merger of Verilog Language Reference Manual (LRM) and SystemVerilog 1800 LRM into a single LRM, integration with Verilog-AMS, and ensures interoperability with other languages such as SystemC and VHDL.

1.2 Purpose

The purpose of this project is to provide the EDA, Semiconductor, and System Design communities with a solid and well-defined IEEE Unified Hardware Design, Specification and Verification standard language, while resolving errata and developing enhancements to the current IEEE 1800 SystemVerilog standard. The language is designed to co-exist, be interoperable, possibly merge, and enhance those hardware description languages presently used by designers.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

1.3 Merger of IEEE Std 1364-2005 and IEEE Std 1800-2005

This standard represents a merger of two previous standards: the IEEE Std 1364-2005[1] Verilog Hardware Description Language (HDL) and the IEEE Std 1800-2005 SystemVerilog Unified Hardware Design, Specification, and Verification Language. In these previous standards, Verilog was the base language and defined a completely self-contained standard. SystemVerilog defined a number of significant extensions to Verilog, but IEEE Std 1800-2005 was not a self-contained standard; IEEE Std 1800-2005 referred to, and relied on, IEEE Std 1364-2005. These two standards were designed to be used as one language. Merging the base Verilog language and the SystemVerilog extensions into a single standard enables users to have all information regarding syntax and semantics in a single document.

This standard serves as a complete specification of the SystemVerilog language. This standard contains the following:

— The formal syntax and semantics of all SystemVerilog constructs
— Simulation system tasks and system functions, such as text output display commands
— Compiler directives, such as text substitution macros and simulation time scaling
— The Programming Language Interface (PLI) mechanism
— The formal syntax and semantics of the SystemVerilog Verification Procedural Interface (VPI)
— An Application Programming Interface (API) for coverage access not included in VPI
— Direct programming interface (DPI) for interoperation with the C programming language
— VPI, API, and DPI header files
— Concurrent assertion formal semantics
— The formal syntax and semantics of standard delay format (SDF) constructs
— Informative usage examples

1.4 Special terms

Throughout this standard, the following terms apply:

— *SystemVerilog* refers to the unified Verilog base language (IEEE Std 1364) with the SystemVerilog extensions to Verilog.
— *SystemVerilog 3.1a* refers to the Accellera SystemVerilog Language Reference Manual [B3],[2] a precursor to IEEE Std 1800-2005.
— *Verilog* refers to IEEE Std 1364-2005 for the Verilog HDL.
— *Language Reference Manual* (*LRM*) refers to the document describing a Verilog or SystemVerilog standard.
— *Tool* refers to a software implementation that reads SystemVerilog source code, such as a logic simulator.

NOTE—In the IEEE Std 1800-2005 version of the standard, *SystemVerilog* referred to just the extensions to the IEEE Std 1364-2005 Verilog language, and did not include the Verilog base language.[3]

1.5 Conventions used in this standard

This standard is organized into clauses, each of which focuses on a specific area of the language. There are subclauses within each clause to discuss individual constructs and concepts. The discussion begins with an

[1]Information on references can be found in Clause 2.
[2]The numbers in brackets correspond to those of the bibliography in Annex R.
[3]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.
Documents
Document Sharing Hub
Full document details: IEC 62530 2011

introduction and an optional rationale for the construct or the concept, followed by syntax and semantic descriptions, followed by examples and notes.

The terminology conventions used throughout this standard are as follows:

— The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).

— The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).

— The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

— The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

1.6 Syntactic description

The main text uses the following conventions:

— *Italicized* font when a term is being defined

— `Constant-width` font for examples, file names, and references to constants, especially `0`, `1`, `x`, and `z` values

— **`Boldface constant-width`** font for SystemVerilog keywords, when referring to the actual keyword

The formal syntax of SystemVerilog is described using Backus-Naur Form (BNF). The following conventions are used:

— Lowercase words, some containing embedded underscores, denote syntactic categories. For example:

```
module_declaration
```

— **Boldface-red** characters denote reserved keywords, operators, and punctuation marks as a required part of the syntax. For example:

```
module   =>   ;
```

— A vertical bar (|) that is not in boldface-red separates alternative items. For example:

```
unary_operator ::=
      + | - | ! | ~ | & | ~& | | | ~| | ^ | ~^ | ^~
```

— Square brackets ([ ]) that are not in boldface-red enclose optional items. For example:

```
function_declaration ::= function [ lifetime ] function_body_declaration
```

— Braces ({ }) that are not in boldface-red enclose a repeated item. The item may appear zero or more times; the repetitions occur from left to right as with an equivalent left-recursive rule. Thus, the following two rules are equivalent:

```
list_of_param_assignments ::= param_assignment { , param_assignment }
list_of_param_assignments ::=
      param_assignment
    | list_of_param_assignments , param_assignment
```

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

A *qualified term* in the syntax is a term such as *array_identifier* for which the "array" portion represents some semantic intent and the "identifier" term indicates that the qualified term reduces to the "identifier" term in the syntax. The syntax does not completely define the semantics of such qualified terms; for example while an identifier which would qualify semantically as an array_identifier is created by a declaration, such declaration forms are not explicitly described using *array_identifier* in the syntax.

1.7 Use of color in this standard

This standard uses a minimal amount of color to enhance readability. The coloring is not essential and does not affect the accuracy of this standard when viewed in pure black and white. The places where color is used are the following:

— Cross references that are hyperlinked to other portions of this standard are shown in underlined-blue text (hyperlinking works when this standard is viewed interactively as a PDF file).

— Syntactic keywords and tokens in the formal language definitions are shown in **`boldface-red text`**.

— Some figures use a minimal amount of color to enhance readability.

1.8 Contents of this standard

A synopsis of the clauses and annexes is presented as a quick reference. All clauses and several of the annexes are normative parts of this standard. Some annexes are included for informative purposes only.

Part One: Design and Verification Constructs

Clause 1 describes the contents of this standard and the conventions used in this standard.

Clause 2 lists references to other standards that are required in order to implement this standard.

Clause 3 introduces the major building blocks that make up a SystemVerilog design and verification environment: modules, programs, interfaces, checkers, packages and configurations. This clause also discusses primitives, name spaces, the `$unit` compilation space, and the concept of simulation time.

Clause 4 describes the SystemVerilog simulation scheduling semantics.

Clause 5 describes the lexical tokens used in SystemVerilog source text and their conventions.

Clause 6 describes SystemVerilog data objects and types, including nets and variables, their declaration syntax and usage rules, and charge strength of the values on nets. This clause also discusses strings and string methods, enumerated types, user-defined types, constants, data scope and lifetime, and type compatibility.

Clause 7 describes SystemVerilog compound data types: structures, unions, arrays, including packed and unpacked arrays, dynamic arrays, associative arrays, and queues. This clause also describes various array methods.

Clause 8 describes the object-oriented programming capabilities in SystemVerilog. Topics include defining classes, dynamically constructing objects, inheritance and subclasses, data hiding and encapsulation, polymorphism, and parameterized classes.

Clause 9 describes the SystemVerilog procedural blocks: **`initial`**, **`always`**, **`always_comb`**, **`always_ff`**, **`always_latch`**, and **`final`**. Sequential and parallel statement grouping, block names, statement labels, and process control are also described.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Clause 10 describes continuous assignments, blocking and nonblocking procedural assignments, and procedural continuous assignments.

Clause 11 describes the operators and operands that can be used in expressions. This clause also discusses operations on arrays, operator methods, and operator overloading.

Clause 12 describes SystemVerilog procedural programming statements, such as decision statements and looping constructs.

Clause 13 describes tasks and functions, which are subroutines that can be called from more than one place in a behavioral model.

Clause 14 defines clocking blocks, input and output skews, cycle delays, and default clocking.

Clause 15 describes interprocess communications using event types and event controls, and built-in semaphore and mailbox classes.

Clause 16 describes immediate and concurrent assertions, properties, sequences, sequence operations, multiclock sequences, and clock resolution.

Clause 17 describes checkers. Checkers allow the encapsulation of assertions and modeling code to create a single verification entity.

Clause 18 describes generating random numbers, constraining random number generation, dynamically changing constraints, seeding random number generators (RNGs), and randomized `case` statement execution.

Clause 19 describes coverage groups, coverage points, cross coverage, coverage options, and coverage methods.

Clause 20 describes most of the built-in system tasks and system functions.

Clause 21 describes additional system tasks and system functions that are specific to I/O operations.

Clause 22 describes various compiler directives, including a directive for controlling reserved keyword compatibility between versions of previous Verilog and SystemVerilog standards.

Part Two: Hierarchy Constructs

Clause 23 describes how hierarchies are created in SystemVerilog using module instances and interface instances, and port connection rules. This clause also discusses the $root top-level instances, nested modules, extern modules, identifier search rules, how parameter values can be overridden, and binding auxiliary code to scopes or instances.

Clause 24 describes the testbench program construct, the elimination of testbench race conditions, and program control tasks.

Clause 25 describes interface syntax, interface ports, modports, interface subroutines, parameterized interfaces, virtual interfaces, and accessing objects within interfaces.

Clause 26 describes user-defined packages and the std built-in package.

Clause 27 describes the generate construct and how generated constructs can be used to do conditional or multiple instantiations of procedural code or hierarchy.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Clause 28 describes the gate- and switch-level primitives and logic strength modeling.

Clause 29 describes how a User Defined Primitive (UDP) can be defined and how these primitives are included in SystemVerilog models.

Clause 30 describes how to specify timing relationships between input and output ports of a module.

Clause 31 describes how timing checks are used in specify blocks to determine whether signals obey the timing constraints.

Clause 32 describes the syntax and semantics of SDF constructs.

Clause 33 describes how to configure the contents of a design.

Clause 34 describes encryption and decryption of source text regions.

Part Three: Application Programming Interfaces (APIs)

Clause 35 describes SystemVerilog's direct programming interface (DPI), a direct interface to foreign languages and the syntax for importing functions from a foreign language and exporting subroutines to a foreign language.

Clause 36 provides an overview of the Programming Language Interface (PLI and VPI).

Clause 37 presents the VPI data model diagrams, which document the VPI object relationships and access methods.

Clause 38 describes the VPI routines.

Clause 39 describes the assertion API in SystemVerilog.

Clause 40 describes the coverage API in SystemVerilog.

Part Four: Annexes

Annex A (normative) defines the formal syntax of SystemVerilog, using BNF.

Annex B (normative) lists the SystemVerilog keywords.

Annex C (informative) lists constructs that have been deprecated from SystemVerilog. The annex also discusses the possible deprecation of the **`defparam`** statement and the procedural **`assign`**/**`deassign`** statements.

Annex D (informative) describes system tasks and system functions that are frequently used, but that are not required in this standard.

Annex E (informative) describes compiler directives that are frequently used, but that are not required in this standard.

Annex F (normative) describes a formal semantics for SystemVerilog concurrent assertions.

Annex G (normative) describes the SystemVerilog standard package, containing type definitions for mailbox, semaphore, randomize, and process.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

Annex H (normative) defines the C-language layer for the SystemVerilog DPI.

Annex I (normative) defines the standard `svdpi.h` include file for use with SystemVerilog DPI applications.

Annex J (normative) describes common guidelines for the inclusion of foreign language code into a SystemVerilog application.

Annex K (normative) provides a listing of the contents of the `vpi_user.h` file.

Annex L (normative) provides a listing of the contents of the `vpi_compatibility.h` file, which extends the `vpi_user.h` include file.

Annex M (normative) provides a listing of the contents of the `sv_vpi_user.h` file, which extends the `vpi_user.h` include file.

Annex N (normative) provides the C source code for the SystemVerilog random distribution system functions.

Annex O (informative) describes various scenarios that can be used for intellectual property (IP) protection, and it also shows how the relevant pragmas can be used to achieve the desired effect of securely protecting, distributing, and decrypting the model.

Annex P (informative) defines terms that are used in this standard.

Annex Q (informative) provides a general mapping of clause numbers from IEEE Std 1364-2005 Verilog standard and IEEE Std 1800-2005 SystemVerilog standard into this standard.

Annex R (informative) lists reference documents that are related to this standard.

1.9 Deprecated clauses

Annex C lists constructs that appeared in previous versions of either IEEE Std 1364 or IEEE Std 1800, but that have been deprecated and do not appear in this standard. This annex also lists constructs that appear in this standard, but that are under consideration for deprecation in a future version of this standard.

1.10 Examples

Small SystemVerilog code examples are shown throughout this standard. These examples are informative. They are intended to illustrate the usage of SystemVerilog constructs in a simple context and do not define the full syntax.

1.11 Prerequisites

Some clauses of this standard presuppose a working knowledge of the C programming language.

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

2. Normative references

The following referenced documents are indispensable for the application of this standard (i.e., they must be understood and used, so each referenced document is cited in the text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

Anderson, R., Biham, E., and Knudsen, L. "Serpent: A Proposal for the Advanced Encryption Standard," NIST AES Proposal, 1998, http://www.cl.cam.ac.uk/~rja14/Papers/serpent.tar.gz.

ANSI Std X9.52-1998, American National Standard for Financial Services—Triple Data Encryption Algorithm Modes of Operation.[4]

ElGamal, T., "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469–472, July 1985.

FIPS 46-3 (October 1999), Data Encryption Standard (DES).[5]

FIPS 180-2 (August 2002), Secure Hash Standard (SHS).

FIPS 197 (November 2001), Advanced Encryption Standard (AES).

IEC 62530:2007, Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.
IEEE Std 1800™-2005, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

NOTE—IEEE Std 1800™-2005 was adopted as IEC 62530:2007.

IEEE Std 754™, IEEE Standard for Binary Floating-Point Arithmetic.[6, 7]

IEEE Std 1003.1™, IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®).

IEEE Std 1364™-1995, IEEE Standard Hardware Description Language Based on the Verilog™ Hardware Description Language.

IEEE Std 1364™-2001, IEEE Standard for Verilog Hardware Description Language.

IEEE Std 1364™-2005, IEEE Standard for Verilog Hardware Description Language.

IETF RFC 1319 (April 1992), The MD2 Message-Digest Algorithm.[8]

IETF RFC 1321 (April 1992), The MD5 Message-Digest Algorithm.

IETF RFC 2045 (November 1996), Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies.

IETF RFC 2144 (May 1997), The CAST-128 Encryption Algorithm.

---

[4]ANSI publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[5]FIPS publications are available from the National Technical Information Service (NTIS), U. S. Dept. of Commerce, 5285 Port Royal Rd., Springfield, VA 22161 (http://www.ntis.org/).

[6]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[7]The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[8]IETF requests for comments (RFCs) are available from the Internet Engineering Task Force (http://www.ieft.org).

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011

IETF RFC 2437 (October 1998), PKCS #1: RSA Cryptography Specifications, Version 2.0.

IETF RFC 2440 (November 1998), OpenPGP Message Format.

ISO/IEC 10118-3:2004, Information technology—Security techniques—Hash-functions—Part 3: Dedicated hash-functions.[9]

Schneier, B., "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993)*, Springer-Verlag, 1994, pp. 191–204.

Schneier, B., et al, *The Twofish Encryption Algorithm: A 128-Bit Block Cipher*, 1st ed., Wiley, 1999.

[9]ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 chemin de la Voie-Creuse, CH-1211 Genève 20, Switzerland/Suisse (http://www.iso.ch/) and from the IEC Central Office, Case Postale 131, 3 rue de Varembé, CH-1211 Genève 20, Switzerland/Suisse (http://www.iec.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

Published by IEC under license from IEEE. © 2009 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: IEC 62530 2011