BS IEC 62530:2021

BSI Standards Publication

SystemVerilog — Unified Hardware Design, Specification, and Verification Language

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

National foreword

This British Standard is the UK implementation of IEC 62530:2021. It supersedes BS IEC 62530:2011, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee EPL/501, Electronic Assembly Technology.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2021
Published by BSI Standards Limited 2021

ISBN 978 0 539 17341 3

ICS 25.040.01; 35.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 August 2021.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

IEC 62530

Edition 3.0 2021-07

INTERNATIONAL STANDARD

IEEE Std 1800™

SystemVerilog – Unified Hardware Design, Specification, and Verification Language

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.01; 35.060

ISBN 978-2-8322-9977-7

Warning! Make sure that you obtained this publication from an authorized distributor.

® Registered trademark of the International Electrotechnical Commission

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

Contents

Part One: Design and Verification Constructs

1. Overview ... 38

1.1 Scope ... 38
1.2 Purpose ... 38
1.3 Content summary ... 38
1.4 Special terms ... 39
1.5 Conventions used in this standard ... 39
1.6 Syntactic description ... 40
1.7 Use of color in this standard ... 40
1.8 Contents of this standard ... 41
1.9 Deprecated clauses ... 44
1.10 Examples ... 44
1.11 Prerequisites ... 44

2. Normative references ... 45

3. Design and verification building blocks ... 47

3.1 General ... 47
3.2 Design elements ... 47
3.3 Modules ... 47
3.4 Programs ... 48
3.5 Interfaces ... 49
3.6 Checkers ... 50
3.7 Primitives ... 50
3.8 Subroutines ... 50
3.9 Packages ... 50
3.10 Configurations ... 51
3.11 Overview of hierarchy ... 51
3.12 Compilation and elaboration ... 52
3.13 Name spaces ... 54
3.14 Simulation time units and precision ... 55

4. Scheduling semantics ... 59

4.1 General ... 59
4.2 Execution of a hardware model and its verification environment ... 59
4.3 Event simulation ... 59
4.4 Stratified event scheduler ... 60
4.5 SystemVerilog simulation reference algorithm ... 65
4.6 Determinism ... 65
4.7 Nondeterminism ... 66
4.8 Race conditions ... 66

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

4.9 Scheduling implication of assignments ... 66
4.10 PLI callback control points ... 68

5. Lexical conventions ... 69

5.1 General ... 69
5.2 Lexical tokens ... 69
5.3 White space ... 69
5.4 Comments ... 69
5.5 Operators ... 69
5.6 Identifiers, keywords, and system names ... 70
5.7 Numbers ... 71
5.8 Time literals ... 76
5.9 String literals ... 76
5.10 Structure literals ... 78
5.11 Array literals ... 79
5.12 Attributes ... 79
5.13 Built-in methods ... 81

6. Data types ... 83

6.1 General ... 83
6.2 Data types and data objects ... 83
6.3 Value set ... 83
6.4 Singular and aggregate types ... 84
6.5 Nets and variables ... 85
6.6 Net types ... 86
6.7 Net declarations ... 97
6.8 Variable declarations ... 100
6.9 Vector declarations ... 102
6.10 Implicit declarations ... 103
6.11 Integer data types ... 104
6.12 Real, shortreal, and realtime data types ... 105
6.13 Void data type ... 105
6.14 Chandle data type ... 105
6.15 Class ... 106
6.16 String data type ... 106
6.17 Event data type ... 112
6.18 User-defined types ... 112
6.19 Enumerations ... 114
6.20 Constants ... 119
6.21 Scope and lifetime ... 126
6.22 Type compatibility ... 128
6.23 Type operator ... 131
6.24 Casting ... 132
6.25 Parameterized data types ... 137

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

7. Aggregate data types ... 139

7.1 General ... 139
7.2 Structures ... 139
7.3 Unions ... 141
7.4 Packed and unpacked arrays ... 145
7.5 Dynamic arrays ... 149
7.6 Array assignments ... 152
7.7 Arrays as arguments to subroutines ... 153
7.8 Associative arrays ... 155
7.9 Associative array methods ... 157
7.10 Queues ... 161
7.11 Array querying functions ... 165
7.12 Array manipulation methods ... 165

8. Classes ... 170

8.1 General ... 170
8.2 Overview ... 170
8.3 Syntax ... 171
8.4 Objects (class instance) ... 172
8.5 Object properties and object parameter data ... 173
8.6 Object methods ... 174
8.7 Constructors ... 174
8.8 Typed constructor calls ... 176
8.9 Static class properties ... 177
8.10 Static methods ... 177
8.11 This ... 177
8.12 Assignment, renaming, and copying ... 178
8.13 Inheritance and subclasses ... 180
8.14 Overridden members ... 180
8.15 Super ... 181
8.16 Casting ... 182
8.17 Chaining constructors ... 182
8.18 Data hiding and encapsulation ... 183
8.19 Constant class properties ... 184
8.20 Virtual methods ... 184
8.21 Abstract classes and pure virtual methods ... 186
8.22 Polymorphism: dynamic method lookup ... 187
8.23 Class scope resolution operator :: ... 187
8.24 Out-of-block declarations ... 189
8.25 Parameterized classes ... 191
8.26 Interface classes ... 194
8.27 Typedef class ... 203
8.28 Classes and structures ... 203
8.29 Memory management ... 204

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

9. Processes ... 205

9.1 General ... 205
9.2 Structured procedures ... 205
9.3 Block statements ... 209
9.4 Procedural timing controls ... 215
9.5 Process execution threads ... 224
9.6 Process control ... 225
9.7 Fine-grain process control ... 229

10. Assignment statements ... 232

10.1 General ... 232
10.2 Overview ... 232
10.3 Continuous assignments ... 233
10.4 Procedural assignments ... 236
10.5 Variable declaration assignment (variable initialization) ... 241
10.6 Procedural continuous assignments ... 241
10.7 Assignment extension and truncation ... 243
10.8 Assignment-like contexts ... 244
10.9 Assignment patterns ... 245
10.10 Unpacked array concatenation ... 249
10.11 Net aliasing ... 252

11. Operators and expressions ... 254

11.1 General ... 254
11.2 Overview ... 254
11.3 Operators ... 255
11.4 Operator descriptions ... 259
11.5 Operands ... 279
11.6 Expression bit lengths ... 282
11.7 Signed expressions ... 285
11.8 Expression evaluation rules ... 286
11.9 Tagged union expressions and member access ... 287
11.10 String literal expressions ... 288
11.11 Minimum, typical, and maximum delay expressions ... 290
11.12 Let construct ... 291

12. Procedural programming statements ... 298

12.1 General ... 298
12.2 Overview ... 298
12.3 Syntax ... 298
12.4 Conditional if–else statement ... 299
12.5 Case statement ... 304
12.6 Pattern matching conditional statements ... 309
12.7 Loop statements ... 313
12.8 Jump statements ... 317

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

13. Tasks and functions (subroutines) ... 319

13.1 General ... 319
13.2 Overview ... 319
13.3 Tasks ... 319
13.4 Functions ... 323
13.5 Subroutine calls and argument passing ... 329
13.6 Import and export functions ... 334
13.7 Task and function names ... 334
13.8 Parameterized tasks and functions ... 334

14. Clocking blocks ... 336

14.1 General ... 336
14.2 Overview ... 336
14.3 Clocking block declaration ... 336
14.4 Input and output skews ... 338
14.5 Hierarchical expressions ... 339
14.6 Signals in multiple clocking blocks ... 340
14.7 Clocking block scope and lifetime ... 340
14.8 Multiple clocking blocks example ... 340
14.9 Interfaces and clocking blocks ... 341
14.10 Clocking block events ... 342
14.11 Cycle delay: ## ... 342
14.12 Default clocking ... 343
14.13 Input sampling ... 344
14.14 Global clocking ... 345
14.15 Synchronous events ... 349
14.16 Synchronous drives ... 349

15. Interprocess synchronization and communication ... 354

15.1 General ... 354
15.2 Overview ... 354
15.3 Semaphores ... 354
15.4 Mailboxes ... 356
15.5 Named events ... 359

16. Assertions ... 364

16.1 General ... 364
16.2 Overview ... 364
16.3 Immediate assertions ... 364
16.4 Deferred assertions ... 367
16.5 Concurrent assertions overview ... 374
16.6 Boolean expressions ... 377
16.7 Sequences ... 378
16.8 Declaring sequences ... 382
16.9 Sequence operations ... 390

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

16.10 Local variables ... 413
16.11 Calling subroutines on match of a sequence ... 419
16.12 Declaring properties ... 420
16.13 Multiclock support ... 446
16.14 Concurrent assertions ... 456
16.15 Disable iff resolution ... 473
16.16 Clock resolution ... 475
16.17 Expect statement ... 481
16.18 Clocking blocks and concurrent assertions ... 482

17. Checkers ... 484

17.1 Overview ... 484
17.2 Checker declaration ... 484
17.3 Checker instantiation ... 487
17.4 Context inference ... 490
17.5 Checker procedures ... 491
17.6 Covergroups in checkers ... 493
17.7 Checker variables ... 494
17.8 Functions in checkers ... 500
17.9 Complex checker example ... 501

18. Constrained random value generation ... 503

18.1 General ... 503
18.2 Overview ... 503
18.3 Concepts and usage ... 503
18.4 Random variables ... 506
18.5 Constraint blocks ... 509
18.6 Randomization methods ... 528
18.7 In-line constraints—randomize() with ... 530
18.8 Disabling random variables with rand_mode() ... 532
18.9 Controlling constraints with constraint_mode() ... 534
18.10 Dynamic constraint modification ... 535
18.11 In-line random variable control ... 535
18.12 Randomization of scope variables—std::randomize() ... 536
18.13 Random number system functions and methods ... 538
18.14 Random stability ... 540
18.15 Manually seeding randomize ... 542
18.16 Random weighted case—randcase ... 543
18.17 Random sequence generation—randsequence ... 544

19. Functional coverage ... 553

19.1 General ... 553
19.2 Overview ... 553
19.3 Defining the coverage model: covergroup ... 554
19.4 Using covergroup in classes ... 556
19.5 Defining coverage points ... 558

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

19.6 Defining cross coverage ... 569
19.7 Specifying coverage options ... 578
19.8 Predefined coverage methods ... 582
19.9 Predefined coverage system tasks and system functions ... 585
19.10 Organization of option and type_option members ... 585
19.11 Coverage computation ... 586

20. Utility system tasks and system functions ... 591

20.1 General ... 591
20.2 Simulation control system tasks ... 592
20.3 Simulation time system functions ... 592
20.4 Timescale system tasks ... 594
20.5 Conversion functions ... 597
20.6 Data query functions ... 598
20.7 Array query functions ... 600
20.8 Math functions ... 603
20.9 Bit vector system functions ... 604
20.10 Severity tasks ... 605
20.11 Elaboration system tasks ... 606
20.12 Assertion control system tasks ... 608
20.13 Sampled value system functions ... 614
20.14 Coverage system functions ... 615
20.15 Probabilistic distribution functions ... 615
20.16 Stochastic analysis tasks and functions ... 617
20.17 Programmable logic array modeling system tasks ... 619
20.18 Miscellaneous tasks and functions ... 623

21. Input/output system tasks and system functions ... 624

21.1 General ... 624
21.2 Display system tasks ... 624
21.3 File input/output system tasks and system functions ... 635
21.4 Loading memory array data from a file ... 645
21.5 Writing memory array data to a file ... 649
21.6 Command line input ... 650
21.7 Value change dump (VCD) files ... 653

22. Compiler directives ... 674

22.1 General ... 674
22.2 Overview ... 674
22.3 `resetall ... 674
22.4 `include ... 675
22.5 `define, `undef, and `undefineall ... 675
22.6 `ifdef, `else, `elsif, `endif, `ifndef ... 681
22.7 `timescale ... 684
22.8 `default_nettype ... 685
22.9 `unconnected_drive and `nounconnected_drive ... 686

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

22.10 `celldefine and `endcelldefine ... 686
22.11 `pragma ... 686
22.12 `line ... 687
22.13 `__FILE__ and `__LINE__ ... 688
22.14 `begin_keywords, `end_keywords ... 689

Part Two: Hierarchy Constructs

23. Modules and hierarchy ... 696

23.1 General ... 696
23.2 Module definitions ... 696
23.3 Module instances (hierarchy) ... 708
23.4 Nested modules ... 719
23.5 Extern modules ... 720
23.6 Hierarchical names ... 721
23.7 Member selects and hierarchical names ... 725
23.8 Upwards name referencing ... 727
23.9 Scope rules ... 729
23.10 Overriding module parameters ... 731
23.11 Binding auxiliary code to scopes or instances ... 738

24. Programs ... 742

24.1 General ... 742
24.2 Overview ... 742
24.3 The program construct ... 742
24.4 Eliminating testbench races ... 746
24.5 Blocking tasks in cycle/event mode ... 746
24.6 Programwide space and anonymous programs ... 747
24.7 Program control tasks ... 747

25. Interfaces ... 748

25.1 General ... 748
25.2 Overview ... 748
25.3 Interface syntax ... 749
25.4 Ports in interfaces ... 753
25.5 Modports ... 754
25.6 Interfaces and specify blocks ... 760
25.7 Tasks and functions in interfaces ... 761
25.8 Parameterized interfaces ... 767
25.9 Virtual interfaces ... 769
25.10 Access to interface objects ... 774

26. Packages ... 775

26.1 General ... 775
26.2 Package declarations ... 775

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

26.3 Referencing data in packages ... 776
26.4 Using packages in module headers ... 780
26.5 Search order rules ... 781
26.6 Exporting imported names from packages ... 783
26.7 The std built-in package ... 784

27. Generate constructs ... 786

27.1 General ... 786
27.2 Overview ... 786
27.3 Generate construct syntax ... 786
27.4 Loop generate constructs ... 788
27.5 Conditional generate constructs ... 792
27.6 External names for unnamed generate blocks ... 795

28. Gate-level and switch-level modeling ... 797

28.1 General ... 797
28.2 Overview ... 797
28.3 Gate and switch declaration syntax ... 797
28.4 and, nand, nor, or, xor, and xnor gates ... 803
28.5 buf and not gates ... 804
28.6 bufif1, bufif0, notif1, and notif0 gates ... 805
28.7 MOS switches ... 806
28.8 Bidirectional pass switches ... 807
28.9 CMOS switches ... 808
28.10 pullup and pulldown sources ... 809
28.11 Logic strength modeling ... 809
28.12 Strengths and values of combined signals ... 811
28.13 Strength reduction by nonresistive devices ... 823
28.14 Strength reduction by resistive devices ... 823
28.15 Strengths of net types ... 823
28.16 Gate and net delays ... 824

29. User-defined primitives ... 828

29.1 General ... 828
29.2 Overview ... 828
29.3 UDP definition ... 828
29.4 Combinational UDPs ... 832
29.5 Level-sensitive sequential UDPs ... 833
29.6 Edge-sensitive sequential UDPs ... 833
29.7 Sequential UDP initialization ... 834
29.8 UDP instances ... 836
29.9 Mixing level-sensitive and edge-sensitive descriptions ... 837
29.10 Level-sensitive dominance ... 838

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

30. Specify blocks ... 839

30.1 General ... 839
30.2 Overview ... 839
30.3 Specify block declaration ... 839
30.4 Module path declarations ... 840
30.5 Assigning delays to module paths ... 849
30.6 Mixing module path delays and distributed delays ... 853
30.7 Detailed control of pulse filtering behavior ... 854

31. Timing checks ... 863

31.1 General ... 863
31.2 Overview ... 863
31.3 Timing checks using a stability window ... 866
31.4 Timing checks for clock and control signals ... 873
31.5 Edge-control specifiers ... 882
31.6 Notifiers: user-defined responses to timing violations ... 883
31.7 Enabling timing checks with conditioned events ... 885
31.8 Vector signals in timing checks ... 886
31.9 Negative timing checks ... 887

32. Backannotation using the standard delay format ... 892

32.1 General ... 892
32.2 Overview ... 892
32.3 The SDF annotator ... 892
32.4 Mapping of SDF constructs to SystemVerilog ... 892
32.5 Multiple annotations ... 897
32.6 Multiple SDF files ... 898
32.7 Pulse limit annotation ... 898
32.8 SDF to SystemVerilog delay value mapping ... 899
32.9 Loading timing data from an SDF file ... 900

33. Configuring the contents of a design ... 902

33.1 General ... 902
33.2 Overview ... 902
33.3 Libraries ... 903
33.4 Configurations ... 905
33.5 Using libraries and configs ... 911
33.6 Configuration examples ... 912
33.7 Displaying library binding information ... 914
33.8 Library mapping examples ... 914

34. Protected envelopes ... 917

34.1 General ... 917
34.2 Overview ... 917

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

34.3 Processing protected envelopes ... 917
34.4 Protect pragma directives ... 919
34.5 Protect pragma keywords ... 921

Part Three: Application Programming Interfaces

35. Direct programming interface ... 938

35.1 General ... 938
35.2 Overview ... 938
35.3 Two layers of DPI ... 939
35.4 Global name space of imported and exported functions ... 940
35.5 Imported tasks and functions ... 941
35.6 Calling imported functions ... 948
35.7 Exported functions ... 950
35.8 Exported tasks ... 951
35.9 Disabling DPI tasks and functions ... 951

36. Programming language interface (PLI/VPI) overview ... 953

36.1 General ... 953
36.2 PLI purpose and history ... 953
36.3 User-defined system task and system function names ... 954
36.4 User-defined system task and system function arguments ... 955
36.5 User-defined system task and system function types ... 955
36.6 User-supplied PLI applications ... 955
36.7 PLI include files ... 955
36.8 VPI sizetf, compiletf, and calltf routines ... 955
36.9 PLI mechanism ... 956
36.10 VPI access to SystemVerilog objects and simulation objects ... 958
36.11 List of VPI routines by functional category ... 959
36.12 VPI backwards compatibility features and limitations ... 961

37. VPI object model diagrams ... 966

37.1 General ... 966
37.2 VPI Handles ... 966
37.3 VPI object classifications ... 967
37.4 Key to data model diagrams ... 973
37.5 Module ... 976
37.6 Interface ... 977
37.7 Modport ... 977
37.8 Interface task or function declaration ... 977
37.9 Program ... 978
37.10 Instance ... 979
37.11 Instance arrays ... 981
37.12 Scope ... 982
37.13 IO declaration ... 983
37.14 Ports ... 984

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

37.15 Reference objects ... 985
37.16 Nets ... 987
37.17 Variables ... 991
37.18 Packed array variables ... 994
37.19 Variable select ... 995
37.20 Memory ... 996
37.21 Variable drivers and loads ... 996
37.22 Object Range ... 997
37.23 Typespec ... 998
37.24 Structures and unions ... 1000
37.25 Named events ... 1001
37.26 Parameter, spec param, def param, param assign ... 1002
37.27 Virtual interface ... 1003
37.28 Interface typespec ... 1005
37.29 Class definition ... 1006
37.30 Class typespec ... 1007
37.31 Class variables and class objects ... 1009
37.32 Constraint, constraint ordering, distribution ... 1011
37.33 Primitive, prim term ... 1012
37.34 UDP ... 1013
37.35 Intermodule path ... 1013
37.36 Constraint expression ... 1014
37.37 Module path, path term ... 1015
37.38 Timing check ... 1016
37.39 Task and function declaration ... 1017
37.40 Task and function call ... 1018
37.41 Frames ... 1020
37.42 Threads ... 1021
37.43 Delay terminals ... 1021
37.44 Net drivers and loads ... 1022
37.45 Continuous assignment ... 1023
37.46 Clocking block ... 1024
37.47 Assertion ... 1025
37.48 Concurrent assertions ... 1026
37.49 Property declaration ... 1027
37.50 Property specification ... 1028
37.51 Sequence declaration ... 1029
37.52 Sequence expression ... 1030
37.53 Immediate assertions ... 1031
37.54 Multiclock sequence expression ... 1032
37.55 Let ... 1032
37.56 Simple expressions ... 1033
37.57 Expressions ... 1034
37.58 Atomic statement ... 1037
37.59 Dynamic prefixing ... 1038
37.60 Event statement ... 1039
37.61 Process ... 1039
37.62 Assignment ... 1040
37.63 Event control ... 1040
37.64 While, repeat ... 1041

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

37.65 Waits 1041
37.66 Delay control 1041
37.67 Repeat control 1042
37.68 Forever 1042
37.69 If, if–else 1042
37.70 Case, pattern 1043
37.71 Expect 1044
37.72 For 1044
37.73 Do-while, foreach 1044
37.74 Alias statement 1045
37.75 Disables 1045
37.76 Return statement 1045
37.77 Assign statement, deassign, force, release 1046
37.78 Callback 1046
37.79 Time queue 1047
37.80 Active time format 1047
37.81 Attribute 1048
37.82 Iterator 1049
37.83 Generates 1050

38. VPI routine definitions 1052

38.1 General 1052
38.2 vpi_chk_error() 1052
38.3 vpi_compare_objects() 1053
38.4 vpi_control() 1055
38.5 vpi_flush() 1056
38.6 vpi_get() 1056
38.7 vpi_get64() 1057
38.8 vpi_get_cb_info() 1057
38.9 vpi_get_data() 1058
38.10 vpi_get_delays() 1059
38.11 vpi_get_str() 1061
38.12 vpi_get_systf_info() 1062
38.13 vpi_get_time() 1063
38.14 vpi_get_userdata() 1064
38.15 vpi_get_value() 1064
38.16 vpi_get_value_array() 1070
38.17 vpi_get_vlog_info() 1074
38.18 vpi_handle() 1075
38.19 vpi_handle_by_index() 1076
38.20 vpi_handle_by_multi_index() 1076
38.21 vpi_handle_by_name() 1077
38.22 vpi_handle_multi() 1078
38.23 vpi_iterate() 1078
38.24 vpi_mcd_close() 1079
38.25 vpi_mcd_flush() 1080
38.26 vpi_mcd_name() 1080
38.27 vpi_mcd_open() 1081

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

38.28 vpi_mcd_printf() ... 1082
38.29 vpi_mcd_vprintf() ... 1083
38.30 vpi_printf() ... 1083
38.31 vpi_put_data() ... 1084
38.32 vpi_put_delays() ... 1086
38.33 vpi_put_userdata() ... 1089
38.34 vpi_put_value() ... 1089
38.35 vpi_put_value_array() ... 1092
38.36 vpi_register_cb() ... 1096
38.37 vpi_register_systf() ... 1104
38.38 vpi_release_handle() ... 1108
38.39 vpi_remove_cb() ... 1108
38.40 vpi_scan() ... 1109
38.41 vpi_vprintf() ... 1110

39. Assertion API ... 1111

39.1 General ... 1111
39.2 Overview ... 1111
39.3 Static information ... 1111
39.4 Dynamic information ... 1112
39.5 Control functions ... 1116

40. Code coverage control and API ... 1120

40.1 General ... 1120
40.2 Overview ... 1120
40.3 SystemVerilog real-time coverage access ... 1121
40.4 FSM recognition ... 1126
40.5 VPI coverage extensions ... 1129

41. Data read API ... 1134

Part Four: Annexes

Annex A (normative) Formal syntax ... 1136

Annex B (normative) Keywords ... 1182

Annex C (normative) Deprecation ... 1184

Annex D (informative) Optional system tasks and system functions ... 1188

Annex E (informative) Optional compiler directives ... 1195

Annex F (normative) Formal semantics of concurrent assertions ... 1197

Annex G (normative) Std package ... 1220

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

Annex H (normative) DPI C layer 1222

Annex I (normative) svdpi.h 1255

Annex J (normative) Inclusion of foreign language code 1264

Annex K (normative) vpi_user.h 1268

Annex L (normative) vpi_compatibility.h 1285

Annex M (normative) sv_vpi_user.h 1288

Annex N (normative) Algorithm for probabilistic distribution functions 1298

Annex O (informative) Encryption/decryption flow 1306

Annex P (informative) Glossary 1310

Annex Q (informative) Bibliography 1313

Annex R (informative) Participants 1314

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

List of figures

Figure 4-1—Event scheduling regions ... 64
Figure 6-1—Simulation values of a trireg and its driver ... 89
Figure 6-2—Simulation results of a capacitive network ... 90
Figure 6-3—Simulation results of charge sharing ... 91
Figure 7-1—VInt type with packed qualifier ... 144
Figure 7-2—Instr type with packed qualifier ... 145
Figure 9-1—Intra-assignment repeat event control utilizing a clock edge ... 224
Figure 14-1—Sample and drive times including skew with respect to the positive edge of the clock ... 339
Figure 16-1—Sampling a variable in a simulation time step ... 376
Figure 16-2—Concatenation of sequences ... 381
Figure 16-3—Value change expressions ... 396
Figure 16-4—Future value change ... 401
Figure 16-5—ANDing (and) two sequences ... 403
Figure 16-6—ANDing (and) two sequences, including a time range ... 404
Figure 16-7—ANDing (and) two Boolean expressions ... 404
Figure 16-8—Intersecting two sequences ... 405
Figure 16-9—ORing (or) two Boolean expressions ... 406
Figure 16-10—ORing (or) two sequences ... 407
Figure 16-11—ORing (or) two sequences, including a time range ... 407
Figure 16-12—Match with throughout restriction fails ... 410
Figure 16-13—Match with throughout restriction succeeds ... 410
Figure 16-14—Conditional sequence matching ... 427
Figure 16-15—Conditional sequences ... 428
Figure 16-16—Results without the condition ... 428
Figure 16-17—Clocking blocks and concurrent assertion ... 483
Figure 17-1—Nondeterministic free checker variable ... 495
Figure 18-1—Example of randc ... 509
Figure 18-2—Global constraints ... 518
Figure 18-3—Truth tables for conjunction, disjunction, and negation rules ... 523
Figure 21-1—Creating the 4-state VCD file ... 653
Figure 21-2—Creating the extended VCD file ... 663
Figure 23-1—Hierarchy in a model ... 723
Figure 23-2—Scopes available to upward name referencing ... 730
Figure 28-1—Schematic diagram of interconnections in array of instances ... 803
Figure 28-2—Scale of strengths ... 811
Figure 28-3—Combining unequal strengths ... 811
Figure 28-4—Combination of signals of equal strength and opposite values ... 812
Figure 28-5—Weak x signal strength ... 812
Figure 28-6—Bufifs with control inputs of x ... 813

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Figure 28-7—Strong H range of values ... 813
Figure 28-8—Strong L range of values ... 813
Figure 28-9—Combined signals of ambiguous strength ... 814
Figure 28-10—Range of strengths for an unknown signal ... 814
Figure 28-11—Ambiguous strengths from switch networks ... 814
Figure 28-12—Range of two strengths of a defined value ... 815
Figure 28-13—Range of three strengths of a defined value ... 815
Figure 28-14—Unknown value with a range of strengths ... 815
Figure 28-15—Strong X range ... 816
Figure 28-16—Ambiguous strength from gates ... 816
Figure 28-17—Ambiguous strength signal from a gate ... 816
Figure 28-18—Weak 0 ... 817
Figure 28-19—Ambiguous strength in combined gate signals ... 817
Figure 28-20—Elimination of strength levels ... 818
Figure 28-21—Result showing a range and the elimination of strength levels of two values ... 819
Figure 28-22—Result showing a range and the elimination of strength levels of one value ... 820
Figure 28-23—A range of both values ... 820
Figure 28-24—Wired logic with unambiguous strength signals ... 821
Figure 28-25—Wired logic and ambiguous strengths ... 822
Figure 28-26—Trireg net with capacitance ... 827
Figure 29-1—Module schematic and simulation times of initial value propagation ... 836
Figure 30-1—Module path delays ... 841
Figure 30-2—Difference between parallel and full connection paths ... 847
Figure 30-3—Module path delays longer than distributed delays ... 853
Figure 30-4—Module path delays shorter than distributed delays ... 854
Figure 30-5—Example of pulse filtering ... 854
Figure 30-6—On-detect versus on-event ... 857
Figure 30-7—Current event cancellation problem and correction ... 859
Figure 30-8—NAND gate with nearly simultaneous input switching where one event is scheduled prior to another that has not matured ... 860
Figure 30-9—NAND gate with nearly simultaneous input switching with output event scheduled at same time ... 861
Figure 31-1—Sample $timeskew ... 875
Figure 31-2—Sample $timeskew with remain_active_flag set ... 876
Figure 31-3—Sample $fullskew ... 878
Figure 31-4—Data constraint interval, positive setup/hold ... 887
Figure 31-5—Data constraint interval, negative setup/hold ... 888
Figure 31-6—Timing check violation windows ... 891
Figure 37-1—Example of object relationships diagram ... 968
Figure 37-2—Accessing a class of objects using tags ... 969
Figure 38-1—s_vpi_error_info structure definition ... 1053

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Figure 38-2—s_cb_data structure definition 1058
Figure 38-3—s_vpi_delay structure definition 1059
Figure 38-4—s_vpi_time structure definition 1059
Figure 38-5—s_vpi_systf_data structure definition 1062
Figure 38-6—s_vpi_time structure definition 1063
Figure 38-7—s_vpi_value structure definition 1065
Figure 38-8—s_vpi_vecval structure definition 1065
Figure 38-9—s_vpi_strengthval structure definition 1065
Figure 38-10—s_vpi_vlog_info structure definition 1074
Figure 38-11—s_vpi_delay structure definition 1087
Figure 38-12—s_vpi_time structure definition 1087
Figure 38-13—s_vpi_value structure definition 1091
Figure 38-14—s_vpi_time structure definition 1091
Figure 38-15—s_vpi_vecval structure definition 1092
Figure 38-16—s_vpi_strengthval structure definition 1092
Figure 38-17—s_cb_data structure definition 1096
Figure 38-18—s_vpi_systf_data structure definition 1105
Figure 39-1—Assertions with global clocking future sampled value functions 1116
Figure 40-1—Hierarchical instance example 1124
Figure 40-2—FSM specified with pragmas 1129

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

List of tables

Table 3-1—Time unit strings ... 55
Table 6-2—Truth table for wire and tri nets ... 87
Table 6-3—Truth table for wand and triand nets ... 88
Table 6-4—Truth table for wor and trior nets ... 88
Table 6-5—Truth table for tri0 net ... 92
Table 6-6—Truth table for tri1 net ... 92
Table 6-7—Default variable initial values ... 102
Table 6-8—Integer data types ... 104
Table 6-10—Enumeration element ranges ... 116
Table 6-11—Differences between specparams and parameters ... 125
Table 9-2—Detecting posedge and negedge ... 217
Table 9-3—Intra-assignment timing control equivalence ... 223
Table 11-2—Operator precedence and associativity ... 257
Table 11-3—Arithmetic operators defined ... 260
Table 11-4—Power operator rules ... 261
Table 11-5—Unary operators defined ... 261
Table 11-6—Examples of modulus and power operators ... 261
Table 11-7—Data type interpretation by arithmetic operators ... 262
Table 11-8—Definitions of relational operators ... 263
Table 11-9—Definitions of equality operators ... 264
Table 11-10—Wildcard equality and wildcard inequality operators ... 264
Table 11-11—Bitwise binary AND operator ... 266
Table 11-12—Bitwise binary OR operator ... 266
Table 11-13—Bitwise binary exclusive OR operator ... 267
Table 11-14—Bitwise binary exclusive NOR operator ... 267
Table 11-15—Bitwise unary negation operator ... 267
Table 11-16—Reduction unary AND operator ... 268
Table 11-17—Reduction unary OR operator ... 268
Table 11-18—Reduction unary exclusive OR operator ... 268
Table 11-19—Results of unary reduction operations ... 268
Table 11-20—Ambiguous condition results for conditional operator ... 270
Table 11-21—Bit lengths resulting from self-determined expressions ... 283
Table 16-2—Global clocking future sampled value functions ... 401
Table 16-3—Sequence and property operator precedence and associativity ... 423
Table 18-2—Ordered constraint c legal value probability ... 519
Table 18-3—rand_mode argument ... 533

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Table 18-4—constraint_mode argument ... 534
Table 19-2—Coverage options per syntactic level ... 580
Table 19-3—Coverage group type (static) options ... 581
Table 19-4—Coverage type options ... 582
Table 19-5—Predefined coverage methods ... 583
Table 20-2—$timeformat units_number arguments ... 595
Table 20-3—$timeformat default value for arguments ... 596
Table 20-4—SystemVerilog to C real math function cross-listing ... 603
Table 20-5—Values for control_type for assertion control tasks ... 609
Table 20-6—Values for assertion_type for assertion control tasks ... 609
Table 20-7—Values for directive_type for assertion control tasks ... 610
Table 20-8—VPI callbacks for assertion control tasks ... 613
Table 20-9—Types of queues of $q_type values ... 617
Table 20-10—Argument values for $q_exam system task ... 618
Table 20-11—Status code values ... 619
Table 20-12—PLA modeling system tasks ... 620
Table 21-2—Escape sequences for format specifications ... 626
Table 21-3—Format specifications for real numbers ... 628
Table 21-4—Logic value component of strength format ... 631
Table 21-5—Mnemonics for strength levels ... 631
Table 21-6—Explanation of strength formats ... 632
Table 21-7—Types for file descriptors ... 636
Table 21-8—$fscanf input field characters ... 641
Table 21-9—Rules for left-extending vector values ... 659
Table 21-10—How the VCD can shorten values ... 659
Table 21-11—Keyword commands ... 660
Table 21-12—VCD type mapping ... 672
Table 22-2—IEEE 1364-2001 additional reserved keywords ... 692
Table 22-3—IEEE 1364-2005 additional reserved keywords ... 692
Table 22-4—IEEE 1800-2005 additional reserved keywords ... 693
Table 22-5—IEEE 1800-2009 additional reserved keywords ... 693
Table 22-6—IEEE 1800-2012 additional reserved keywords ... 694
Table 28-2—Valid gate types for strength specifications ... 799
Table 28-3—Truth tables for multiple input logic gates ... 804
Table 28-4—Truth tables for multiple output logic gates ... 805
Table 28-5—Truth tables for three-state logic gates ... 806
Table 28-6—Truth tables for MOS switches ... 807
Table 28-7—Strength levels for scalar net signal values ... 810

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Table 28-8—Strength reduction rules ... 823
Table 28-9—Rules for propagation delays ... 824
Table 29-2—Initial statements in UDPs and modules ... 834
Table 29-3—Mixing of level-sensitive and edge-sensitive entries ... 838
Table 30-2—Associating path delay expressions with transitions ... 850
Table 30-3—Calculating delays for x transitions ... 852
Table 31-2—$hold arguments ... 867
Table 31-3—$setuphold arguments ... 868
Table 31-4—$removal arguments ... 870
Table 31-5—$recovery arguments ... 871
Table 31-6—$recrem arguments ... 872
Table 31-7—$skew arguments ... 874
Table 31-8—$timeskew arguments ... 875
Table 31-9—$fullskew arguments ... 877
Table 31-10—$width arguments ... 879
Table 31-11—$period arguments ... 880
Table 31-12—$nochange arguments ... 881
Table 31-13—Notifier value responses to timing violations ... 883
Table 32-2—Mapping of SDF timing check constructs to SystemVerilog ... 894
Table 32-3—SDF annotation of interconnect delays ... 896
Table 32-4—SDF to SystemVerilog delay value mapping ... 899
Table 32-5—mtm_spec argument ... 900
Table 32-6—scale_type argument ... 901
Table 34-2—Encoding algorithm identifiers ... 924
Table 34-3—Encryption algorithm identifiers ... 926
Table 34-4—Message digest algorithm identifiers ... 931
Table 36-2—VPI routines for system task or system function callbacks ... 960
Table 36-3—VPI routines for traversing SystemVerilog hierarchy ... 960
Table 36-4—VPI routines for accessing properties of objects ... 960
Table 36-5—VPI routines for accessing objects from properties ... 960
Table 36-6—VPI routines for delay processing ... 960
Table 36-7—VPI routines for logic and strength value processing ... 960
Table 36-8—VPI routines for simulation time processing ... 961
Table 36-9—VPI routines for miscellaneous utilities ... 961
Table 36-10—Summary of VPI incompatibilities across versions ... 962
Table 38-2—Size of the s_vpi_delay->da array ... 1060
Table 38-3—Return value field of the s_vpi_value structure union ... 1066
Table 38-4—Size of the s_vpi_delay->da array ... 1087

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

Table 38-5—Value format field of cb_data_p->value->format 1098
Table 38-6—cbStmt callbacks 1100
Table 40-2—Instance coverage permutations 1123
Table 40-3—Assertion coverage results 1131
Table B.1—Reserved keywords 1182
Table D.1—Argument return value for $countdriver function 1189
Table H.1—Mapping data types 1227
Table N.1—SystemVerilog to C function cross-listing 1298

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

List of syntax excerpts

Syntax 5-1—Syntax for system tasks and system functions (excerpt from Annex A) ... 71
Syntax 5-2—Syntax for integer and real numbers (excerpt from Annex A) ... 72
Syntax 5-3—Syntax for attributes (excerpt from Annex A) ... 80
Syntax 6-1—Syntax for net type declarations (excerpt from Annex A) ... 92
Syntax 6-2—Syntax for net declarations (excerpt from Annex A) ... 97
Syntax 6-3—Syntax for variable declarations (excerpt from Annex A) ... 101
Syntax 6-4—User-defined types (excerpt from Annex A) ... 112
Syntax 6-5—Enumerated types (excerpt from Annex A) ... 114
Syntax 6-6—Parameter declaration syntax (excerpt from Annex A) ... 120
Syntax 6-7—Casting (excerpt from Annex A) ... 132
Syntax 7-1—Structure declaration syntax (excerpt from Annex A) ... 139
Syntax 7-2—Union declaration syntax (excerpt from Annex A) ... 142
Syntax 7-3—Dynamic array new constructor syntax (excerpt from Annex A) ... 150
Syntax 7-4—Declaration of queue dimension (excerpt from Annex A) ... 161
Syntax 7-5—Array method call syntax (not in Annex A) ... 165
Syntax 8-1—Class syntax (excerpt from Annex A) ... 172
Syntax 8-2—Calling a constructor (excerpt from Annex A) ... 176
Syntax 8-3—Class syntax (excerpt from Annex A) ... 195
Syntax 9-1—Syntax for structured procedures (excerpt from Annex A) ... 205
Syntax 9-2—Syntax for sequential block (excerpt from Annex A) ... 210
Syntax 9-3—Syntax for parallel block (excerpt from Annex A) ... 211
Syntax 9-4—Delay and event control syntax (excerpt from Annex A) ... 216
Syntax 9-5—Syntax for wait statement (excerpt from Annex A) ... 221
Syntax 9-6—Syntax for intra-assignment delay and event control (excerpt from Annex A) ... 222
Syntax 9-7—Syntax for process control statements (excerpt from Annex A) ... 225
Syntax 10-1—Syntax for continuous assignment (excerpt from Annex A) ... 233
Syntax 10-2—Blocking assignment syntax (excerpt from Annex A) ... 237
Syntax 10-3—Nonblocking assignment syntax (excerpt from Annex A) ... 238
Syntax 10-4—Syntax for procedural continuous assignments (excerpt from Annex A) ... 241
Syntax 10-5—Assignment patterns syntax (excerpt from Annex A) ... 246
Syntax 10-6—Syntax for net aliasing (excerpt from Annex A) ... 252
Syntax 11-1—Operator syntax (excerpt from Annex A) ... 255
Syntax 11-2—Conditional operator syntax (excerpt from Annex A) ... 269
Syntax 11-3—Inside expression syntax (excerpt from Annex A) ... 273
Syntax 11-4—Streaming concatenation syntax (excerpt from Annex A) ... 274
Syntax 11-5—With expression syntax (excerpt from Annex A) ... 277
Syntax 11-6—Tagged union syntax (excerpt from Annex A) ... 287
Syntax 11-7—Syntax for min:typ:max expression (excerpt from Annex A) ... 291
Syntax 11-8—Let syntax (excerpt from Annex A) ... 292

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

Syntax 12-1—Procedural statement syntax (excerpt from Annex A) ... 299
Syntax 12-2—Syntax for if–else statement (excerpt from Annex A) ... 299
Syntax 12-3—Syntax for case statements (excerpt from Annex A) ... 304
Syntax 12-4—Pattern syntax (excerpt from Annex A) ... 309
Syntax 12-5—Loop statement syntax (excerpt from Annex A) ... 313
Syntax 12-6—Jump statement syntax (excerpt from Annex A) ... 317
Syntax 13-1—Task syntax (excerpt from Annex A) ... 320
Syntax 13-2—Function syntax (excerpt from Annex A) ... 324
Syntax 13-3—Task or function call syntax (excerpt from Annex A) ... 330
Syntax 14-1—Clocking block syntax (excerpt from Annex A) ... 337
Syntax 14-2—Cycle delay syntax (excerpt from Annex A) ... 343
Syntax 14-3—Default clocking syntax (excerpt from Annex A) ... 344
Syntax 14-4—Global clocking syntax (excerpt from Annex A) ... 345
Syntax 14-5—Synchronous drive syntax (excerpt from Annex A) ... 350
Syntax 15-1—Event trigger syntax (excerpt from Annex A) ... 360
Syntax 15-2—Wait_order event sequencing syntax (excerpt from Annex A) ... 361
Syntax 16-1—Immediate assertion syntax (excerpt from Annex A) ... 365
Syntax 16-2—Deferred immediate assertion syntax (excerpt from Annex A) ... 368
Syntax 16-3—Sequence syntax (excerpt from Annex A) ... 379
Syntax 16-4—Sequence concatenation syntax (excerpt from Annex A) ... 380
Syntax 16-5—Sequence declaration syntax (excerpt from Annex A) ... 383
Syntax 16-6—Sequence repetition syntax (excerpt from Annex A) ... 391
Syntax 16-7—And operator syntax (excerpt from Annex A) ... 402
Syntax 16-8—Intersect operator syntax (excerpt from Annex A) ... 404
Syntax 16-9—Or operator syntax (excerpt from Annex A) ... 405
Syntax 16-10—First_match operator syntax (excerpt from Annex A) ... 408
Syntax 16-11—Throughout construct syntax (excerpt from Annex A) ... 409
Syntax 16-12—Within construct syntax (excerpt from Annex A) ... 411
Syntax 16-13—Assertion variable declaration syntax (excerpt from Annex A) ... 413
Syntax 16-14—Variable assignment syntax (excerpt from Annex A) ... 414
Syntax 16-15—Subroutine call in sequence syntax (excerpt from Annex A) ... 419
Syntax 16-16—Property construct syntax (excerpt from Annex A) ... 421
Syntax 16-17—Implication syntax (excerpt from Annex A) ... 425
Syntax 16-18—Followed-by syntax (excerpt from Annex A) ... 429
Syntax 16-19—Property statement case syntax (excerpt from Annex A) ... 439
Syntax 16-20—Concurrent assert construct syntax (excerpt from Annex A) ... 457
Syntax 16-21—Default clocking and default disable syntax (excerpt from Annex A) ... 473
Syntax 16-22—Expect statement syntax (excerpt from Annex A) ... 481
Syntax 17-1—Checker declaration syntax (excerpt from Annex A) ... 485
Syntax 17-2—Checker instantiation syntax (excerpt from Annex A) ... 488

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Syntax 18-1—Random variable declaration syntax (excerpt from Annex A) ... 506
Syntax 18-2—Constraint syntax (excerpt from Annex A) ... 510
Syntax 18-3—Constraint distribution syntax (excerpt from Annex A) ... 512
Syntax 18-4—Uniqueness constraint syntax (excerpt from Annex A) ... 513
Syntax 18-5—Constraint implication syntax (excerpt from Annex A) ... 514
Syntax 18-6—If–else constraint syntax (excerpt from Annex A) ... 515
Syntax 18-7—Foreach iterative constraint syntax (excerpt from Annex A) ... 516
Syntax 18-8—Solve...before constraint ordering syntax (excerpt from Annex A) ... 520
Syntax 18-9—Static constraint syntax (excerpt from Annex A) ... 520
Syntax 18-10—Inline constraint syntax (excerpt from Annex A) ... 530
Syntax 18-11—Scope randomize function syntax (not in Annex A) ... 537
Syntax 18-12—Randcase syntax (excerpt from Annex A) ... 543
Syntax 18-13—Randsequence syntax (excerpt from Annex A) ... 544
Syntax 18-14—Random production weights syntax (excerpt from Annex A) ... 546
Syntax 18-15—If–else conditional random production syntax (excerpt from Annex A) ... 546
Syntax 18-16—Case random production syntax (excerpt from Annex A) ... 547
Syntax 18-17—Repeat random production syntax (excerpt from Annex A) ... 547
Syntax 18-18—Rand join random production syntax (excerpt from Annex A) ... 548
Syntax 18-19—Random production syntax (excerpt from Annex A) ... 550
Syntax 19-1—Covergroup syntax (excerpt from Annex A) ... 554
Syntax 19-2—Coverage point syntax (excerpt from Annex A) ... 559
Syntax 19-3—Transition bin syntax (excerpt from Annex A) ... 563
Syntax 19-4—Cross coverage syntax (excerpt from Annex A) ... 570
Syntax 20-1—Syntax for simulation control tasks (not in Annex A) ... 592
Syntax 20-2—Syntax for time system functions (not in Annex A) ... 592
Syntax 20-3—Syntax for $printtimescale (not in Annex A) ... 594
Syntax 20-4—Syntax for $timeformat (not in Annex A) ... 595
Syntax 20-5—Type name function syntax (not in Annex A) ... 598
Syntax 20-6—Size function syntax (not in Annex A) ... 599
Syntax 20-7—Range function syntax (not in Annex A) ... 600
Syntax 20-8—Array querying function syntax (not in Annex A) ... 601
Syntax 20-9—Bit vector system function syntax (not in Annex A) ... 604
Syntax 20-10—Severity system task syntax (not in Annex A) ... 605
Syntax 20-11—Elaboration system task syntax (excerpt from Annex A) ... 606
Syntax 20-12—Assertion control syntax (not in Annex A) ... 608
Syntax 20-13—Sampled value system function syntax (not in Annex A) ... 615
Syntax 20-14—Syntax for $random (not in Annex A) ... 616
Syntax 20-15—Syntax for probabilistic distribution functions (not in Annex A) ... 616
Syntax 20-16—Syntax for PLA modeling system task (not in Annex A) ... 619
Syntax 20-17—$system function syntax (not in Annex A) ... 623

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Syntax 21-1—Syntax for $display and $write system tasks (not in Annex A) 625
Syntax 21-2—Syntax for $strobe system tasks (not in Annex A) 634
Syntax 21-3—Syntax for $monitor system tasks (not in Annex A) 634
Syntax 21-4—Syntax for $fopen and $fclose system tasks (not in Annex A) 635
Syntax 21-5—Syntax for file output system tasks (not in Annex A) 637
Syntax 21-6—Syntax for formatting data tasks (not in Annex A) 638
Syntax 21-7—Syntax for file read system functions (not in Annex A) 639
Syntax 21-8—Syntax for file positioning system functions (not in Annex A) 643
Syntax 21-9—Syntax for file flush system task (not in Annex A) 644
Syntax 21-10—Syntax for file I/O error detection system function (not in Annex A) 645
Syntax 21-11—Syntax for end-of-file file detection system function (not in Annex A) 645
Syntax 21-12—Syntax for memory load system tasks (not in Annex A) 645
Syntax 21-13—$writemem system task syntax (not in Annex A) 649
Syntax 21-14—Syntax for $dumpfile task (not in Annex A) 653
Syntax 21-15—Syntax for $dumpvars task (not in Annex A) 654
Syntax 21-16—Syntax for $dumpoff and $dumpon tasks (not in Annex A) 655
Syntax 21-17—Syntax for $dumpall task (not in Annex A) 655
Syntax 21-18—Syntax for $dumplimit task (not in Annex A) 656
Syntax 21-19—Syntax for $dumpflush task (not in Annex A) 656
Syntax 21-20—Syntax for output 4-state VCD file (not in Annex A) 658
Syntax 21-21—Syntax for $dumpports task (not in Annex A) 663
Syntax 21-22—Syntax for $dumpportsoff and $dumpportson system tasks (not in Annex A) 664
Syntax 21-23—Syntax for $dumpportsall system task (not in Annex A) 665
Syntax 21-24—Syntax for $dumpportslimit system task (not in Annex A) 665
Syntax 21-25—Syntax for $dumpportsflush system task (not in Annex A) 665
Syntax 21-26—Syntax for $vcdclose keyword (not in Annex A) 666
Syntax 21-27—Syntax for output extended VCD file (not in Annex A) 667
Syntax 21-28—Syntax for extended VCD node information (not in Annex A) 668
Syntax 21-29—Syntax for value change section (not in Annex A) 669
Syntax 22-1—Syntax for include compiler directive (not in Annex A) 675
Syntax 22-2—Syntax for text macro definition (not in Annex A) 676
Syntax 22-3—Syntax for text macro usage (not in Annex A) 677
Syntax 22-4—Syntax for undef compiler directive (not in Annex A) 681
Syntax 22-5—Syntax for conditional compilation directives (not in Annex A) 681
Syntax 22-6—Syntax for timescale compiler directive (not in Annex A) 684
Syntax 22-7—Syntax for default_nettype compiler directive (not in Annex A) 686
Syntax 22-8—Syntax for pragma compiler directive (not in Annex A) 687
Syntax 22-9—Syntax for line compiler directive (not in Annex A) 688
Syntax 22-10—Syntax for begin_keywords and end_keywords compiler directives (not in Annex A) 689
Syntax 23-1—Module declaration syntax (excerpt from Annex A) 697

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Syntax 23-2—Non-ANSI style module header declaration syntax (excerpt from Annex A) 698
Syntax 23-3—Non-ANSI style port declaration syntax (excerpt from Annex A)..... 699
Syntax 23-4—ANSI style list_of_port_declarations syntax (excerpt from Annex A)..... 702
Syntax 23-5—Module item syntax (excerpt from Annex A) 708
Syntax 23-6—Module instance syntax (excerpt from Annex A) 709
Syntax 23-7—Syntax for hierarchical path names (excerpt from Annex A)..... 722
Syntax 23-8—Syntax for upward name referencing (not in Annex A)..... 727
Syntax 23-9—Bind construct syntax (excerpt from Annex A) 739
Syntax 24-1—Program declaration syntax (excerpt from Annex A) 743
Syntax 25-1—Interface syntax (excerpt from Annex A)..... 750
Syntax 25-2—Modport clocking declaration syntax (excerpt from Annex A) 759
Syntax 25-3—Virtual interface declaration syntax (excerpt from Annex A)..... 769
Syntax 26-1—Package declaration syntax (excerpt from Annex A)..... 776
Syntax 26-2—Package import syntax (excerpt from Annex A)..... 777
Syntax 26-3—Package import in header syntax (excerpt from Annex A) 781
Syntax 26-4—Package export syntax (excerpt from Annex A) 783
Syntax 26-5—Std package import syntax (not in Annex A) 785
Syntax 27-1—Syntax for generate constructs (excerpt from Annex A)..... 788
Syntax 28-1—Syntax for gate instantiation (excerpt from Annex A) 798
Syntax 29-1—Syntax for UDPs (excerpt from Annex A)..... 829
Syntax 29-2—Syntax for UDP instances (excerpt from Annex A)..... 836
Syntax 30-1—Syntax for specify block (excerpt from Annex A) 839
Syntax 30-2—Syntax for module path declaration (excerpt from Annex A)..... 840
Syntax 30-3—Syntax for simple module path (excerpt from Annex A)..... 841
Syntax 30-4—Syntax for edge-sensitive path declaration (excerpt from Annex A)..... 842
Syntax 30-5—Syntax for state-dependent paths (excerpt from Annex A)..... 843
Syntax 30-6—Syntax for path delay value (excerpt from Annex A) 850
Syntax 30-7—Syntax for PATHPULSE$ pulse control (excerpt from Annex A) 855
Syntax 30-8—Syntax for pulse style declarations (excerpt from Annex A) 857
Syntax 30-9—Syntax for showcancelled declarations (excerpt from Annex A)..... 858
Syntax 31-1—Syntax for system timing checks (excerpt from Annex A) 864
Syntax 31-2—Syntax for time check conditions and timing check events (excerpt from Annex A)..... 865
Syntax 31-3—Syntax for $setup (excerpt from Annex A) 866
Syntax 31-4—Syntax for $hold (excerpt from Annex A) 867
Syntax 31-5—Syntax for $setuphold (excerpt from Annex A)..... 868
Syntax 31-6—Syntax for $removal (excerpt from Annex A) 870
Syntax 31-7—Syntax for $recovery (excerpt from Annex A)..... 870
Syntax 31-8—Syntax for $recrem (excerpt from Annex A) 871
Syntax 31-9—Syntax for $skew (excerpt from Annex A) 873
Syntax 31-10—Syntax for $timeskew (excerpt from Annex A) 874

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Syntax 31-11—Syntax for $fullskew (excerpt from Annex A) ... 877
Syntax 31-12—Syntax for $width (excerpt from Annex A) ... 879
Syntax 31-13—Syntax for $period (excerpt from Annex A) ... 880
Syntax 31-14—Syntax for $nochange (excerpt from Annex A) ... 881
Syntax 31-15—Syntax for edge-control specifier (excerpt from Annex A) ... 882
Syntax 31-16—Syntax for controlled timing check events (excerpt from Annex A) ... 885
Syntax 32-1—Syntax for $sdf_annotate system task (not in Annex A) ... 900
Syntax 33-1—Syntax for cell (excerpt from Annex A) ... 903
Syntax 33-2—Syntax for declaring library in library map file (excerpt from Annex A) ... 904
Syntax 33-3—Syntax for include command (excerpt from Annex A) ... 905
Syntax 33-4—Syntax for configurations (excerpt from Annex A) ... 906
Syntax 35-1—DPI import declaration syntax (excerpt from Annex A) ... 945
Syntax 35-2—DPI export declaration syntax (excerpt from Annex A) ... 950

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

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 document(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.

   IEEE Standards documents are developed within IEEE Societies and Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of IEEE and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards. Use of IEEE Standards documents is wholly voluntary. *IEEE documents are made available for use subject to important notices and legal disclaimers (see* https://standards.ieee.org/ipr/disclaimers.html *for more information).*

   IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations. This Dual Logo International Standard was jointly developed by the IEC and IEEE under the terms of that agreement.

2) The formal decisions 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. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE SA) Standards Board.

3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE 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 (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.

8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material 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. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility.

IEC 62530/IEEE Std 1800 was processed through IEC technical committee 91: Electronics assembly technology, under the IEC/IEEE Dual Logo Agreement. It is an International Standard.

AVA Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

The text of this International Standard is based on the following documents:

IEEE Std	FDIS	Report on voting
1800 (2017)	91/1714/FDIS	91/1726/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

The IEC Technical Committee and IEEE Technical Committee have decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

Documents
Document Sharing Hub
Full document details: BS IEC 62530:2021

IEEE Std 1800™-2017
(Revision of
IEEE Std 1800-2012)

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 6 December 2017

IEEE-SA Standards Board

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

Abstract: The definition of the language syntax and semantics for SystemVerilog, which is a unified hardware design, specification, and verification language, is provided. This standard includes support for modeling hardware at the behavioral, register transfer level (RTL), and gate-level abstraction levels, and for writing testbenches using coverage, assertions, object-oriented programming, and constrained random verification. The standard also provides application programming interfaces (APIs) to foreign programming languages.

Keywords: assertions, design automation, design verification, hardware description language, HDL, HDVL, IEEE 1800™, PLI, programming language interface, SystemVerilog, Verilog®, VPI

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

Translations

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

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

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

Laws and regulations

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

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted 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 these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, 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.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://ieeexplore.ieee.org or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of 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 standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Documents
Document Sharing Hub

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Full document details: BS IEC 62530:2021

Introduction

This introduction is not part of IEEE Std 1800-2017, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

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 hardware description language (HDL) [B4], 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, 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 (IP). 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 that they can support in order to deliver their solutions in this area.

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

Part One:
Design and Verification Constructs

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

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

1. Overview

1.1 Scope

This standard provides the definition of the language syntax and semantics for the IEEE 1800™ SystemVerilog language, which is a unified hardware design, specification, and verification language. The standard includes support for behavioral, register transfer level (RTL), and gate-level hardware descriptions; testbench, coverage, assertion, object-oriented, and constrained random constructs; and also provides application programming interfaces (APIs) to foreign programming languages.

1.2 Purpose

This standard develops the IEEE 1800 SystemVerilog language in order to meet the increasing usage of the language in specification, design, and verification of hardware. This revision corrects errors and clarifies aspects of the language definition in IEEE Std 1800-2012.[1] This revision also provides enhanced features that ease design, improve verification, and enhance cross-language interactions.

1.3 Content summary

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

[1]Information on references can be found in Clause 2.

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

— 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

NOTE—An earlier standard, IEEE Std 1800-2009, represented a merger of two previous standards: IEEE Std 1364™-2005 and IEEE Std 1800-2005. 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 into the SystemVerilog standard enabled users to have all information regarding syntax and semantics in a single document.[2, 3]

1.4 Special terms

Throughout this standard, the following terms apply:

— *SystemVerilog 3.1a* refers to the Accellera *SystemVerilog 3.1a Language Reference Manual* [B4], a precursor to IEEE Std 1800-2005.[4]
— *Verilog* refers to IEEE Std 1364-2005 for the Verilog hardware description language (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 IEEE Std 1800-2005, *SystemVerilog* referred to just the extensions to the IEEE 1364-2005 Verilog language and did not include the Verilog base language.

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 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*).

[2]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.
[3]Verilog is a registered trademark of Cadence Design Systems, Inc.
[4]The numbers in brackets correspond to those of the bibliography in Annex Q.

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Documents
Document Sharing Hub
Full document details: BS IEC 62530:2021

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

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 that 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.

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Documents
Document Sharing Hub
Full document details: BS IEC 62530:2021

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, interface 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.

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.

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.

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

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 input/output (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.

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.

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Documents
Document Sharing Hub
Full document details: BS IEC 62530:2021

Part Three: Application Programming Interfaces

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.

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.

Documents
Document Sharing Hub
Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.
Full document details: BS IEC 62530:2021

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) 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 licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

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.[5]

ANSI X9.52-1998, American National Standard for Financial Services—Triple Data Encryption Algorithm Modes of Operation.[6]

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).[7]

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

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

IEEE Std 754™, IEEE Standard for Floating-Point Arithmetic.[8, 9]

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 Verilog Hardware Description Language.

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

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

IEEE Std 1800™-2009, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

IEEE Std 1800™-2012, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

IETF RFC 1319 (April 1992), The MD2 Message-Digest Algorithm.[10]

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.

[5]This document is available at http://www.cl.cam.ac.uk/~rja14/Papers/serpent.tar.gz.
[6]ANSI publications are available from the American National Standards Institute (http://www.ansi.org/).
[7]FIPS publications are available from the National Technical Information Service (http://www.ntis.gov/).
[8]IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).
[9]The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.
[10]IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

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

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.[11]

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.

[11]ISO publications are available from the International Organization for Standardization (http://www.iso.org/). IEC publications are available from the International Electrotechnical Commission (http://www.iec.ch). ISO/IEC publications are available from the American National Standards Institute (http://www.ansi.org/)..

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021

3. Design and verification building blocks

3.1 General

This clause describes the following:

— The purpose of modules, programs, interfaces, checkers, and primitives
— An overview of subroutines
— An overview of packages
— An overview of configurations
— An overview of design hierarchy
— Definition of compilation and elaboration
— Declaration name spaces
— Simulation time, time units, and time precision

This clause defines several important SystemVerilog terms and concepts that are used throughout this document. The clause also provides an overview of the purpose and usage of the modeling blocks used to represent a hardware design and its verification environment.

3.2 Design elements

A *design element* is a SystemVerilog module (see Clause 23), program (see Clause 24), interface (see Clause 25), checker (see Clause 17), package (see Clause 26), primitive (see Clause 28) or configuration (see Clause 33). These constructs are introduced by the keywords **`module`**, **`program`**, **`interface`**, **`checker`**, **`package`**, **`primitive`**, and **`config`**, respectively.

Design elements are the primary building blocks used to model and build up a design and verification environment. These building blocks are the containers for the declarations and procedural code that are discussed in subsequent clauses of this document.

This clause describes the purpose of these building blocks. Full details on the syntax and semantics of these blocks are defined in later clauses of this standard.

3.3 Modules

The basic building block in SystemVerilog is the module, enclosed between the keywords **`module`** and **`endmodule`**. Modules are primarily used to represent design blocks, but can also serve as containers for verification code and interconnections between verification blocks and design blocks. Some of the constructs that modules can contain include the following:

— Ports, with port declarations
— Data declarations, such as nets, variables, structures, and unions
— Constant declarations
— User-defined type definitions
— Class definitions
— Imports of declarations from packages
— Subroutine definitions
— Instantiations of other modules, programs, interfaces, checkers, and primitives
— Instantiations of class objects
— Continuous assignments

Published by IEC under licence from IEEE. © 2017 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: BS IEC 62530:2021