



IEEE

IEC 61523-1

Edition 3.0 2023-10

**INTERNATIONAL
STANDARD**

IEEE Std 1481™



**Delay and power calculation standards –
Part 1: Integrated Circuit (IC) Open Library Architecture (OLA)**

(<https://standards.iteh.ai>)

Document Preview

[IEC 61523-1:2023](#)

<https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023>





THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2019 IEEE

All rights reserved. IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the IEC Central Office. Any questions about IEEE copyright should be addressed to the IEEE. Enquiries about obtaining additional rights to this publication and other information requests should be addressed to the IEC or your local IEC member National Committee.

IEC Secretariat
3, rue de Varembe
CH-1211 Geneva 20
Switzerland
Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue
New York, NY 10016-5997
United States of America
stds.info@ieee.org
www.ieee.org

About the IEC

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

About IEC publications

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

IEC publications search - webstore.iec.ch/advsearchform

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

[IEC 61523-1:2023](https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023)

<https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023>

INTERNATIONAL STANDARD IEEE Std 1481™



Delay and power calculation standards –
Part 1: Integrated Circuit (IC) Open Library Architecture (OLA)

iteh Standards
(<https://standards.iteh.ai>)
Document Preview

[IEC 61523-1:2023](https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023)

<https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023>

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.01, 35.060

ISBN 978-2-8322-7539-9

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

Contents

1. Overview	30
1.1 Scope	30
1.2 Purpose	30
1.3 Introduction	30
1.4 Word usage	31
2. Normative references.....	32
3. Definitions.....	32
4. Acronyms and abbreviations	40
5. Typographical conventions.....	41
5.1 Syntactic elements	41
5.2 Conventions.....	42
6. DPCS flow.....	42
6.1 Overview	42
6.1.1 Procedural interface	44
6.1.2 Global policies and conventions	44
6.2 Flow of control	44
6.3 DPCM—application relationships	44
6.3.1 Technology library	45
6.3.2 Subrule	45
6.4 Interoperability	45
7. Delay calculation language (DCL).....	45
7.1 Character set	45
7.2 Lexical elements	45
7.2.1 Whitespace.....	46
7.2.2 Comments	46
7.2.3 Tokens	46
7.2.4 Header names.....	56
7.2.5 Preprocessing directives.....	56
7.3 Context.....	56
7.3.1 Space.....	56
7.3.2 Plane.....	56
7.3.3 Context operation.....	56
7.3.4 Library parallelism.....	56
7.3.5 Application parallelism	57
7.4 Data types	57
7.4.1 Base types	57
7.4.2 Native data types.....	57
7.4.3 Mathematical calculation data types	57
7.4.4 Pointer data types.....	58
7.4.5 Aggregate data types	58
7.5 Identifiers.....	64
7.5.1 Name spaces of identifiers	64
7.5.2 Storage durations of objects	64
7.5.3 Scope of identifiers	65
7.5.4 Linkages of identifiers	66

7.6 Operator descriptions.....	66
7.6.1 String prefix operator.....	66
7.6.2 Explicit string prefix operator.....	66
7.6.3 Embedded string prefix operator.....	67
7.6.4 String prefix semantics.....	67
7.6.5 Assignment operator.....	67
7.6.6 New operator.....	67
7.6.7 SCOPE operator(s).....	68
7.6.8 Launch operator.....	69
7.6.9 Purity operator.....	69
7.6.10 Force operator.....	70
7.7 Timing propagation.....	70
7.7.1 Timing checks.....	71
7.7.2 Test mode operators.....	71
7.8 Expressions.....	73
7.8.1 Array subscripting.....	74
7.8.2 Statement calls.....	74
7.8.3 General syntax.....	74
7.8.4 Method statement calls.....	74
7.8.5 Assign variable reference.....	75
7.8.6 Store variable reference.....	75
7.8.7 Mathematical expressions.....	75
7.8.8 Mathematical operators.....	76
7.8.9 Discrete math expression.....	77
7.8.10 INT discrete.....	78
7.8.11 PINLIST discrete.....	78
7.8.12 Logical expressions and operators.....	78
7.8.13 MODE expressions.....	78
7.8.14 Embedded C code expressions.....	80
7.8.15 Computation order.....	81
7.9 DCL mathematical statements.....	83
7.9.1 Statement names.....	83
7.9.2 Modifiers.....	87
7.9.3 Prototypes.....	89
7.9.4 Statement failure.....	92
7.9.5 Type definition statements.....	92
7.9.6 Interfacing statements.....	93
7.9.7 DCL to C communication.....	95
7.9.8 Constant statement.....	95
7.9.9 Calculation statements.....	96
7.9.10 METHOD statement.....	98
7.10 Predefined types.....	99
7.10.1 ACTIVITY_HISTORY_TYPE.....	99
7.10.2 HISTORY_TYPE.....	100
7.10.3 LOAD_HISTORY_TYPE.....	101
7.10.4 CELL_LIST_TYPE.....	102
7.10.5 TECH_TYPE.....	102
7.10.6 DELAY_REC_TYPE.....	102
7.10.7 SLEW_REC_TYPE.....	103
7.10.8 CHECK_REC_TYPE.....	103
7.10.9 CCDB_TYPE.....	103
7.10.10 CELL_DATA_TYPE.....	103
7.10.11 PCDB_TYPE.....	103
7.10.12 PIN_ASSOCIATION.....	104
7.10.13 PATH_DATA_TYPE.....	104
7.10.14 STD_STRUCT.....	104

7.11	Predefined variables.....	105
7.11.1	ARGV	105
7.11.2	CONTROL_PARM.....	105
7.12	Built-in function calls	105
7.12.1	ABS.....	106
7.12.2	Complex number components.....	106
7.12.3	EXPAND.....	106
7.12.4	Array functions.....	106
7.12.5	Messaging functions	107
7.13	Tables.....	109
7.13.1	TABLEDEF statement	109
7.13.2	Table visibility rules.....	111
7.13.3	TABLE statement.....	112
7.13.4	LOAD_TABLE statement.....	115
7.13.5	UNLOAD_TABLE statement.....	117
7.13.6	WRITE_TABLE statement	118
7.13.7	ADD_ROW statement.....	118
7.13.8	DELETE_ROW statement.....	119
7.14	Built-in library functions	120
7.14.1	Numeric conversion functions	120
7.14.2	Tech_family functions.....	122
7.14.3	Trigonometric functions.....	123
7.14.4	Context manipulation functions	124
7.14.5	Debug controls.....	126
7.14.6	Utility functions	126
7.14.7	Table functions	127
7.14.8	Subrule controls	128
7.15	Library control statements	129
7.15.1	Meta-variables.....	129
7.15.2	TECH_FAMILY.....	129
7.15.3	RULENAME	129
7.15.4	CONTROL_PARM.....	129
7.15.5	SUBRULE statement	129
7.15.6	Path list expansion rules.....	130
7.15.7	SUBRULES statement.....	131
7.15.8	Control file.....	131
7.15.9	TECH_FAMILY statement.....	133
7.15.10	SUBRULE and SUBRULES statements.....	134
7.16	Modeling.....	134
7.16.1	Types of modeling.....	134
7.16.2	Model organization	135
7.16.3	MODELPROC statement.....	136
7.16.4	SUBMODEL statement	137
7.16.5	Modeling statements	139
7.16.6	TEST_BUS statement.....	148
7.16.7	INPUT statement.....	149
7.16.8	OUTPUT statement.....	152
7.16.9	DO statement	153
7.16.10	PROPERTIES statement.....	174
7.16.11	SETVAR statement	175
7.17	Embedded C code.....	176
7.18	Definition of a subrule	176
7.19	Pragma.....	177
7.19.1	IMPORT_EXPORT_TAG.....	177

8. Power modeling and calculation.....	177
8.1 Power overview	177
8.2 Caching state information.....	178
8.2.1 Initializing the state cache.....	178
8.2.2 State cache lifetime	178
8.3 Caching load and slew information	178
8.3.1 Loading the load and slew cache	179
8.3.2 Load and slew cache lifetime.....	179
8.4 Simulation switching events	180
8.5 Partial swing events	180
8.6 Power calculation.....	180
8.7 Accumulation of power consumption by the design.....	182
8.8 Group Pin List syntax and semantics.....	182
8.8.1 Syntax	182
8.8.2 Semantics	183
8.8.3 Example	183
8.9 Group Condition List syntax and semantics	183
8.9.1 Syntax	184
8.9.2 Semantics	184
8.9.3 Example	184
8.10 Sensitivity list syntax and semantics.....	184
8.10.1 Syntax	185
8.10.2 Semantics.....	185
8.10.3 Example	185
8.11 Group condition language.....	185
8.11.1 Syntax	186
8.11.2 Semantics	186
8.11.3 Condition expression operator precedence.....	188
8.11.4 Condition expressions referencing pin states and transitions	188
8.11.5 Semantics of nonexistent pins.....	189
9. Application and library interaction	189
9.1 behavior model domain	189
9.2 vectorTiming and vectorPower model domains	190
9.2.1 Power unit conversion.....	190
9.2.2 Vector power calculation.....	190
10. Procedural interface (PI).....	191
10.1 Overview	191
10.1.1 DPCM	191
10.1.2 Application.....	192
10.1.3 libdcmr	192
10.2 Control and data flow	192
10.3 Architectural requirements.....	192
10.4 Data ownership technique.....	193
10.4.1 Persistence of data passed across the PI.....	193
10.4.2 Data cache guidelines for the DPCM.....	193
10.4.3 Application/DPCM interaction.....	194
10.4.4 Application initializes message/memory handling.....	194
10.4.5 Application loads and initializes the DPCM	194
10.4.6 Application requests timing models for cell instances	194
10.5 Model domain issues	194
10.5.1 Model domain selection	194
10.5.2 Model domain determination	195
10.5.3 DPCM invokes application modeling callback functions	195
10.5.4 Application requests propagation delay	195
10.5.5 DPCM calls application EXTERNAL functions.....	196
10.6 Reentry requirements.....	196

10.7 Application responsibilities when using a DPCM	196
10.7.1 Standard Structure rules	196
10.7.2 User object registration	197
10.7.3 Selection of early and late slew values.....	197
10.7.4 Semantics of slew values	198
10.7.5 Slew calculations	198
10.8 Application use of the DPCM	198
10.8.1 Initialization of the DPCM.....	198
10.8.2 Context creation	199
10.8.3 Dynamic linking.....	199
10.8.4 Subrule initialization	200
10.8.5 Use of the DPCM.....	201
10.8.6 Application control.....	201
10.8.7 Application execution	201
10.8.8 Termination of DPCM.....	201
10.9 DPCM library organization.....	202
10.9.1 Multiple technologies.....	202
10.9.2 Model names.....	202
10.9.3 DPCM error handling.....	202
10.10 C level language for EXPOSE and EXTERNAL functions.....	203
10.10.1 Integer return code	203
10.10.2 The Standard Structure pointer.....	203
10.10.3 Result structure pointer	203
10.10.4 Passed arguments	203
10.10.5 DCL array indexing.....	204
10.10.6 Conversion to C data types	204
10.10.7 include files.....	204
10.11 PIN and BLOCK data structure requirements.....	205
10.12 DCM_STD_STRUCT Standard Structure	206
10.12.1 Alternate semantics for Standard Structure fields	208
10.12.2 Reserved fields.....	209
10.12.3 Standard Structure value restriction	209
10.13 DCMTransmittedInfo structure.....	209
10.14 Environment or user variables	209
10.15 Procedural interface (PI) functions summary	209
10.15.1 Expose functions.....	209
10.15.2 External functions	217
10.15.3 Deprecated functions.....	220
10.16 Implicit functions.....	222
10.16.1 libdcmlr.....	222
10.16.2 Run-time library utility functions.....	223
10.16.3 Memory control functions.....	223
10.16.4 Message and error control functions	224
10.16.5 Calculation functions	225
10.16.6 Modeling functions	225
10.17 PI function table description	225
10.17.1 Arguments	226
10.17.2 DCL syntax	227
10.17.3 C syntax	227

10.18 PI function descriptions	227
10.18.1 Interconnect loading related functions	227
10.18.2 Interconnect delay related functions	234
10.18.3 Functions accessing netlist information	237
10.18.4 Functions exporting limit information	246
10.18.5 Functions getting/setting model information	247
10.18.6 Functions importing instance name information	260
10.18.7 Process information functions	263
10.18.8 Miscellaneous standard interface functions	265
10.18.9 Power-related functions	274
10.19 Application context	282
10.19.1 pathData association	282
10.20 Application and library interaction	282
10.20.1 behavior model domain	283
10.20.2 vectorTiming and vectorPower model domains	283
10.20.3 Power unit conversion	284
10.20.4 Vector power calculation	284
10.21 Parasitic analysis	285
10.21.1 Assumptions	285
10.21.2 Parasitic networks	285
10.21.3 Basic definitions	285
10.21.4 Parasitic element data structure	287
10.21.5 Coordinates	291
10.21.6 Parasitic subnets	291
10.21.7 Pin parasitics	298
10.21.8 Modeling internal nodes	301
10.21.9 Load and interconnect models	303
10.21.10 Obtaining parasitic networks	307
10.21.11 Persistent storage of load and interconnect models	308
10.21.12 Calculating effective capacitances and driving resistances	311
10.21.13 Parasitic estimation	314
10.21.14 Threshold voltages	319
10.21.15 Obtaining aggressor window overlaps	320
10.22 Noise analysis	327
10.22.1 Types of noise	328
10.22.2 Noise models	329
10.22.3 Noise waveforms	332
10.22.4 Noise network models	338
10.22.5 Calculating composite noise at cell inputs	344
10.22.6 Calculating composite noise at cell outputs	347
10.22.7 Setting noise budgets	351
10.22.8 Reporting noise violations	352
10.23 Delay and slew calculations for differential circuits	354
10.23.1 Sample figures	354
10.23.2 appGetArrivalOffsetsByName	356
10.23.3 API extensions for function modeling	357
10.23.4 Explicit APIs for user-defined primitives	364
10.23.5 APIs for hierarchy	366
10.23.6 Built-in APIs for function modeling	366
10.23.7 API Extensions for VECTOR modeling	367
10.23.8 APIs for XWF	368
10.23.9 Extensions and changes to voltages and temperature APIs	372
10.23.10 Operating conditions	374
10.23.11 On-chip process variation	377
10.23.12 Accessing properties and attributes	383
10.23.13 APIs for attribute within a PIN object	404

10.23.14 Connectivity	412
10.23.15 Control of timing arc existence and state	414
10.23.16 Modeling cores.....	419
10.23.17 Default pin slews and interface version calls	424
10.23.18 API to access library required resources	426
10.23.19 Resource types	427
10.23.20 Library extensions for phase locked loop processing.....	428
10.23.21 API definitions for external conditions	429
10.23.22 Extensions for listing pins.....	433
10.23.23 Memory BIST mapping	434
10.23.24 dpcmGetCellTestProcedure.....	436
10.24 Interconnect delay calculation intraface	437
10.24.1 Control and data flows	437
10.24.2 Model generation functions.....	438
10.24.3 Calculation functions	440
10.24.4 Cell calculation functions.....	441
10.24.5 ICM initialization	446
10.25 DCL run-time support.....	453
10.25.1 Array manipulation functions.....	453
10.25.2 Memory management	457
10.25.3 Structure manipulation functions	457
10.25.4 Initialization functions	462
10.26 Calculation functions	473
10.26.1 delay.....	473
10.26.2 slew	474
10.26.3 check	474
10.27 Modeling functions	476
10.27.1 modelSearch.....	476
10.27.2 Mode operators	478
10.27.3 Arrival time merging.....	479
10.27.4 Edge propagation communication to the application	480
10.27.5 Edge propagation communication to the DPCM	482
10.27.6 newTimingPin	483
10.27.7 newDelayMatrixRow	483
10.27.8 newNetSinkPropagateSegments	484
10.27.9 newNetSourcePropagateSegments.....	486
10.27.10 newPropagateSegment	487
10.27.11 newTestMatrixRow	488
10.27.12 newAltTestSegment	489
10.27.13 Interactions between interconnect modeling and modeling functions	489
10.28 Deprecated functions	490
10.28.1 Parasitic handling.....	490
10.28.2 Array manipulation functions.....	499
10.28.3 Memory management	501
10.28.4 Initialization functions	504
10.29 Standard Structure (std_stru.h) file.....	515
10.30 Standard macros (std_macros.h) file	535
10.31 Standard interface structures (dcmintf.h) file	543
10.32 Standard loading (dcmload.h) file	547
10.33 Standard debug (dcmdebug.h) file	551
10.34 Standard array (dcmgarray.h) file	578
10.35 Standard user array defines (dcmuarray.h) file.....	582
10.36 Standard platform-dependency (dcmpltfm.h) file	586
10.37 Standard state variables (dcmstate.h) file.....	592

11. Parasitics.....	595
11.1 Introduction.....	596
11.2 Targeted applications for SPEF.....	596
11.3 SPEF specification.....	596
11.3.1 Grammar.....	596
11.3.2 Escaping rules.....	598
11.3.3 File syntax.....	598
11.3.4 Comments.....	604
11.3.5 File semantics.....	604
11.4 Examples.....	623
11.4.1 Basic *D_NET file.....	623
11.4.2 Basic *R_NET file.....	626
11.4.3 *R_NET with poles and residues plus name mapping.....	627
11.4.4 *D_NET with triplet par_value.....	628
11.4.5 *R_NET with poles and residues plus triplet par_value.....	631
11.4.6 Merging SPEF files.....	633
11.4.7 A SPEF file header section with *VARIATION_PARAMETERS definition.....	637
11.4.8 CAP and RES statements with sensitivity information in a SPEF file.....	637
Annex A (informative) Implementation requirements.....	638
Annex B (informative) Participants.....	640

iteh Standards
(<https://standards.iteh.ai>)
Document Preview

[IEC 61523-1:2023](https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023)

<https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023>

Figures

Figure 1—High-level DPCS architecture linkage structure	43
Figure 2—Function graph form.....	135
Figure 3—DPCM/application procedural interface.....	192
Figure 4—PIN and PINLIST.....	205
Figure 5—Parallel drivers example.....	239
Figure 6—Subnet node mapping.....	294
Figure 7—Differential buffer chain.....	354
Figure 8—Timing models for a differential buffer chain	355
Figure 9—Arrival offset for differential signals.....	355
Figure 10—Priority operation	360
Figure 11—Precedence.....	361
Figure 12—Strand ranges.....	362
Figure 13—Various methods of using XWF to model waveforms for slew computation.....	368
Figure 14—Propagation of XWF “handles” by application.....	369
Figure 15—Application, CCM, and ICM control and data flows	438
Figure 16—Clock separation.....	475
Figure 17—Bias calculation.....	476
Figure 18—Clock pulse width.....	476
Figure 19—Sample MODELPROC results.....	486
Figure 20—Additional MODELPROC results.....	486
Figure 21—Capacitance value example.....	491
Figure 22—Equation for poles and residues	492
Figure 23—Example RC network	498
Figure 24—SPEF targeted applications.....	596

Standards
standards.iteh.ai
Document Preview

[IEC 61523-1:2023](https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023)

<https://standards.iteh.ai/catalog/standards/sist/00f8007e-5c25-487e-9476-66186f41bb5b/iec-61523-1-2023>

Tables

Table 1—Keywords.....	46
Table 2—DCL predefined references to Standard Structure fields.....	49
Table 3—DCL compiler generated predefined identifiers.....	52
Table 4—Edge types and conversions.....	54
Table 5—Propagation mode conversions.....	54
Table 6—Calculation mode conversions.....	54
Table 7—TEST_TYPE conversions.....	55
Table 8—Purity operator.....	70
Table 9—Timing resolution modes.....	70
Table 10—Test mode operators table.....	71
Table 11—Mathematical operators.....	77
Table 12—Logical operators.....	78
Table 13—Mathematical operator precedence (high to low).....	81
Table 14—Logical operator precedence (high to low).....	81
Table 15—Type definition for ACTIVE_HISTORY_TYPE.....	99
Table 16—Permitted activityCode values.....	100
Table 17—Type definition for HISTORY_TYPE.....	100
Table 18—Rule history info message types.....	101
Table 19—Table History inform message types.....	101
Table 20—Permitted kind values.....	101
Table 21—LOAD_HISTORY_TYPE.....	101
Table 22—CELL_LIST_TYPE.....	102
Table 23—TECH_TYPE.....	102
Table 24—DELAY_REC_TYPE.....	102
Table 25—SLEW_REC_TYPE.....	103
Table 26—CHECK_REC_TYPE.....	103
Table 27—CCDB_TYPE.....	103
Table 28—CELL_DATA_TYPE.....	103
Table 29—CCDB_TYPE.....	104
Table 30—PIN_ASSOCIATION.....	104
Table 31—PATH_DATA_TYPE.....	104
Table 32—STD_STRUCT.....	105
Table 33—ARGV.....	105
Table 34—CONTROL_PARM.....	105
Table 35—Library function floor.....	121
Table 36—Library function ifloor.....	121
Table 37—Library function ceil.....	121
Table 38—Library Function iceil.....	121
Table 39—Library function rint.....	121
Table 40—Library function round.....	122
Table 41—Library function trunc.....	122
Table 42—Library function itrunc.....	122
Table 43—Library function map_tech_family.....	122
Table 44—Library function current_tech_type.....	122
Table 45—Library function subrule_tech_type.....	123
Table 46—Library function subrule_tech_type.....	123
Table 47—Library function get_technology_list.....	123
Table 48—Library function cos.....	123
Table 49—Library function sin.....	123
Table 50—Library function tan.....	124
Table 51—Library function new_plane.....	124
Table 52—Library function get_plane_name.....	124
Table 53—Library function get_space_name.....	124
Table 54—Library function get_max_spaces.....	125

Table 55—Library function <code>get_max_planes</code>	125
Table 56—Library function <code>get_space_coordinate</code>	125
Table 57—Library function <code>get_plane_coordinate</code>	125
Table 58—Library function <code>set_busy_wait</code>	126
Table 59—Library function <code>change_debug_level</code>	126
Table 60—Library function <code>get_caller_stack</code>	126
Table 61—Library function <code>GET_LOAD_HISTORY</code>	126
Table 62—Library function <code>GET_CELL_LIST</code>	127
Table 63—Library function <code>GET_ROW_COUNT</code>	127
Table 64—Library function <code>STEP_TABLE</code>	127
Table 65—Library function <code>GET_LOAD_PATH</code>	128
Table 66—Library function <code>GET_RULE_NAME</code>	128
Table 67—Library function <code>ADD_RULE</code>	128
Table 68—Data type clause.....	143
Table 69—Arc data types	144
Table 70—Validity of predefined identifiers for <code>STORE</code> clause	152
Table 71—Logic operators (valid for behavior, vectorTiming, and vectorPower model domains).....	157
Table 72—Logical equivalence operators (valid for behavior model domain)	157
Table 73—Unary bitwise operators (valid for behavior, vectorTiming, and vectorPower model domains).....	157
Table 74—Binary bitwise operators (valid for behavior, vectorTiming, and vectorPower model domains).....	158
Table 75—Binary operators (valid for behavior model domain).....	158
Table 76—Node primitives for control operators (valid for behavior model domain).....	158
Table 77—Node primitives for edge operators (continued) (valid for behavior, vectorTiming, and vectorPower model domains).....	159
Table 78—Node primitives for precedence control operators (valid for behavior model domain).....	159
Table 79—Node primitives for constant operators (valid for behavior, vectorTiming, vectorPower model domains).....	160
Table 80—Node primitives for user-defined operators	160
Table 81—Node primitives for miscellaneous operators.....	160
Table 82—Binary reduction operators.....	161
Table 83—Bitwise reduction operators	162
Table 84—Logical reduction operators	163
Table 85—Array of bits operators	164
Table 86—Edge operators	165
Table 87—Higher function nodes.....	166
Table 88—Constant value nodes	168
Table 89—Miscellaneous operators	169
Table 90—User-defined operators.....	169
Table 91—Valid modifier enumerations for given node primitive operators	170
Table 92—Syntax for a <code>GroupPinString</code>	183
Table 93—Syntax for a <code>GroupConditionString</code>	184
Table 94—Syntax for a <code>SensitivityPinString</code>	185
Table 95—Syntax for a <code>condition_expression</code>	186
Table 96— <code>PinName_Identifier</code> semantics	187
Table 97— <code>PinName_Level</code> semantics.....	187
Table 98— <code>PinName_State</code> semantics	187
Table 99—Condition expression operators.....	188
Table 100—Interaction between multiple technologies and application	202
Table 101—Return code most significant byte.....	203
Table 102—Return code least significant bytes	203
Table 103—Data types defined in DCL and C	204
Table 104—Header files.....	205
Table 105—Predefined macro names	206
Table 106—Alternate semantics for Standard Structure fields.....	208