



**IEEE**

**IEC 61523-1**

Edition 2.0 2012-06

# INTERNATIONAL STANDARD

**IEEE Std 1481™**

Delay and power calculation standards –  
Part 1: Integrated circuit delay and power calculation systems  
(standards.iteh.ai)

IEC 61523-1:2012  
<https://standards.iteh.ai/catalog/standards/sist/0c1ed01e-318e-4f0f-ba05-dac666cce572/iec-61523-1-2012>



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

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

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

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

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland  
Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://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](mailto:stds.info@ieee.org)  
[www.ieee.org](http://www.ieee.org)

### About the IEC

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

### About IEC publications

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

#### Useful links:

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

The advanced search enables you 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It also gives information on projects, replaced and withdrawn publications.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](http://webstore.iec.ch/justpublished)

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

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

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - [webstore.iec.ch/csc](http://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: [csc@iec.ch](mailto:csc@iec.ch).

INTERNATIONAL STANDARD PREVIEW  
(standards.iteh.ai)  
IEC 61523-1:2012  
<http://standards.iteh.ai/technicalstandards/iec/standard/3180-456805-dac666cce572/iec-61523-1-2012>



IEEE

IEC 61523-1

Edition 2.0 2012-06

# INTERNATIONAL STANDARD

IEEE Std 1481™

Delay and power calculation standards –  
Part 1: Integrated circuit delay and power calculation systems

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

<https://standards.iteh.ai/catalog/standards/sist/0c1ed01e-318e-4f0f-ba05-dac666cce572/iec-61523-1-2012>

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

PRICE CODE **XH**

ICS 25.040; 35.060

ISBN 978-2-83220-107-7

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

## Contents

1	Overview.....	1
1.1	Scope.....	1
1.2	Purpose.....	2
1.3	Introduction.....	2
2	Normative references.....	3
3	Definitions.....	4
4	Acronyms and abbreviations.....	13
5	Typographical conventions.....	14
5.1	Syntactic elements.....	14
5.2	Conventions.....	15
6	DPCS flow.....	16
6.1	Overview.....	16
6.1.1	Procedural interface.....	17
6.1.2	Global policies and conventions.....	17
6.2	Flow of control.....	17
6.3	DPCM—application relationships.....	18
6.3.1	Technology library.....	18
6.3.2	Subrule.....	18
6.4	Interoperability.....	18
7	Delay calculation language (DCL).....	19
7.1	Character set.....	19
7.2	Lexical elements.....	19
7.2.1	Whitespace.....	19
7.2.2	Comments.....	19
7.2.3	Tokens.....	19
7.2.4	Header names.....	31
7.2.5	Preprocessing directives.....	31
7.3	Context.....	31
7.3.1	Space.....	31
7.3.2	Plane.....	31
7.3.3	Context operation.....	31
7.3.4	Library parallelism.....	31
7.3.5	Application parallelism.....	32
7.4	Data types.....	32
7.4.1	Base types.....	32
7.4.2	Native data types.....	32
7.4.3	Mathematical calculation data types.....	32
7.4.4	Pointer data types.....	33
7.4.5	Aggregate data types.....	33
7.5	Identifiers.....	39
7.5.1	Name spaces of identifiers.....	39
7.5.2	Storage durations of objects.....	39
7.5.3	Scope of identifiers.....	40
7.5.4	Linkages of identifiers.....	41
7.6	Operator descriptions.....	41
7.6.1	String prefix operator.....	41
7.6.2	Explicit string prefix operator.....	41
7.6.3	Embedded string prefix operator.....	42
7.6.4	String prefix semantics.....	42
7.6.5	Assignment operator.....	42
7.6.6	New operator.....	42
7.6.7	SCOPE operator(s).....	43

7.6.8	Launch operator.....	44
7.6.9	Purity operator.....	44
7.6.10	Force operator.....	45
7.7	Timing propagation.....	45
7.7.1	Timing checks.....	46
7.7.2	Test mode operators.....	46
7.8	Expressions.....	48
7.8.1	Array subscripting.....	49
7.8.2	Statement calls.....	49
7.8.3	General syntax.....	49
7.8.4	Method statement calls.....	49
7.8.5	Assign variable reference.....	50
7.8.6	Store variable reference.....	50
7.8.7	Mathematical expressions.....	50
7.8.8	Mathematical operators.....	51
7.8.9	Discrete math expression.....	52
7.8.10	INT discrete.....	52
7.8.11	PINLIST discrete.....	53
7.8.12	Logical expressions and operators.....	53
7.8.13	MODE expressions.....	53
7.8.14	Embedded C code expressions.....	55
7.8.15	Computation order.....	56
7.9	DCL mathematical statements.....	58
7.9.1	Statement names.....	58
7.9.2	Clauses.....	58
7.9.3	Modifiers.....	62
7.9.4	Prototypes.....	64
7.9.5	Statement failure.....	67
7.9.6	Type definition statements.....	67
7.9.7	Interfacing statements.....	68
7.9.8	DCL to C communication.....	70
7.9.9	Constant statement.....	71
7.9.10	Calculation statements.....	71
7.9.11	METHOD statement.....	74
7.10	Predefined types.....	75
7.10.1	ACTIVITY_HISTORY_TYPE.....	75
7.10.2	HISTORY_TYPE.....	76
7.10.3	LOAD_HISTORY_TYPE.....	77
7.10.4	CELL_LIST_TYPE.....	77
7.10.5	TECH_TYPE.....	78
7.10.6	DELAY_REC_TYPE.....	78
7.10.7	SLEW_REC_TYPE.....	78
7.10.8	CHECK_REC_TYPE.....	78
7.10.9	CCDB_TYPE.....	79
7.10.10	CELL_DATA_TYPE.....	79
7.10.11	PCDB_TYPE.....	79
7.10.12	PIN_ASSOCIATION.....	79
7.10.13	PATH_DATA_TYPE.....	80
7.10.14	STD STRUCT.....	80
7.11	Predefined variables.....	80
7.11.1	ARGV.....	80
7.11.2	CONTROL_PARM.....	81
7.12	Built-in function calls.....	81
7.12.1	ABS.....	81
7.12.2	Complex number components.....	81
7.12.3	EXPAND.....	82

7.12.4	Array functions.....	82
7.12.5	Messaging functions.....	82
7.13	Tables.....	84
7.13.1	TABLEDEF statement.....	85
7.13.2	Table visibility rules.....	87
7.13.3	TABLE statement.....	87
7.13.4	LOAD_TABLE statement.....	91
7.13.5	UNLOAD_TABLE statement.....	93
7.13.6	WRITE_TABLE statement.....	94
7.13.7	ADD_ROW statement.....	94
7.13.8	DELETE_ROW statement.....	95
7.14	Built-in library functions.....	96
7.14.1	Numeric conversion functions.....	96
7.14.2	Tech_family functions.....	98
7.14.3	Trigonometric functions.....	99
7.14.4	Context manipulation functions.....	99
7.14.5	Debug controls.....	101
7.14.6	Utility functions.....	102
7.14.7	Table functions.....	102
7.14.8	Subrule controls.....	103
7.15	Library control statements.....	104
7.15.1	Meta-variables.....	105
7.15.2	TECH_FAMILY.....	105
7.15.3	RULENAME.....	105
7.15.4	CONTROL_PARM.....	105
7.15.5	SUBRULE statement.....	105
7.15.6	Path list expansion rules.....	106
7.15.7	SUBRULES statement.....	107
7.15.8	Control file.....	107
7.15.9	TECH_FAMILY statement.....	109
7.15.10	SUBRULE and SUBRULES statements.....	109
7.16	Modeling.....	110
7.16.1	Types of modeling.....	110
7.16.2	Model organization.....	111
7.16.3	MODELPROC statement.....	112
7.16.4	SUBMODEL statement.....	113
7.16.5	Modeling statements.....	114
7.16.6	TEST_BUS statement.....	124
7.16.7	INPUT statement.....	124
7.16.8	OUTPUT statement.....	128
7.16.9	DO statement.....	129
7.16.10	PROPERTIES statement.....	153
7.16.11	SETVAR statement.....	154
7.17	Embedded C code.....	155
7.18	Definition of a subrule.....	155
7.19	Pragma.....	156
7.19.1	IMPORT_EXPORT_TAG.....	156
8	Power modeling and calculation.....	157
8.1	Power overview.....	157
8.2	Caching state information.....	158
8.2.1	Initializing the state cache.....	158
8.2.2	State cache lifetime.....	158
8.3	Caching load and slew information.....	158
8.3.1	Loading the load and slew cache.....	159
8.3.2	Load and slew cache lifetime.....	159

8.4	Simulation switching events.....	159
8.5	Partial swing events.....	160
8.6	Power calculation.....	160
8.7	Accumulation of power consumption by the design.....	162
8.8	Group Pin List syntax and semantics.....	162
8.8.1	Syntax.....	162
8.8.2	Semantics.....	162
8.8.3	Example.....	163
8.9	Group Condition List syntax and semantics.....	163
8.9.1	Syntax.....	163
8.9.2	Semantics.....	163
8.9.3	Example.....	164
8.10	Sensitivity list syntax and semantics.....	164
8.10.1	Syntax.....	164
8.10.2	Semantics.....	164
8.10.3	Example.....	165
8.11	Group condition language.....	165
8.11.1	Syntax.....	165
8.11.2	Semantics.....	166
8.11.3	Condition expression operator precedence.....	168
8.11.4	Condition expressions referencing pin states and transitions.....	168
8.11.5	Semantics of nonexistent pins.....	168
9	Application and library interaction.....	170
9.1	behavior model domain.....	170
9.2	vectorTiming and vectorPower model domains.....	170
9.2.1	Power unit conversion.....	170
9.2.2	Vector power calculation.....	171
10	Procedural interface (PI).....	172
10.1	Overview.....	172
10.1.1	DPCM.....	172
10.1.2	Application.....	172
10.1.3	libdcmr.....	172
10.2	Control and data flow.....	173
10.3	Architectural requirements.....	173
10.4	Data ownership technique.....	173
10.4.1	Persistence of data passed across the PI.....	173
10.4.1	Data cache guidelines for the DPCM.....	174
10.4.2	Application/DPCM interaction.....	174
10.4.3	Application initializes message/memory handling.....	174
10.4.4	Application loads and initializes the DPCM.....	174
10.4.5	Application requests timing models for cell instances.....	175
10.5	Model domain issues.....	175
10.5.1	Model domain selection.....	175
10.5.2	Model domain determination.....	175
10.5.3	DPCM invokes application modeling callback functions.....	175
10.5.4	Application requests propagation delay.....	176
10.5.5	DPCM calls application EXTERNAL functions.....	177
10.6	Reentry requirements.....	177
10.7	Application responsibilities when using a DPCM.....	177
10.7.1	Standard Structure rules.....	177
10.7.2	User object registration.....	177
10.7.3	Selection of early and late slew values.....	178
10.7.4	Semantics of slew values.....	178
10.7.5	Slew calculations.....	179



10.8	Application use of the DPCM.....	179
10.8.1	Initialization of the DPCM.....	179
10.8.2	Context creation.....	180
10.8.3	Dynamic linking.....	180
10.8.4	Subrule initialization.....	181
10.8.5	Use of the DPCM.....	181
10.8.6	Application control.....	181
10.8.7	Application execution.....	182
10.8.8	Termination of DPCM.....	182
10.9	DPCM library organization.....	182
10.9.1	Multiple technologies.....	182
10.9.2	Model names.....	183
10.9.3	DPCM error handling.....	183
10.10	C level language for EXPOSE and EXTERNAL functions.....	183
10.10.1	Integer return code.....	183
10.10.2	The Standard Structure pointer.....	184
10.10.3	Result structure pointer.....	184
10.10.4	Passed arguments.....	184
10.10.5	DCL array indexing.....	184
10.10.6	Conversion to C data types.....	184
10.10.7	include files.....	185
10.11	PIN and BLOCK data structure requirements.....	186
10.12	DCM_STD_STRUCT Standard Structure.....	186
10.12.1	Alternate semantics for Standard Structure fields.....	189
10.12.2	Reserved fields.....	190
10.12.3	Standard Structure value restriction.....	190
10.13	DCMTransmittedInfo structure.....	190
10.14	Environment or user variables.....	190
10.15	Procedural interface (PI) functions summary.....	190
10.15.1	Expose functions.....	191
10.15.2	External functions.....	199
10.15.3	Deprecated functions.....	202
10.16	Implicit functions.....	205
10.16.1	libdcmlr.....	205
10.16.2	Run-time library utility functions.....	206
10.16.3	Memory control functions.....	206
10.16.4	Message and error control functions.....	208
10.16.5	Calculation functions.....	208
10.16.6	Modeling functions.....	208
10.17	PI function table description.....	209
10.17.1	Arguments.....	209
10.17.2	DCL syntax.....	210
10.17.3	C syntax.....	210
10.18	PI function descriptions.....	210
10.18.1	Interconnect loading related functions.....	210
10.18.2	Interconnect delay related functions.....	217
10.18.3	Functions accessing netlist information.....	221
10.18.4	Functions exporting limit information.....	229
10.18.5	Functions getting/setting model information.....	231
10.18.6	Functions importing instance name information.....	244
10.18.7	Process information functions.....	246
10.18.8	Miscellaneous standard interface functions.....	247
10.18.9	Power-related functions.....	257
10.19	Application context.....	265
10.19.1	pathData association.....	265



10.20	Application and library interaction.....	265
10.20.1	behavior model domain.....	266
10.20.2	vectorTiming and vectorPower model domains.....	267
10.20.3	Power unit conversion.....	267
10.20.4	Vector power calculation.....	267
10.21	Parasitic analysis.....	268
10.21.1	Assumptions.....	268
10.21.2	Parasitic networks.....	268
10.21.3	Basic definitions.....	268
10.21.4	Parasitic element data structure.....	270
10.21.5	Coordinates.....	274
10.21.6	Parasitic subnets.....	274
10.21.7	Pin parasitics.....	282
10.21.8	Modeling internal nodes.....	285
10.21.9	Load and interconnect models.....	287
10.21.10	Obtaining parasitic networks.....	291
10.21.11	Persistent storage of load and interconnect models.....	292
10.21.12	Calculating effective capacitances and driving resistances.....	295
10.21.13	Parasitic estimation.....	298
10.21.14	Threshold voltages.....	303
10.21.15	Obtaining aggressor window overlaps.....	304
10.22	Noise analysis.....	311
10.22.1	Types of noise.....	312
10.22.2	Noise models.....	313
10.22.3	Noise waveforms.....	315
10.22.4	Noise network models.....	322
10.22.5	Calculating composite noise at cell inputs.....	327
10.22.6	Calculating composite noise at cell outputs.....	330
10.22.7	Setting noise budgets.....	334
10.22.8	Reporting noise violations.....	335
10.23	Delay and slew calculations for differential circuits.....	338
10.23.1	Sample figures.....	338
10.23.2	appGetArrivalOffsetsByName.....	339
10.23.3	API extensions for function modeling.....	340
10.23.4	Explicit APIs for user-defined primitives.....	348
10.23.5	APIs for hierarchy.....	350
10.23.6	Built-in APIs for function modeling.....	350
10.23.7	API Extensions for VECTOR modeling.....	351
10.23.8	APIs for XWF.....	352
10.23.9	Extensions and changes to voltages and temperature APIs.....	356
10.23.10	Operating conditions.....	358
10.23.11	On-chip process variation.....	360
10.23.12	Accessing properties and attributes.....	367
10.23.13	APIs for attribute within a PIN object.....	387
10.23.14	Connectivity.....	395
10.23.15	Control of timing arc existence and state.....	397
10.23.16	Modeling cores.....	402
10.23.17	Default pin slews and interface version calls.....	407
10.23.18	API to access library required resources.....	408
10.23.19	Resource types.....	410
10.23.20	Library extensions for phase locked loop processing.....	411
10.23.21	API definitions for external conditions.....	412
10.23.22	Extensions for listing pins.....	416
10.23.23	Memory BIST mapping.....	417
10.23.24	dpcmGetCellTestProcedure.....	419

10.24	Interconnect delay calculation intraface.....	419
10.24.1	Control and data flows.....	420
10.24.2	Model generation functions.....	421
10.24.3	Calculation functions.....	423
10.24.4	Cell calculation functions.....	424
10.24.5	ICM initialization.....	429
10.25	DCL run-time support.....	435
10.25.1	Array manipulation functions.....	435
10.25.2	Memory management.....	438
10.25.3	Structure manipulation functions.....	439
10.25.4	Initialization functions.....	443
10.26	Calculation functions.....	455
10.26.1	delay.....	455
10.26.2	slew.....	456
10.26.3	check.....	457
10.27	Modeling functions.....	459
10.27.1	modelSearch.....	459
10.27.2	Mode operators.....	461
10.27.3	Arrival time merging.....	462
10.27.4	Edge propagation communication to the application.....	462
10.27.5	Edge propagation communication to the DPCM.....	466
10.27.6	newTimingPin.....	466
10.27.7	newDelayMatrixRow.....	467
10.27.8	newNetSinkPropagateSegments.....	468
10.27.9	newNetSourcePropagateSegments.....	470
10.27.10	newPropagateSegment.....	471
10.27.11	newTestMatrixRow.....	471
10.27.12	newAllTestSegment.....	472
10.27.13	Interactions between interconnect modeling and modeling functions.....	473
10.28	Deprecated functions.....	473
10.28.1	Parasitic handling.....	474
10.28.2	Array manipulation functions.....	482
10.28.3	Memory management.....	484
10.28.4	Initialization functions.....	487
10.29	Standard Structure (std_stru.h) file.....	499
10.30	Standard macros (std_macros.h) file.....	519
10.31	Standard interface structures (dcmintf.h) file.....	527
10.32	Standard loading (dcmload.h) file.....	531
10.33	Standard debug (dcmdebug.h) file.....	534
10.34	Standard array (dcmgarray.h) file.....	561
10.35	Standard user array defines (dcmuarray.h) file.....	566
10.36	Standard platform-dependency (dcmpltfm.h) file.....	570
10.37	Standard state variables (dcmstate.h) file.....	576
11	Parasitics.....	580
11.1	Introduction.....	580
11.2	Targeted applications for SPEF.....	580
11.3	SPEF specification.....	580
11.3.1	Grammar.....	580
11.3.2	Escaping rules.....	582
11.3.3	File syntax.....	583
11.3.4	Comments.....	589
11.3.5	File semantics.....	589
11.4	Examples.....	609
11.4.1	Basic *D_NET file.....	609
11.4.2	Basic *R_NET file.....	612
11.4.3	*R_NET with poles and residues plus name mapping.....	613

11.4.4	*D_NET with triplet par_value.....	615
11.4.5	*R_NET with poles and residues plus triplet par_value.....	618
11.4.6	Merging SPEF files.....	619
11.4.7	A SPEF file header section with *VARIATION_PARAMETERS definition.....	624
11.4.8	CAP and RES statements with sensitivity information in a SPEF file.....	624
<b>Annex A</b>	(normative) Implementation requirements.....	625
<b>Annex B</b>	(informative) IEEE List of Participants.....	629

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

[IEC 61523-1:2012](https://standards.iteh.ai/catalog/standards/sist/0c1ed01e-318e-4f0f-ba05-dac666cce572/iec-61523-1-2012)

<https://standards.iteh.ai/catalog/standards/sist/0c1ed01e-318e-4f0f-ba05-dac666cce572/iec-61523-1-2012>

## Table of Tables

Table 1—Keywords.....	20
Table 2—DCL predefined references to Standard Structure fields.....	23
Table 3—DCL compiler generated predefined identifiers.....	27
Table 4—Edge types and conversions .....	29
Table 5—Propagation mode conversions.....	29
Table 6—Calculation mode conversions.....	29
Table 7—TEST_TYPE conversions.....	30
Table 8—Purity operator.....	45
Table 9—Timing resolution modes.....	45
Table 10—Test mode operators table.....	46
Table 11—Mathematical operators.....	51
Table 12—Logical operators.....	53
Table 13—Mathematical operator precedence (high to low).....	56
Table 14—Logical operator precedence (high to low).....	56
Table 15—Type definition for ACTIVE_HISTORY_TYPE.....	75
Table 16—Permitted activityCode values.....	76
Table 17—Type definition for HISTORY_TYPE.....	76
Table 18—Rule history info message types.....	76
Table 19—Table History inform message types .....	77
Table 20—Permitted kind values.....	77
Table 21—LOAD_HISTORY_TYPE.....	77
Table 22—CELL_LIST_TYPE.....	78
Table 23—TECH_TYPE.....	78
Table 24—DELAY_REC_TYPE .....	78
Table 25—SLEW_REC_TYPE .....	78
Table 26—CHECK_REC_TYPE.....	79
Table 27—CCDB_TYPE.....	79
Table 28—CELL_DATA_TYPE.....	79
Table 29—CCDB_TYPE .....	79
Table 30—PIN_ASSOCIATION .....	80
Table 31—PATH_DATA_TYPE.....	80
Table 32—STD_STRUCT.....	80
Table 33—ARGV.....	81
Table 34—CONTROL_PARM.....	81
Table 35—Library function floor.....	96
Table 36—Library function ifloor .....	97
Table 37—Library function ceil.....	97
Table 38—Library Function iceil.....	97
Table 39—Library function rint.....	97
Table 40—Library function round .....	97
Table 41—Library function trunc .....	98
Table 42—Library function itrunc .....	98
Table 43—Library function map_tech_family .....	98
Table 44—Library function current_tech_type .....	98
Table 45—Library function subrule_tech_type .....	98
Table 46—Library function subrule_tech_type.....	99
Table 47—Library function get_technology_list.....	99
Table 48—Library function cos.....	99
Table 49—Library function sin .....	99
Table 50—Library function tan .....	99
Table 51—Library function new_plane .....	100
Table 52—Library function get_plane_name .....	100
Table 53—Library function get_space_name.....	100

Table 54—Library function <code>get_max_spaces</code> .....	100
Table 55—Library function <code>get_max_planes</code> .....	100
Table 56—Library function <code>get_space_coordinate</code> .....	101
Table 57—Library function <code>get_plane_coordinate</code> .....	101
Table 58—Library function <code>set_busy_wait</code> .....	101
Table 59—Library function <code>change_debug_level</code> .....	102
Table 60—Library function <code>get_caller_stack</code> .....	102
Table 61—Library function <code>GET_LOAD_HISTORY</code> .....	102
Table 62—Library function <code>GET_CELL_LIST</code> .....	102
Table 63—Library function <code>GET_ROW_COUNT</code> .....	103
Table 64—Library function <code>STEP_TABLE</code> .....	103
Table 65—Library function <code>GET_LOAD_PATH</code> .....	103
Table 66—Library function <code>GET_RULE_NAME</code> .....	104
Table 67—Library function <code>ADD_RULE</code> .....	104
Table 68—Data type clause.....	118
Table 69—Arc data types.....	119
Table 70—Validity of predefined identifiers for <code>STORE</code> clause.....	128
Table 71—Logic operators (valid for behavior, vectorTiming, and vectorPower model domains).....	133
Table 72—Logical equivalence operators (valid for behavior model domain).....	134
Table 73—Unary bitwise operators (valid for behavior, vectorTiming, and vectorPower model domains) .....	134
Table 74—Binary bitwise operators (valid for behavior, vectorTiming, and vectorPower model domains) .....	134
Table 75—Binary operators (valid for behavior model domain).....	134
Table 76—Node primitives for control operators (valid for behavior model domain).....	135
Table 77—Node primitives for edge operators (continued) (valid for behavior, vectorTiming, and vectorPower model domains).....	135
Table 78—Node primitives for precedence control operators (valid for behavior model domain).....	136
Table 79—Node primitives for constant operators (valid for behavior, vectorTiming, vectorPower model domains).....	136
Table 80—Node primitives for user-defined operators.....	136
Table 81—Node primitives for miscellaneous operators.....	137
Table 82—Binary reduction operators.....	138
Table 83—Bitwise reduction operators.....	139
Table 84—Logical reduction operators.....	140
Table 85—Array of bits operators.....	141
Table 86—Edge operators.....	142
Table 87—Higher function nodes.....	143
Table 88—Constant value nodes.....	146
Table 89—Miscellaneous operators .....	146
Table 90—User-defined operators.....	147
Table 91—Valid modifier enumerations for given node primitive operators.....	148
Table 92—Syntax for a <code>GroupPinString</code> .....	162
Table 93—Syntax for a <code>GroupConditionString</code> .....	163
Table 94—Syntax for a <code>SensitivityPinString</code> .....	164
Table 95—Syntax for a <code>condition_expression</code> .....	165
Table 96— <code>PinName_Identifier</code> semantics .....	167
Table 97— <code>PinName_Level</code> semantics.....	167
Table 98— <code>PinName_State</code> semantics.....	167
Table 99—Condition expression operators.....	168
Table 100—Interaction between multiple technologies and application.....	183
Table 101—Return code most significant byte.....	183
Table 102—Return code least significant bytes.....	184
Table 103—Data types defined in DCL and C.....	184
Table 104—Header files.....	185
Table 105—Predefined macro names.....	186

Table 106—Alternate semantics for Standard Structure fields.....	189
Table 107—Expose functions.....	191
Table 108—External functions.....	199
Table 109—Deprecated functions.....	203
Table 110—libdcmr functions.....	205
Table 111—Module control functions.....	206
Table 112—Memory control functions.....	207
Table 113—Message and error control functions.....	208
Table 114—Calculation functions.....	208
Table 115—Modeling functions.....	209
Table 116—PI function table example.....	209
Table 117—Standard Structure field semantics.....	210
Table 118—appGetTotalLoadCapacitanceByPin.....	211
Table 119—appGetTotalLoadCapacitanceByName.....	211
Table 120—appGetTotalPinCapacitanceByPin.....	212
Table 121—appGetTotalPinCapacitanceByName.....	212
Table 122—appGetSourcePinCapacitanceByPin.....	213
Table 123—appGetSourcePinCapacitanceByName.....	213
Table 124—dpcmGetDefCellSize.....	214
Table 125—appGetCellCoordinates.....	214
Table 126—appGetCellOrientation.....	215
Table 127—dpcmGetEstLoadCapacitance.....	215
Table 128—dpcmGetEstWireCapacitance.....	216
Table 129—dpcmGetEstWireResistance.....	216
Table 130—dpcmGetPinCapacitance.....	217
Table 131—dpcmGetCellIOLists.....	217
Table 132—appGetRC.....	218
Table 133—dpcmGetDelayGradient.....	219
Table 134—dpcmGetSlewGradient.....	219
Table 135—dpcmGetEstimateRC.....	220
Table 136—dpcmGetDefPortSlew.....	220
Table 137—dpcmGetDefPortCapacitance.....	221
Table 138—appGetNumDriversByPin.....	221
Table 139—appGetNumDriversByName.....	222
Table 140—appForEachParallelDriverByPin.....	222
Table 141—appForEachParallelDriverByName.....	224
Table 142—appGetNumPinsByPin.....	225
Table 143—appGetNumPinsByName.....	225
Table 144—appGetNumSinksByPin.....	226
Table 145—appGetNumSinksByName.....	226
Table 146—dpcmAddWireLoadModel.....	227
Table 147—dpcmGetWireLoadModel.....	228
Table 148—dpcmGetWireLoadModelForBlockSize.....	229
Table 149—appGetInstanceCount.....	229
Table 150—dpcmGetCapacitanceLimit.....	230
Table 151—dpcmGetSlewLimit.....	230
Table 152—dpcmGetXovers.....	231
Table 153—dpcmGetFunctionalModeArray.....	231
Table 154—dpcmGetBaseFunctionalMode.....	232
Table 155—appGetCurrentFunctionalMode.....	233
Table 156—dpcmGetControlExistence.....	233
Table 157—dpcmSetLevel.....	234
Table 158—dpcmGetLibraryAccuracyLevelArrays.....	235
Table 159—dpcmSetLibraryAccuracyLevel.....	236
Table 160—dpcmGetExposePurityAndConsistency.....	236
Table 161—DCM_Purity.....	237



Table 162—DCM_Consistency.....	237
Table 163—dpcmGetRailVoltageArray.....	237
Table 164—dpcmGetBaseRailVoltage.....	238
Table 165—appGetCurrentRailVoltage.....	238
Table 166—dpcmGetWireLoadModelArray.....	239
Table 167—dpcmGetBaseWireLoadModel.....	239
Table 168—appGetCurrentWireLoadModel.....	240
Table 169—dpcmGetBaseTemperature.....	240
Table 170—dpcmGetBaseOpRange.....	241
Table 171—dpcmGetOpRangeArray.....	241
Table 172—appGetCurrentTemperature.....	242
Table 173—appGetCurrentOpRange.....	242
Table 174—dpcmGetTimingStateArray.....	243
Table 175—appGetCurrentTimingState.....	244
Table 176—dpcmGetCellList.....	244
Table 177—appGetCellName.....	245
Table 178—appGetHierPinName.....	245
Table 179—appGetHierBlockName.....	246
Table 180—appGetHierNetName.....	246
Table 181—dpcmGetThresholds.....	246
Table 182—appGetThresholds.....	247
Table 183—appGetExternalStatus.....	248
Table 184—appGetVersionInfo.....	248
Table 185—appGetResource.....	249
Table 186—dpcmGetRuleUnitToSeconds.....	249
Table 187—dpcmGetRuleUnitToOhms.....	249
Table 188—dpcmGetRuleUnitToFarads.....	250
Table 189—dpcmGetRuleUnitToHenries.....	251
Table 190—dpcmGetRuleUnitToWatts.....	251
Table 191—dpcmGetRuleUnitToJoules.....	252
Table 192—dpcmGetTimeResolution.....	252
Table 193—dpcmGetParasiticCoordinateTypes.....	253
Table 194—dpcmIsSlewTime.....	253
Table 195—dpcmDebug.....	254
Table 196—dpcmGetVersionInfo.....	254
Table 197—dpcmHoldControl.....	255
Table 198—dpcmFillPinCache.....	255
Table 199—dpcmFreePinCache.....	256
Table 200—appRegisterCellInfo.....	256
Table 201—dpcmGetCellPowerInfo.....	257
Table 202—dpcmGetCellPowerWithState.....	258
Table 203—dpcmGetAETCellPowerWithSensitivity.....	259
Table 204—Integer LSB example.....	259
Table 205—Mask encoding.....	259
Table 206—dpcmGetPinPower.....	260
Table 207—dpcmAETGetSettlingTime.....	260
Table 208—dpcmAETGetSimultaneousSwitchTime.....	261
Table 209—dpcmGroupGetSettlingTime.....	261
Table 210—dpcmGroupGetSimultaneousSwitchTime.....	262
Table 211—dpcmCalcPartialSwingEnergy.....	262
Table 212—dpcmSetInitialState.....	263
Table 213—dpcmFreeStateCache.....	264
Table 214—appGetStateCache.....	264
Table 215—dpcmGetNetEnergy.....	265
Table 216—parasiticElement structure.....	270
Table 217—Parasitic element variables.....	271