

# INTERNATIONAL STANDARD

# IEC 61523-1

First edition  
2001-09

---

---

## Delay and power calculation standards –

### Part 1: Integrated circuit delay and power calculation systems

© IEC 2001 — Copyright - all rights reserved

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 publisher.

International Electrotechnical Commission 3, rue de Varembé Geneva, Switzerland  
Telefax: +41 22 919 0300 e-mail: [inmail@iec.ch](mailto:inmail@iec.ch) IEC web site <http://www.iec.ch>



Commission Electrotechnique Internationale  
International Electrotechnical Commission  
Международная Электротехническая Комиссия

PRICE CODE **XP**

*For price, see current catalogue*

# Contents

SECTION	PAGE
<b>1. Overview</b> .....	<b>13</b>
1.1 Scope .....	13
1.2 Purpose .....	13
1.3 Contents of this standard .....	14
<b>2. References</b> .....	<b>15</b>
<b>3. Definitions</b> .....	<b>16</b>
<b>4. Acronyms and abbreviations</b> .....	<b>24</b>
<b>5. Delay and power calculation system architecture</b> .....	<b>25</b>
5.1 Overview .....	25
5.2 Procedural interface .....	26
5.2.1 Global policies and conventions .....	26
5.2.2 Flow of control .....	27
5.3 DPCM - application relationships .....	27
5.3.1 Technology library .....	27
5.3.2 Subrule .....	28
5.4 Inter-operability .....	28
<b>6. Delay Calculation Language (DCL)</b> .....	<b>29</b>
6.1 Character set .....	30
6.2 Lexical elements .....	30
6.2.1 Whitespace .....	30
6.2.2 Comments .....	30
6.2.3 Tokens .....	30
6.2.4 Header names .....	40
6.2.5 Preprocessing directives .....	40
6.3 Name spaces of identifiers .....	41
6.4 Storage durations of objects .....	41
6.5 Scope of identifiers .....	41
6.6 Linkages of identifiers .....	42
6.6.1 EXPORT .....	42
6.6.2 IMPORT .....	42
6.6.3 FORWARD .....	42
6.6.4 Chaining of EXPOSE identifiers .....	42
6.7 DCL data types .....	42
6.7.1 Native data types .....	42
6.7.2 Array types .....	43
6.7.3 Derived data types .....	44
6.8 Type conversions .....	45
6.8.1 Implicit conversions .....	46
6.8.2 Explicit conversions .....	46
6.9 Operators .....	46
6.9.1 String prefix operator .....	47
6.9.2 Assignment operator .....	47
6.9.3 New operator .....	47

6.9.4	SCOPE operator	47
6.9.5	Purity operator	48
6.9.6	Timing propagation	48
6.9.7	Timing checks	48
6.9.8	Test mode operators	49
6.10	Expressions	51
6.10.1	Array subscripting	51
6.10.2	Statement calls	52
6.10.3	Assign variable reference	53
6.10.4	Store variable reference	53
6.10.5	Mathematical expressions	53
6.10.6	Logical expressions and operators	55
6.10.7	Pin range	56
6.10.8	Embedded C code expressions	58
6.11	Computation order	59
6.11.1	Mathematical expressions	59
6.11.2	Logical expressions	59
6.11.3	Passed parameters	60
6.11.4	WHEN clause	60
6.11.5	REPEAT - UNTIL clause	60
6.12	DCL statements	60
6.12.1	Clauses	60
6.12.2	Modifiers	63
6.12.3	Prototypes	64
6.12.4	Statement failure	67
6.12.5	Interfacing statements	68
6.12.6	Calculation statements	69
6.12.7	METHOD statement	73
6.13	Tables	74
6.13.1	TABLEDEF statement	74
6.13.2	Table visibility rules	76
6.13.3	TABLE statement	77
6.13.4	Static tables	77
6.13.5	Dynamic tables	79
6.13.6	Dynamic table manipulation	80
6.13.7	Lookup table	84
6.14	Library control statements	87
6.14.1	Meta-variables	87
6.14.2	SUBRULE statement	87
6.14.3	SUBRULES statement	89
6.14.4	TECH_FAMILY statement	91
6.15	Modeling	92
6.15.1	Model organization	92
6.15.2	MODELPROC statement	93
6.15.3	SUBMODEL statement	95
6.15.4	Modeling statements	95
6.16	Embedded C code	112
6.17	Definition of a subrule	112
<b>7.</b>	<b>Power modeling and calculation</b>	<b>114</b>
7.1	Power overview	115
7.2	Caching state information	116
7.2.1	Initializing the state cache	116

7.2.2	State cache lifetime	116
7.3	Caching load and slew information	116
7.3.1	Loading the load and slew cache	117
7.3.2	Load and slew cache lifetime	117
7.4	Simultaneous switching events	117
7.5	Partial swing events	118
7.6	Power calculation	118
7.7	Accumulation of power consumption by the design	120
7.8	Group pin list syntax and semantics	120
7.8.1	Syntax	120
7.8.2	Semantics	121
7.8.3	Example	121
7.9	Group condition list syntax and semantics	121
7.9.1	Syntax	122
7.9.2	Semantics	122
7.9.3	Example	122
7.10	Sensitivity list syntax and semantics	123
7.10.1	Syntax	123
7.10.2	Semantics	123
7.10.3	Example	123
7.11	Group condition language	124
7.11.1	Syntax	124
7.11.2	Semantics	124
7.11.3	Condition expression operator precedence	126
7.11.4	Condition expressions referencing pin states and transitions	127
7.11.5	Semantics of nonexistent pins	127
<b>8.</b>	<b>Procedural Interface (PI)</b>	<b>129</b>
8.1	Overview	129
8.1.1	DPCM	129
8.1.2	Application	129
8.1.3	libdcmlr	129
8.2	Control and data flow	130
8.3	Architectural requirements	130
8.4	Data ownership technique	130
8.4.1	Persistence of data passed across the PI	130
8.4.2	Data cache guidelines for the DPCM	131
8.5	Application/DPCM interaction	131
8.5.1	Application initializes message/memory handling	131
8.5.2	Application loads and initializes the DPCM	131
8.5.3	Application requests timing models for cell instances	132
8.5.4	Model domain issues	132
8.5.5	DPCM invokes application modeling callback functions	132
8.5.6	Application requests propagation delay	133
8.5.7	DPCM calls application EXTERNAL functions	134
8.6	Re-entry requirements	134
8.7	Application responsibilities when using a DPCM	134
8.7.1	Standard structure rules	134
8.7.2	User object registration	134
8.7.3	Selection of early and late slew values	134
8.8	Application use of the DPCM	136
8.8.1	Initialization of the DPCM	136
8.8.2	Use of the DPCM	137

8.8.3	Termination of DPCM	138
8.9	DPCM library organization	138
8.9.1	Multiple technologies	138
8.9.2	Model names	139
8.10	DPCM error handling	139
8.11	C level language for EXPOSE and EXTERNAL functions	139
8.11.1	Integer return code	139
8.11.2	The Standard Structure pointer	140
8.11.3	Result structure pointer	140
8.11.4	Passed arguments	140
8.11.5	DCL array indexing	141
8.11.6	Conversion to C data types	141
8.11.7	include files	141
8.12	PIN and BLOCK data structure requirements	142
8.13	DCM_STD_STRUCT Standard Structure	143
8.13.1	Alternate semantics for Standard Structure fields	146
8.13.2	Reserved fields	146
8.13.3	Standard Structure value restriction	146
8.14	DCMTransmittedInfo structure	147
8.15	Environment or user variables	147
8.16	PI functions summary	147
8.16.1	Expose functions	147
8.16.2	External functions	150
8.16.3	Implicit functions	152
8.16.4	PI function table description	155
8.17	PI function descriptions	157
8.17.1	Interconnect loading related functions	157
8.17.2	Interconnect delay related functions	153
8.17.3	Functions accessing netlist information	156
8.17.4	Functions exporting limit information	187
8.17.5	Functions getting/setting model information	189
8.17.6	Functions importing instance name information	200
8.17.7	Process information functions	203
8.17.8	Miscellaneous standard interface functions	204
8.17.9	Power related functions	214
8.17.10	Array manipulation functions	226
8.17.11	Initialization functions	230
8.17.12	Calculation functions	241
8.17.13	Modeling functions	246
8.18	86Standard structure (dcmstd_stru.h) file	258
8.19	Standard macros (dcmstd_macros.h) file	274
8.20	Standard interface structures (dcmintf.h) file	277
8.21	Standard loading (dcmload.h) file	281
8.22	Standard debug (dcmdebug.h) file	283
8.23	Standard array (dcmgarray.h) file	298
8.24	DCM user array defines (dcmuarray.h) file	302
8.25	Standard platform-dependency (dcmpltfm.h) file	304
8.26	Standard state variables (dcmstate.h) file	311
8.27	Standard table descriptor(dcmstab.h)	313
<b>9.</b>	<b>Parasitics</b>	<b>314</b>
9.1	Introduction	315
9.2	Targeted applications for SPEF	315

9.3	SPEF specification . . . . .	315
9.3.1	Grammar . . . . .	315
9.3.2	File syntax . . . . .	317
9.3.3	Escaping rules . . . . .	324
9.3.4	Comments . . . . .	325
9.3.5	File semantics . . . . .	325
9.4	Examples . . . . .	341
9.4.1	Basic D_NET file . . . . .	341
9.4.2	Basic R_NET file . . . . .	344
9.4.3	R_NET with poles and residues plus name mapping . . . . .	345
9.4.4	D_NET with triplet par_value . . . . .	347
9.4.5	R_NET with poles and residues plus triplet par_value . . . . .	350
9.4.6	Merging SPEF files . . . . .	351
<b>10.</b>	<b>Physical design exchange . . . . .</b>	<b>357</b>
10.1	Introduction . . . . .	358
10.1.1	Scope . . . . .	358
10.1.2	Targeted applications . . . . .	358
10.2	PDEF specification . . . . .	359
10.2.1	PDEF grammar . . . . .	359
10.2.2	PDEF file syntax . . . . .	361
10.2.3	Comments within a PDEF file . . . . .	365
10.2.4	PDEF name semantics . . . . .	365
10.2.5	PDEF file semantics . . . . .	368
10.2.6	Attributes . . . . .	379
10.3	Examples . . . . .	399
10.3.1	Escaping . . . . .	400
10.3.2	Clusters . . . . .	400
10.3.3	Global routing . . . . .	404
10.3.4	Symbolic placement constraints using PDEF . . . . .	405
<b>A.</b>	<b>Implementation requirements . . . . .</b>	<b>408</b>
<b>B.</b>	<b>Calculation of total load capacitance in the DPCS . . . . .</b>	<b>410</b>
<b>C.</b>	<b>Hold Control . . . . .</b>	<b>413</b>
<b>D.</b>	<b>Bibliography . . . . .</b>	<b>418</b>

## List of Figures

FIGURE	PAGE
5-1	High-level DPCS architecture . . . . . 25
5-2	High-level DPCS architecture linkage structure . . . . . 27
8-1	DPCM/application procedural interface . . . . . 130
8-2	PIN and PINLIST . . . . . 143
8-3	PI function table example . . . . . 156
8-4	Parallel drivers example . . . . . 169
8-5	Capacitance value example . . . . . 174
8-6	Passed and receiver pin pointers example . . . . . 185
8-7	Integer LSB example . . . . . 216
8-8	Bias calculation . . . . . 245
8-9	Clock separation . . . . . 245
8-10	Different edges . . . . . 246
8-11	Sample MODELPROC results . . . . . 254
8-12	Additional MODELPROC results . . . . . 255
9-1	SPEF targeted applications . . . . . 315
10-1	PDEF targeted applications . . . . . 359
10-2	Hierarchical routes . . . . . 376
10-3	Clusters cross logic hierarchy . . . . . 401
10-4	Illustration of the chip used in the example . . . . . 401
10-5	A global route . . . . . 405
10-6	Symbolic cell and cluster placement . . . . . 406
10-7	Symbolic pin and net placement . . . . . 407
C.1	Transparent latch pair feedback . . . . . 413
C.2	Overlap period . . . . . 414
C.3	Padding the paths . . . . . 415
C.4	Stable overlap period . . . . . 415
C.5	Hold Control modeling . . . . . 416
C.6	Snipped feedback line . . . . . 417

## List of Tables

TABLE	PAGE
6-1	Keywords . . . . . 32
6-2	DCL predefined references to Standard Structure fields . . . . . 33
6-3	DCL compiler generated predefined identifiers . . . . . 37
6-4	Edge types and conversions . . . . . 38
6-5	Propagation mode conversions . . . . . 38
6-6	Calculation mode conversions. . . . . 39
6-7	TEST_TYPE conversions . . . . . 39
6-8	VAR array modifier. . . . . 44
6-9	Purity operator. . . . . 48
6-10	Timing resolution modes. . . . . 48
6-11	Test mode operators. . . . . 49
6-12	Mathematical operators. . . . . 54
6-13	Logical operators. . . . . 56
6-14	Mathematical operator precedence (high to low) . . . . . 59
6-15	Logical operator precedence (high to low) . . . . . 59
6-16	Validity of predefined identifiers for STORE clause . . . . . 107
7-1	PinName_Identifier semantics . . . . . 125
7-2	PinName_Level semantics . . . . . 126
7-3	PinName_State semantics . . . . . 126
7-4	Condition expression operators. . . . . 127
8-1	Interaction between multiple technologies and application . . . . . 139
8-2	Return code most significant byte. . . . . 140
8-3	Return code least significant bytes . . . . . 140
8-4	Data types defined in DCL and C . . . . . 141
8-5	Header files . . . . . 142
8-6	Predefined macro names . . . . . 143
8-7	Alternate semantics for Standard Structure fields. . . . . 146
8-8	EXPOSE functions . . . . . 147
8-9	EXTERNAL functions . . . . . 150
8-10	libdcmlr functions . . . . . 152
8-11	Run-time library . . . . . 153
8-12	Calculation functions. . . . . 155
8-13	Modeling functions . . . . . 155
8-14	Standard Structure field semantics . . . . . 156
8-15	Mask encoding . . . . . 217
8-16	Mode propagation operators . . . . . 248
8-17	Mode computation operators for delay and slew . . . . . 248
8-18	Mode operator enumerators for check . . . . . 249
8-19	Enumeration pairs . . . . . 249
8-20	Edge propagation communication with DPCM . . . . . 251
9-1	Design flow values . . . . . 326
9-2	conn_attr types . . . . . 330
10-1	Netlist type identifiers . . . . . 369
10-2	Design flow identifiers . . . . . 370
10-3	Gate type attributes . . . . . 381
10-4	Gate restriction attributes . . . . . 382

- 10-5 Pin and gate\_pin attributes ..... 384
- 10-6 Node type attributes..... 385
- 10-7 Net type attributes ..... 387
- 10-8 Net restriction attributes ..... 388
- 10-9 Route restriction attributes ..... 388
- 10-10 Route connectivity attributes..... 389
- 10-11 Cluster restriction attributes ..... 391
- 10-12 Cluster connectivity attributes..... 394
- 10-13 Cell type attributes..... 395
- 10-14 Cell restriction attributes..... 396
- 10-15 Spare cell restriction attributes ..... 397
- 10-16 Pbus and nbus type attributes ..... 398
- 10-17 Pnet type attributes ..... 398
- 10-18 Pnet restriction attribute pre-defined values ..... 399
- 10-19 Gap and skip type attributes ..... 399

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**DELAY AND POWER CALCULATION STANDARDS –**

**Part 1: Integrated circuit delay and power calculation systems**

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the 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, the IEC publishes International Standards. 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. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the 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 National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61523-1 has been prepared by IEC technical committee 93: Design automation.

This standard is based on IEEE Std P1481 (1999): *IEEE Standard for delay and power calculation systems*.

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

FDIS	Report on voting
93/143/FDIS	93/144/RVD

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

*This standard does not follow the rules for the structure of international standards given in Part 3 of the ISO/IEC Directives.*

IEC 61523 consists of the following parts, under the general title: *Delay and calculation standards*:

IEC 61523-1, Part 1: *Integrated circuit delay and power calculation systems*

IEC 61523-2, Part 2: *Prelayout delay calculation model specification of CMOS ASIC libraries* (to be published)

The committee has decided that the contents of this publication will remain unchanged until 2006. At this date, the publication will be

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

## Introduction

The objective of the Delay and Power Calculation System (DPCS) is to make it possible for integrated circuit designers to *consistently* calculate chip delay and power across Electronic Design Automation (EDA) applications and for integrated circuit vendors to express delay and power information only *once* per technology while enabling sufficient EDA application accuracy.

This is accomplished by a coordinated set of standards which support a standard method to describe timing and power characteristics of integrated circuit design units (cells and higher level design elements); a standard method for EDA applications to calculate chip design instance specific delay, slew, and power for logic and interconnects; and standard file formats to exchange chip parasitic and cluster information.

A major integrated circuit design problem is to ensure the chip meets the designer's timing and power requirements. At this time in the EDA industry, there are numerous methodologies — many incompatible or inconsistent with each other — for representing timing and power information. The DPCS responds to the urgency to improve chip design methods.

The DPCS includes all aspects and uses of delay and power calculation (including synthesis, floorplanning, and simulation) during both pre-layout and post-layout phases of chip design. A fundamental requirement placed on the DPCS is that it work equally well with Verilog HDL and VHDL. In this draft of the standard, the scope of the DPCS is limited to integrated circuit designs.

The DPCS gains its leverage out of an effort by Silicon Integration Initiative (Si2) to standardize on contributed technology from IBM in the form of a Delay Calculation Language (DCL) and its companion procedural interface (PI), along with an effort by Open Verilog International (OVI) to standardize on file formats to exchange chip parasitic and cluster information in the form of a Physical Design Exchange Format (PDEF) and a Standard Parasitic Exchange Format (SPEF). SPEF is based on the Standard Parasitic Format (SPF) technology supplied by Cadence Design Systems. PDEF is based on technology provided by Synopsys.

Since each of these specifications is based on existing technology which has proven itself in industry, the time to introduce this standard into industry is greatly reduced over what would have been required for a newly created standard. The industry owes a great deal of gratitude to these companies for contributing these technologies.

The P1481 Working Group is composed of four subgroups: Architecture, Language, Parasitics and Clustering, and Power. The members of each of these groups are members of the P1481 Working Group. The P1481 Working Group has a vice-chair liaison with Japan to coordinate development and review of the specification there. The IEC TC93 WG 2 also participates in the review and development of this specification.

In addition to numerous individual contributors, both OVI, VHDL International (VI), and Si2 have played significant roles in the development of this standard and funding of the technical documentation.

## DELAY AND POWER CALCULATION STANDARDS –

### Part 1: Integrated circuit delay and power calculation systems

#### 1 Overview

The Delay and Power Calculation System (DPCS) standard is a coordinated set of standards which support a standard method to describe timing and power characteristics of integrated circuit design units (cells and higher level design elements); a standard method for EDA applications to calculate chip design instance specific delay, slew, and power for logic and interconnects; and standard file formats to exchange chip parasitic and cluster information. The four distinct standard specifications covered in this document include:

- A description language for timing and power modeling (DCL).
- A software procedural interface (PI) for communications between EDA applications and compiled libraries of DCL descriptions.
- A standard file exchange format for parasitic information about the chip design (SPEF).
- A standard file exchange format for floorplan cluster information relative to the chip design (PDEF).

#### 1.1 Scope

As stated in the introduction, the scope of the DPCS standard is to make it possible for integrated circuit designers to analyze chip timing and power consistently across a broad set of EDA applications, for integrated circuit vendors to express timing and power information once (for a given technology), and for EDA vendors to meet their application performance and capacity needs. The intended use for these standards is integrated circuit timing and power. The standard may be applied to both unit logic cells supplied by the integrated circuit vendor and logical macros defined by the integrated circuit designer. Although this specification is written towards the integrated circuit supplier and EDA developer, its application applies equally well to representation of timing and power for designer defined macros (or hierarchical design elements).

These specifications make it *possible* to achieve consistent timing and power results, but do not *guarantee* it. They provide for a single executable software program which computes delay and power based on integrated circuit vendor-supplied algorithms (or designer-supplied algorithms for macros), but does not guarantee EDA applications correctly communicate the design-specific information required for these algorithms. By specifying standard exchange formats for parasitic data and floorplanning information, the standard provides a marked improvement over design environments with no such standards. However, it is the responsibility of the EDA application to correctly correlate the information between these standard exchange files and the actual design. These specifications also do not detail how the information contained within the standard exchange files shall be obtained.