

INTERNATIONAL STANDARD

NORME INTERNATIONALE



EMC IC modelling –

Part 2: Models of integrated circuits for EMI behavioural simulation – Conducted emissions modelling (ICEM-CE)

Modèles de circuits intégrés pour la CEM –

Partie 2: Modèles de circuits intégrés pour la simulation du comportement lors de perturbations électromagnétiques – Modélisation des émissions conduites (ICEM-CE)

IECNORM.COM : Click to view the full PDF of IEC 62433-2:2017



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2017 IEC, Geneva, Switzerland

All rights reserved. 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 either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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
www.iec.ch

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.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

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 also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

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: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

65 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



EMC IC modelling –

Part 2: Models of integrated circuits for EMI behavioural simulation – Conducted emissions modelling (ICEM-CE)

Modèles de circuits intégrés pour la CEM –

Partie 2: Modèles de circuits intégrés pour la simulation du comportement lors de perturbations électromagnétiques – Modélisation des émissions conduites (ICEM-CE)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

31.200; 33.100.10

ISBN 978-2-8322-3876-9

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

Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

CONTENTS

FOREWORD.....	7
1 Scope.....	9
2 Normative references	9
3 Terms, definitions, abbreviations and conventions	9
3.1 Terms and definitions.....	9
3.2 Abbreviations.....	11
3.3 Conventions.....	11
4 Philosophy.....	11
4.1 General.....	11
4.2 Conducted emission from core activity (digital culprit).....	12
4.3 Conducted emission from I/O activity	12
4.4 Data exchange format.....	12
5 ICEM-CE basic components	13
5.1 General.....	13
5.2 Internal Activity (IA)	13
5.2.1 General	13
5.2.2 Examples of IA	14
5.3 Passive Distribution Network (PDN).....	14
5.3.1 General	14
5.3.2 Examples of PDN	15
6 IC macro-models	16
6.1 Types of IC macro-models	16
6.2 General IC macro-model	16
6.3 Block-based IC macro-model	17
6.3.1 Block component	17
6.3.2 Inter-Block Coupling component (IBC)	18
6.3.3 Block-based IC macro-model structure	19
6.4 Sub-model-based IC macro-model	21
6.4.1 Sub-model component.....	21
6.4.2 Sub-model-based IC macro-model structure	22
7 CEML format	23
7.1 General.....	23
7.2 CEML structure	24
7.3 Global keywords	24
7.4 Header section.....	24
7.5 Lead definitions	25
7.6 SPICE macro-models.....	26
7.7 Validity section	28
7.7.1 General	28
7.7.2 Attribute definitions.....	29
7.8 PDN.....	31
7.8.1 General	31
7.8.2 Attribute definitions.....	32
7.8.3 Description	36
7.9 IBC	40
7.9.1 General	40

7.9.2	Attribute definitions	40
7.10	IA	42
7.10.1	General	42
7.10.2	Attribute definitions	42
7.10.3	Description	46
8	Requirements for parameter extraction	47
8.1	General	47
8.2	Environmental extraction constraints	47
8.3	IA parameter extraction	47
8.4	PDN parameter extraction	47
8.5	IBC parameter extraction	48
Annex A	(normative) Preliminary definitions for XML representation	49
A.1	XML basics	49
A.1.1	XML declaration	49
A.1.2	Basic elements	49
A.1.3	Root element	49
A.1.4	Comments	50
A.1.5	Line terminations	50
A.1.6	Element hierarchy	50
A.1.7	Element attributes	50
A.2	Keyword requirements	50
A.2.1	General	50
A.2.2	Keyword characters	51
A.2.3	Keyword syntax	51
A.2.4	File structure	51
A.2.5	Values	53
Annex B	(normative) CEML valid keywords and usage	56
B.1	Root element keywords	56
B.2	File header keywords	57
B.3	Validity section keywords	58
B.4	Global keywords	59
B.5	Lead Keyword	59
B.6	Lead definitions section attributes	60
B.7	Macromodels section attributes	60
B.8	Pdn section keywords	61
B.8.1	Lead element keywords	61
B.8.2	Netlist section keywords	63
B.9	Ibc section keywords	63
B.9.1	Lead element keywords	63
B.9.2	Netlist section keywords	65
B.10	Ia section keywords	65
B.10.1	Lead element keywords	65
B.10.2	Voltage section keywords	66
B.10.3	Current section keywords	68
Annex C	(informative) Example of ICEM-CE macro-model in CEML format	70
C.1	General	70
C.2	PDN and IBC sub-model	70
C.3	IA sub-model	71

C.4	Frequency domain ICEM-CE in CEML.....	73
C.5	Time domain ICEM-CE in CEML	75
Annex D (informative)	Conversions between parameter types	77
D.1	General.....	77
D.2	Conversion for one-port PDN	77
D.3	Conversion for two-port PDN	77
Annex E (informative)	Model parameter generation.....	79
E.1	General.....	79
E.2	Default structure and values	79
E.2.1	General	79
E.2.2	IA parameters.....	79
E.2.3	PDN parameters	80
E.3	Model parameter generation from design information	81
E.3.1	General	81
E.3.2	IA parameters.....	81
E.3.3	PDN parameters	85
E.4	Model parameter generation from measurements.....	87
E.4.1	IA parameters.....	87
E.4.2	PDN parameters	90
Annex F (informative)	Decoupling capacitors optimization.....	100
Annex G (informative)	Conducted emission prediction	102
Annex H (informative)	Conducted emission prediction at PCB level	103
Bibliography		105
Figure 1	– Decomposition example of a digital IC for conducted emissions analysis	12
Figure 2	– IA component in the case of a current source.....	13
Figure 3	– Example of IA characteristics in the time domain.....	14
Figure 4	– Example of IA characteristics in the frequency domain.....	14
Figure 5	– Example of a four-terminal PDN using lumped elements	15
Figure 6	– Example of a seven-terminal PDN using distributed elements	16
Figure 7	– Example of a twelve-terminal PDN using matrix representation	16
Figure 8	– General IC macro-model	17
Figure 9	– Example of block component with a single IA	18
Figure 10	– Example of block components for I/Os	18
Figure 11	– Example of IBC with four internal terminals	19
Figure 12	– Relationship between blocks and IBC.....	19
Figure 13	– Block-based IC macro-model.....	20
Figure 14	– Example of block-based IC macro-model.....	21
Figure 15	– Example of simple sub-model.....	21
Figure 16	– Sub-model-based IC macro-model	22
Figure 17	– CEML inheritance hierarchy	23
Figure 18	– Example of a netlist file defining a sub-circuit.....	28
Figure 19	– PDN represented as S-parameters in Touchstone format.....	38
Figure 20	– Simulated IA waveform with corresponding parameters.....	45
Figure A.1	– Multiple XML (CEML) files.....	52

Figure A.2 – XML files with data files (*.dat)	52
Figure A.3 – XML files with additional files	53
Figure C.1 – Example pin-out of a microcontroller and the modelled pins	70
Figure C.2 – PDN sub-model topology	71
Figure C.3 – IA sub-model topology	72
Figure C.4 – IA of digital block in frequency domain	72
Figure C.5 – IA of digital block in time domain	73
Figure E.1 – Typical characterization current gate schematic	82
Figure E.2 – Current peak during switching transition	82
Figure E.3 – Example of IA extraction procedure from design	83
Figure E.4 – Technology Influence	83
Figure E.5 – Final current waveform for a program period	84
Figure E.6 – Comparison between measurement and simulation	84
Figure E.7 – Example lumped element model of a package	85
Figure E.8 – Circuit structure of the netlist	87
Figure E.9 – Principle of the IA computation in the frequency domain	88
Figure E.10 – Process involved to model $i_A(t)$	89
Figure E.11 – $i_{Ext}(t)$ measured using IEC 61967-4	89
Figure E.12 – $i_A(t)$ and $i_{Ext}(t)$ profiles	90
Figure E.13 – Conventional one-port S -parameter measurement	90
Figure E.14 – Two-port method for low impedance measurement	91
Figure E.15 – Two-port method for high impedance measurement	91
Figure E.16 – Example of a hardware set-up used to extract the PDN parameters	92
Figure E.17 – Miniature 50 Ω coaxial connectors	93
Figure E.18 – Impedance probe using two miniature coaxial connectors	93
Figure E.19 – Open and short terminations	93
Figure E.20 – Measurement probe model	94
Figure E.21 – De-embedding principle	94
Figure E.22 – Example of a predefined PDN structure	95
Figure E.23 – RL configuration	96
Figure E.24 – RLC configuration	97
Figure E.25 – RLC with magnetic coupling configuration	97
Figure E.26 – Impedance seen from Vcc and Gnd	97
Figure E.27 – Complete PDN component	98
Figure E.28 – Set-up for correlation (left), measurement and prediction model (right)	99
Figure E.29 – Set-up used to measure the internal decoupling capacitor	99
Figure F.1 – Equivalent schematic of the complete electronic system	100
Figure F.2 – Impedance prediction and measurements	101
Figure G.1 – IEC 61967-4 test set-up standard	102
Figure G.2 – Comparison between prediction and measurement	102
Figure H.1 – Prediction of ETVddc noise level at PCB level	103
Figure H.2 – Good agreements on the noise envelope	104

Table 1 – Attributes of Lead keyword in the <i>Lead_definitions</i> section	25
Table 2 – Compatibility between the Mode and Type fields for correct CEML annotation.....	26
Table 3 – <i>Subckt</i> definition.....	26
Table 4 – Definition of the <i>Validity</i> section	28
Table 5 – Definition of the Lead keyword for <i>Pdn</i> section	32
Table 6 – Valid data formats and their default units in the <i>Pdn</i> section.....	35
Table 7 – Valid file extensions in the <i>Pdn</i> section	35
Table 8 – Valid fields of the Lead keyword in the <i>Pdn</i> section	36
Table 9 – <i>Netlist</i> definition	39
Table 10 – Differences between the <i>Pdn</i> and <i>Ibc</i> section fields	41
Table 11 – Valid fields of the <i>Lead</i> keyword for IBC definition	41
Table 12 – Definition of the <i>Lead</i> keyword in the <i>Ia</i> section.....	42
Table 13 – <i>Voltage</i> and <i>Current</i> definition	43
Table 14 – Valid file extensions in the <i>Ia</i> section.....	43
Table 15 – Definition of the <i>Pulse</i> keyword in the <i>Voltage</i> or <i>Current</i> section	44
Table 16 – Base units of the Pulse section's fields.....	44
Table 17 – Valid data formats and their default units for the <i>Voltage</i> and <i>Current</i> elements.....	46
Table A.1 – Valid logarithmic units	54
Table D.1 – One-port conversion	77
Table D.2 – Two-port conversion	78
Table E.1 – Typical parameters for CMOS logic technologies	80
Table E.2 – Typical number of logic gates vs. CPU technology	80
Table E.3 – R, L and C parameters for various package types	81
Table E.4 – Measurement configurations and extracted RLC parameters.....	95

IECNORM.COM : Click to view the full PDF of IEC 62433-2:2017

INTERNATIONAL ELECTROTECHNICAL COMMISSION

EMC IC MODELLING –

Part 2: Models of integrated circuits for EMI behavioural simulation – Conducted emissions modelling (ICEM-CE)

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees 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 Publication or any other IEC 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 some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62433-2 has been prepared by subcommittee 47A: Integrated Circuits, of IEC technical committee 47: Semiconductor devices.

This second edition cancels and replaces the first edition published in 2008. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

Incorporation of an XML based exchange format for model representation.

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

FDIS	Report on voting
47A/999/FDIS	47A/1007/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 publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62433 series, published under the general title *EMC IC modelling*, can be found on the IEC website.

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

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

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

EMC IC MODELLING –

Part 2: Models of integrated circuits for EMI behavioural simulation – Conducted emissions modelling (ICEM-CE)

1 Scope

This part of IEC 62433 specifies macro-models for an Integrated Circuit (IC) to simulate conducted electromagnetic emissions on a printed circuit board. The model is commonly called Integrated Circuit Emission Model – Conducted Emission (ICEM-CE).

The ICEM-CE macro-model can also be used for modelling an IC-die, a functional block and an Intellectual Property (IP) block.

The ICEM-CE macro-model can be used to model both digital and analogue ICs.

Basically, conducted emissions have two origins:

- conducted emissions through power supply terminals and ground reference structures;
- conducted emissions through input/output (I/O) terminals.

The ICEM-CE macro-model addresses those two types of origins in a single approach.

This standard defines structures and components of the macro-model for EMI simulation taking into account the IC's internal activities.

This part of IEC 62433 has two main parts:

- the first is the electrical description of ICEM-CE macro-model elements along with the specific requirements for information.
- the second part proposes a universal data exchange format called CEML based on XML. This format allows encoding the ICEM-CE in a more useable and generic form for simulating the conducted emissions.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC TS 62433-1:2011, *EMC IC modelling – Part 1: General modelling framework*

CISPR 17, *Methods of measurement of the suppression characteristics of passive EMC filtering devices*

3 Terms, definitions, abbreviations and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1

external terminal

terminal of an integrated circuit (IC) macro-model which interfaces the model to the external environment of the IC

EXAMPLE Power supply pins and input/output pins.

Note 1 to entry: In this part of IEC 62433, the name of each external terminal starts with "ET".

3.1.2

internal terminal

terminal of an integrated circuit (IC) macro-model's component which interfaces the component to other components of the IC macro-model

Note 1 to entry: In this part of IEC 62433, the name of each internal terminal starts with "IT".

3.1.3

section

XML element placed one level below the root element or within another section and that contains one or more XML elements, but no value

3.1.4

parent

keyword which is one level above another keyword

3.1.5

child

keyword which is one level below another keyword

3.1.6

parser

tool for syntactic analysis of data that is encoded in a specified format

3.1.7

S-parameter

scattering parameter

S_{ij}

element of the S -matrix expressing the transmission and reflection coefficients of a device

Note 1 to entry: The most commonly used, each S -parameter relates the complex electric field strength of a reflected or transmitted wave to that of an incident wave; the subscripts of a typical S -parameter S_{ij} refer to the output and input ports related by the S -parameter, which may vary with frequency.

[SOURCE: CISPR 17:2011, 3.1.13]

3.1.8

CEML

Conducted Emissions Markup Language

data exchange format for ICEM-CE macro-model

Note 1 to entry: This note applies to the French language only.

3.1.9

CEMLBase

Conducted Emissions Markup Language Base

abstract type from which all CEML model components are directly or indirectly derived in the ICEM-CE macro-model definition

3.1.10**PDN**

Passive Distribution Network

component of an IC model that represents the characteristics of propagation path of electromagnetic noises such as power distribution network

[SOURCE: IEC TS 62433-1:2011, 3.4]

3.1.11**IA**

Internal Activity

component of an IC model represented by a current or voltage source, which originates in activity of active devices in an IC or in a portion of the IC

[SOURCE: IEC TS 62433-1:2011, 3.3]

3.1.12**IBC**

Inter-Block Coupling

network of passive elements that presents a coupling effect between circuit blocks within an IC

[SOURCE: IEC TS 62433-1:2011, 3.5]

3.1.13**VNA**

Vector Network Analyzer

network analyzer capable of measuring complex values of the S -parameters

[SOURCE: CISPR 16-1-4:2010, 3.1.21, modified — for VNA with two or more ports]

3.2 Abbreviations

CEM	Conducted Emission Model
XML	eXtensible Markup Language
SPICE	Simulation Program with Integrated Circuit Emphasis

3.3 Conventions

For the sake of clarity, but with some exceptions, the writing conventions of XML have been used in text and tables.

The symbol “ μ ” is used in the text part to define micro = $1e-6$. The symbol “u” is used in the XML parts to define the micro = $1e-6$.

4 Philosophy**4.1 General**

Integrated circuits will have more and more gates on silicon and technical progress will develop faster. To predict the electromagnetic behaviour of equipment, it is required to model the switching of the input and output interface and the internal activities of an IC effectively.

Figure 1 depicts an example of decomposition of an IC to enable conducted emissions analysis. The internal digital activity (culprit) is a source of electromagnetic noise that originates in switching of active devices. The coupling path propagates the emissions to the

IC's external terminals: pins/pads. The coupling path is the power distribution network or I/O lines inside the IC.

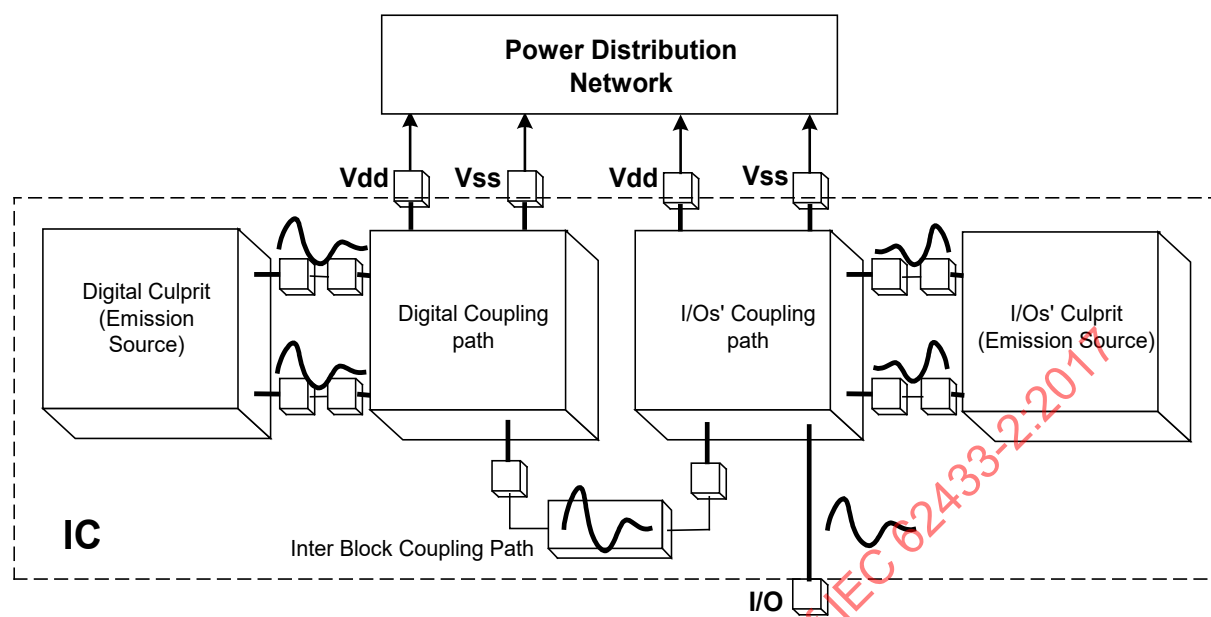


Figure 1 – Decomposition example of a digital IC for conducted emissions analysis

4.2 Conducted emission from core activity (digital culprit)

The current transients are created in the core area on the IC-die. Due to the characteristics of the digital coupling paths, the passive distribution network on printed circuit board (PCB) and the availability of on-chip decoupling, a portion of these current transients will occur at the power supply pins of the IC.

NOTE These off-chip power supply currents can be measured according to the IEC 61967 series [1]¹.

4.3 Conducted emission from I/O activity

I/Os activities may create voltage fluctuations of power and ground levels, and conducted emissions appear at power and ground pins through the I/Os' coupling path. And the output signals at output pins themselves are sources of conducted emissions to the printed circuit boards.

NOTE The measurement set-up is done according to the IEC 61967 series [1].

4.4 Data exchange format

ICEM-CE macro-model data is arranged in a decipherable nested manner using XML format. The objective of this exchange format, called Conducted Emissions Markup Language (CEML), is to create simple and practical universal access to ICEM-CE macro-model. The preliminary definitions for XML representation is given in Annex A. The CEML keywords and their usage rules are detailed in Annex B. An example ICEM-CE macro-model, in CEML format, is given in Annex C.

¹ Numbers in square brackets refer to the Bibliography.

5 ICEM-CE basic components

5.1 General

The basic components are component parts of the IC macro-model or block component or sub-model component. The basic components of the ICEM-CE macro-model are defined in 5.2 and 5.3.

NOTE The block component and the sub-model component are defined in 6.3.1 and 6.4.1 respectively.

5.2 Internal Activity (IA)

5.2.1 General

The Internal Activity (IA) component is the electromagnetic noise source that originates in switching of active devices in the IC or in a portion of the IC. This component is applicable for both analogue and digital circuitry.

The IA is described using an independent current source or an independent voltage source with two internal terminals. Figure 2 illustrates an example of an independent current source.

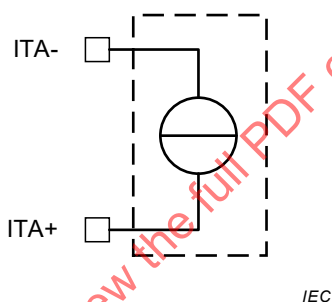


Figure 2 – IA component in the case of a current source

The characteristics of IA component are typically described in the time domain, and the characteristics can also be described in the frequency domain.

The description of an IA component shall contain the following information.

- Name of the IA component
- Names of its internal terminals
- Operational mode or test vector
- Domain (time or frequency)
- Definition of origin of time, and cycle-time for the operational mode (for time domain)
- Definition of origin of phase (for frequency domain)
- Operational conditions and applicable ranges
 - a) Power supply voltage ranges
 - b) Temperature range
 - c) Frequency range
- Characteristics of the IA
 - a) Current or voltage waveform over the whole cycle-time (for time domain)
 - b) Current or voltage amplitude and phase, versus frequency over the whole frequency range (for frequency domain)

5.2.2 Examples of IA

5.2.2.1 Time-domain IA example

Figure 3 shows an example of characteristics of IA in the time domain. The waveform depends on the specific operational mode of function. A simple waveform such as a triangular waveform can be used for the component description.

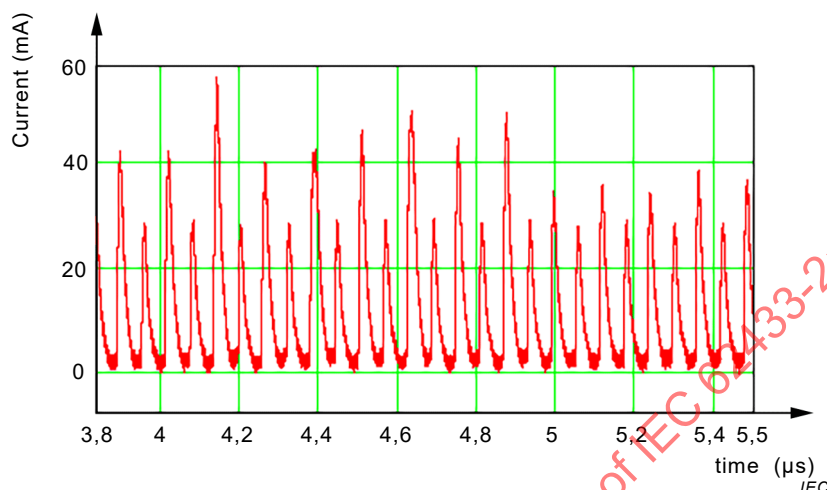


Figure 3 – Example of IA characteristics in the time domain

5.2.2.2 Frequency-domain IA example

Figure 4 shows an example of characteristics of IA in the frequency domain.

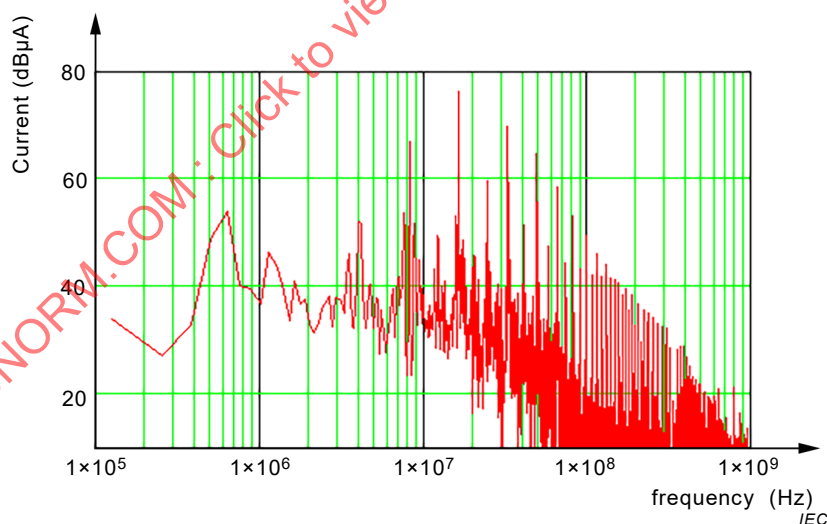


Figure 4 – Example of IA characteristics in the frequency domain

5.3 Passive Distribution Network (PDN)

5.3.1 General

The Passive Distribution Network (PDN) component presents the characteristics of propagation path of electromagnetic noises such as power distribution network (part of the PDN). The PDN can be linear or non-linear.

The PDN consists of passive elements, and is equipped with internal and external terminals.

For cases when the PDN can be assumed to be linear, it can be specified by different network parameters such as:

- $Z(f)$: Impedance, which is the ratio of voltage and current at the disturbance input of the PDN. It represents the electrical schematic of passive input impedance, often consisting of parasitic elements (resistance, inductance and capacitance)
- $Y(f)$: Admittance, which is the inverse of $Z(f)$
- $S(f)$: S -parameter, which is the ratio of the reverse and forward voltage waves at the disturbance input of the PDN. This parameter is typically used to measure the characteristics of Radio Frequency (RF) signal ports

The conversion between the three types of parameters is described in Annex D. The frequency validity range of the PDN is defined by the measurement conditions.

For linear or non-linear PDN, it can be described using a SPICE like netlist [2].

The description of a PDN component shall contain the following information.

- Name of the PDN component
- Names of its internal terminals and external terminals
- Applicable ranges
 - Power supply voltage range
 - Temperature range
 - Applicable load conditions if the PDN is for output
 - Applicable frequency range
- Characteristics of the PDN

5.3.2 Examples of PDN

Figure 5 shows an example of a four-terminal PDN using lumped elements. The ETVdd and ETVss are two external terminals of the PDN. The IT0 and the IT1 are two internal terminals.

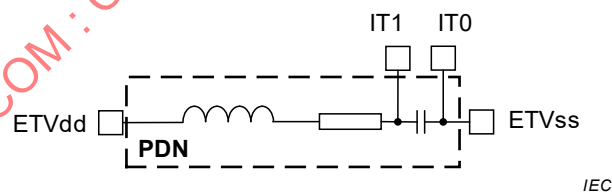
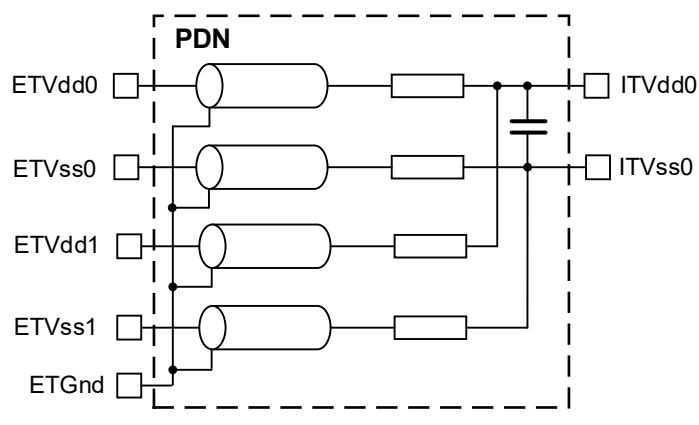


Figure 5 – Example of a four-terminal PDN using lumped elements

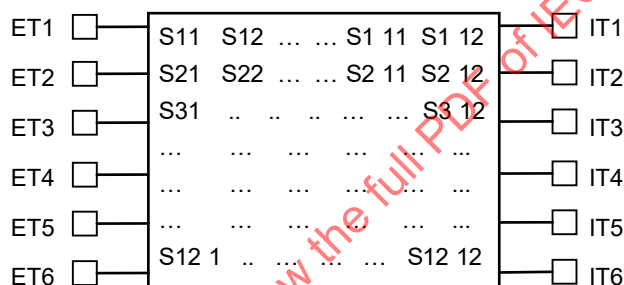
Figure 6 depicts the seven-terminal PDN structure using distributed elements such as transmission lines. The ETVxx are the four external terminals, the ITVxx are two internal terminals and the ETGnd is the common ground of the four transmission lines, connected to the PCB ground.



IEC

Figure 6 – Example of a seven-terminal PDN using distributed elements

Figure 7 shows an example of a twelve-terminal PDN using scattering parameters in a matrix format (black box). The ET1 to ET6 are external terminals. The IT1 to IT6 are internal terminals. A ground plane below the modelled IC is taken as an ideal reference ground for these terminals.



IEC

Figure 7 – Example of a twelve-terminal PDN using matrix representation

6 IC macro-models

6.1 Types of IC macro-models

An IC is modelled as an IC macro-model. Three types of IC macro-models are possible: general model, block-based model and sub-model-based model as defined in 6.2, 6.3 and 6.4 respectively.

The description of an IC macro-model shall contain the following information for model circulation.

- Name of the IC macro-model
- Type of the model, general model or block-based model or sub-model-based model
- Names of components that are included in the IC macro-model
- Names of its external terminals
- Connections of the internal terminals of its components

6.2 General IC macro-model

The general model consists of a single PDN and one or more IAs as shown in Figure 8. The PDN shall include both the whole PDN on the IC die(s) and the whole PDN of the package. An on-chip decoupling capacitor shall also be included in the PDN if it exists.

NOTE This structure is suitable for the model circulation to IC users because of the least disclosure of proprietary information of the IC vendor.

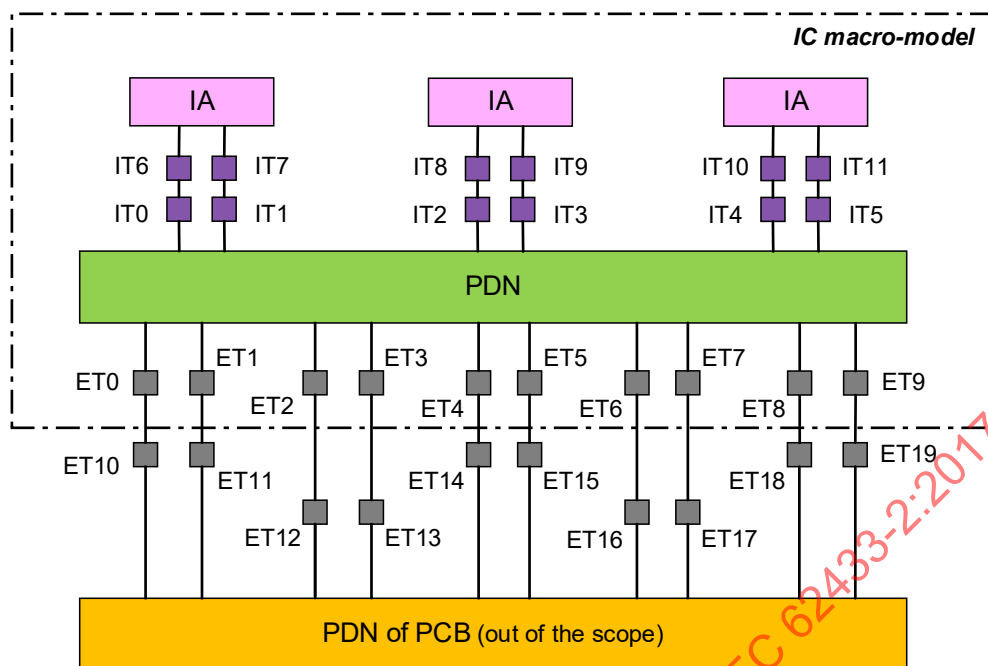


Figure 8 – General IC macro-model

6.3 Block-based IC macro-model

6.3.1 Block component

6.3.1.1 General

The block component represents EMC properties of a specific functional block of an IC such as embedded memory.

The block component consists of a single PDN and one or more IAs. The PDN includes PDN of the specific functional block, a portion of global power/ground network and a portion of package PDN, which are directly involved in the block's functionality. The on-chip decoupling capacitor is a part of the PDN. The component is equipped with external terminals and internal terminals.

The description of a block component shall include the following information.

- Name of the block component
- Names of the basic components that make up the block component
- Connections of the internal terminals of its basic components

6.3.1.2 Example of block-component with a single PDN and IA

Figure 9 shows an example of block component. The block consists of an IA and a PDN. The internal terminals of the IA are connected to the internal terminals of the PDN.

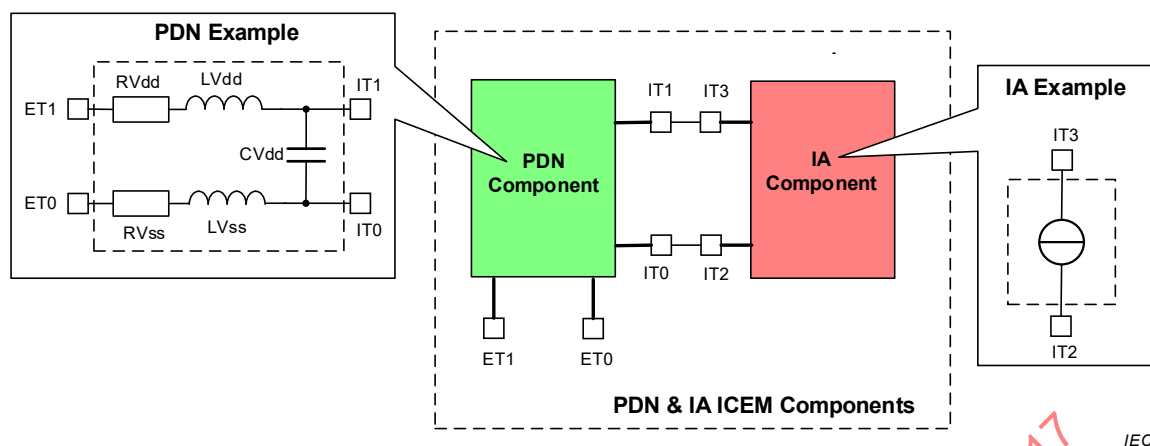


Figure 9 – Example of block component with a single IA

6.3.1.3 Example of block-component with multiple IAs

Figure 10 depicts a three I/Os model. The I/O PDN component describes how I/Os are powered and the I/O PDNA describes noise transfer characteristics among terminals. The I/O IA components describe the current activity. They are built up using two IA components; one to specify the high state behaviour and the other one to specify the low state. Figure 10 shows the two types of I/O PDN components of the complete I/O model.

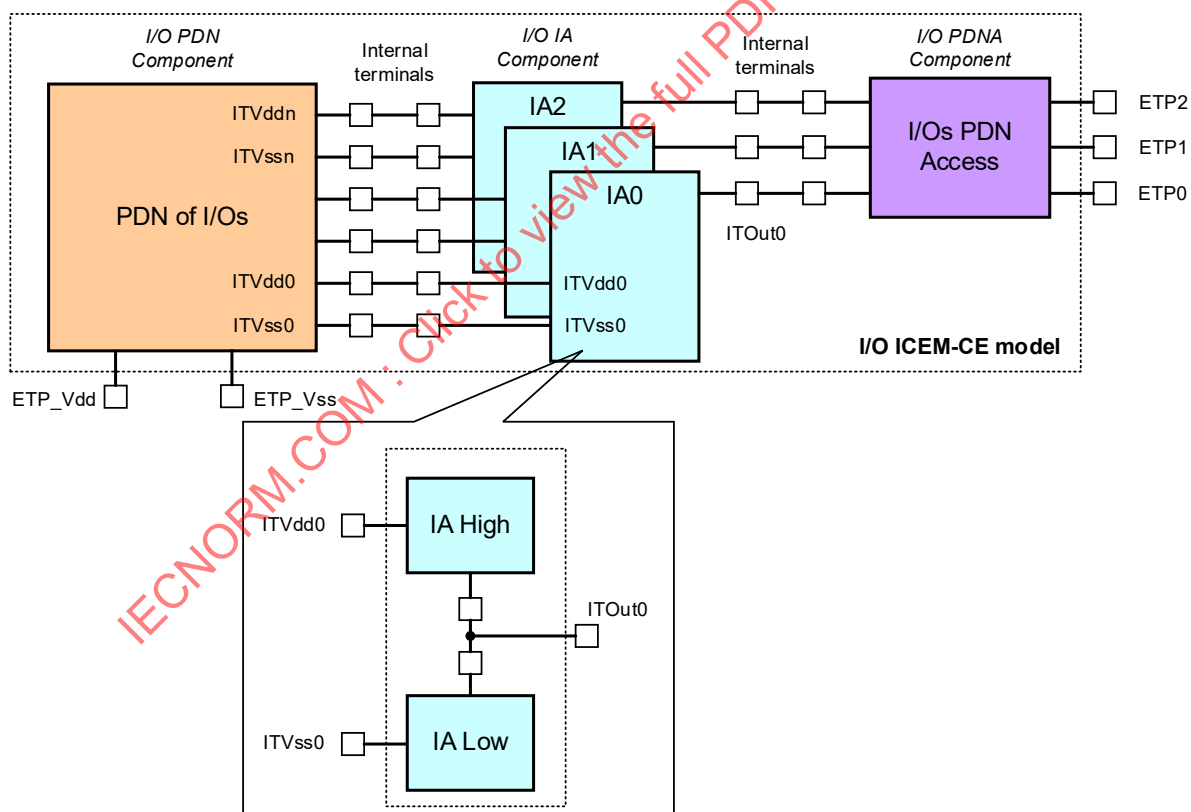


Figure 10 – Example of block components for I/Os

6.3.2 Inter-Block Coupling component (IBC)

6.3.2.1 General

The Inter-Block Coupling (IBC) is a network of passive elements that presents a coupling effect between blocks. The IBC is equipped with two or more internal terminals. Such blocks can be used for modelling the coupling phenomena between different IC ground terminals,

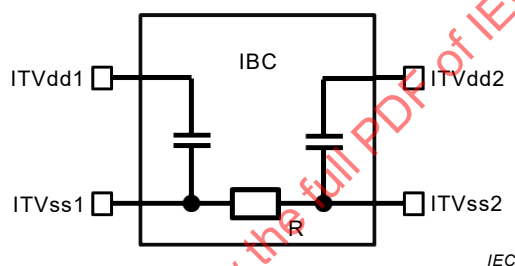
substrate losses, mutual inductances at die-level, insulation between internal ground and power terminals.

The description of an IBC component shall contain the following information.

- a) Name of the IBC
- b) Names of its internal terminals
- c) Applicable ranges
 - 1) Power supply voltage ranges
 - 2) Temperature range
 - 3) Applicable frequency range
- d) Characteristics of the IBC

6.3.2.2 Example of an IBC and its relationship with the PDN

An example of an IBC network is shown in Figure 11. In this example, both capacitors model the dielectric properties and the resistor models the resistive properties of the substrate. Other properties can be modelled using more complex IBC networks.



NOTE ITVdd1, ITVss1, ITVdd2 and ITVss2 are internal terminals.

Figure 11 – Example of IBC with four internal terminals

The relationship between different PDN blocks and the IBC is shown in Figure 12.

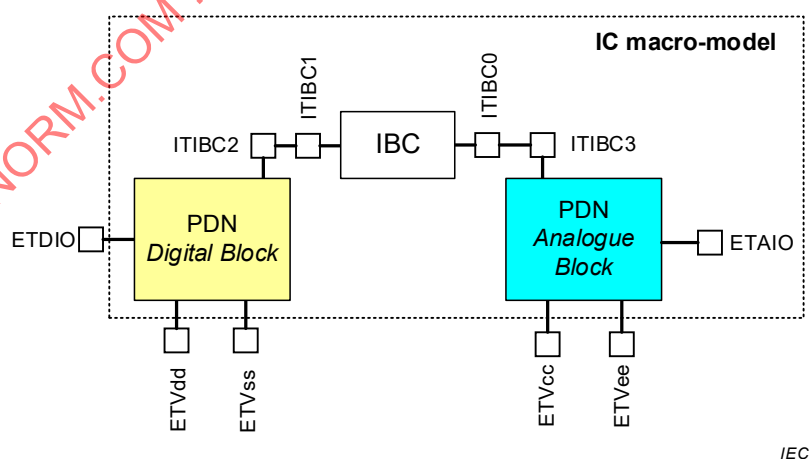


Figure 12 – Relationship between blocks and IBC

6.3.3 Block-based IC macro-model structure

An example of a block-based structure is shown in Figure 13. The model consists of block components and IBC components. The PDN of global wiring on die and the PDN on package are incorporated into the PDNs of blocks.

NOTE 1 This structure is suitable for modelling from measurements.

NOTE 2 By combining PDNs of block and IBCs, this structure can be converted into general structure.

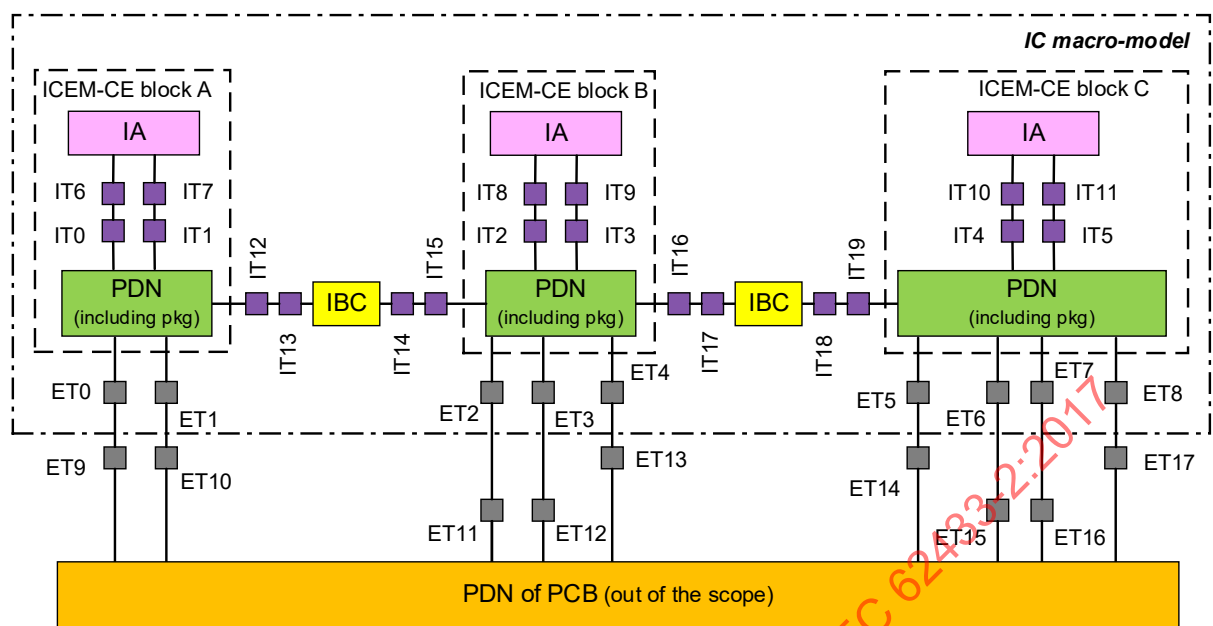


Figure 13 – Block-based IC macro-model

Figure 14 depicts an example of block-based IC macro-model. Each block has two external terminals and three internal terminals. Three IBC blocks interconnect the internal terminals, IT2, IT3, IT4, IT5 and IT6. In this example, IBCs are used to model the substrate coupling caused by the sheet resistance between the three internal ground terminals.

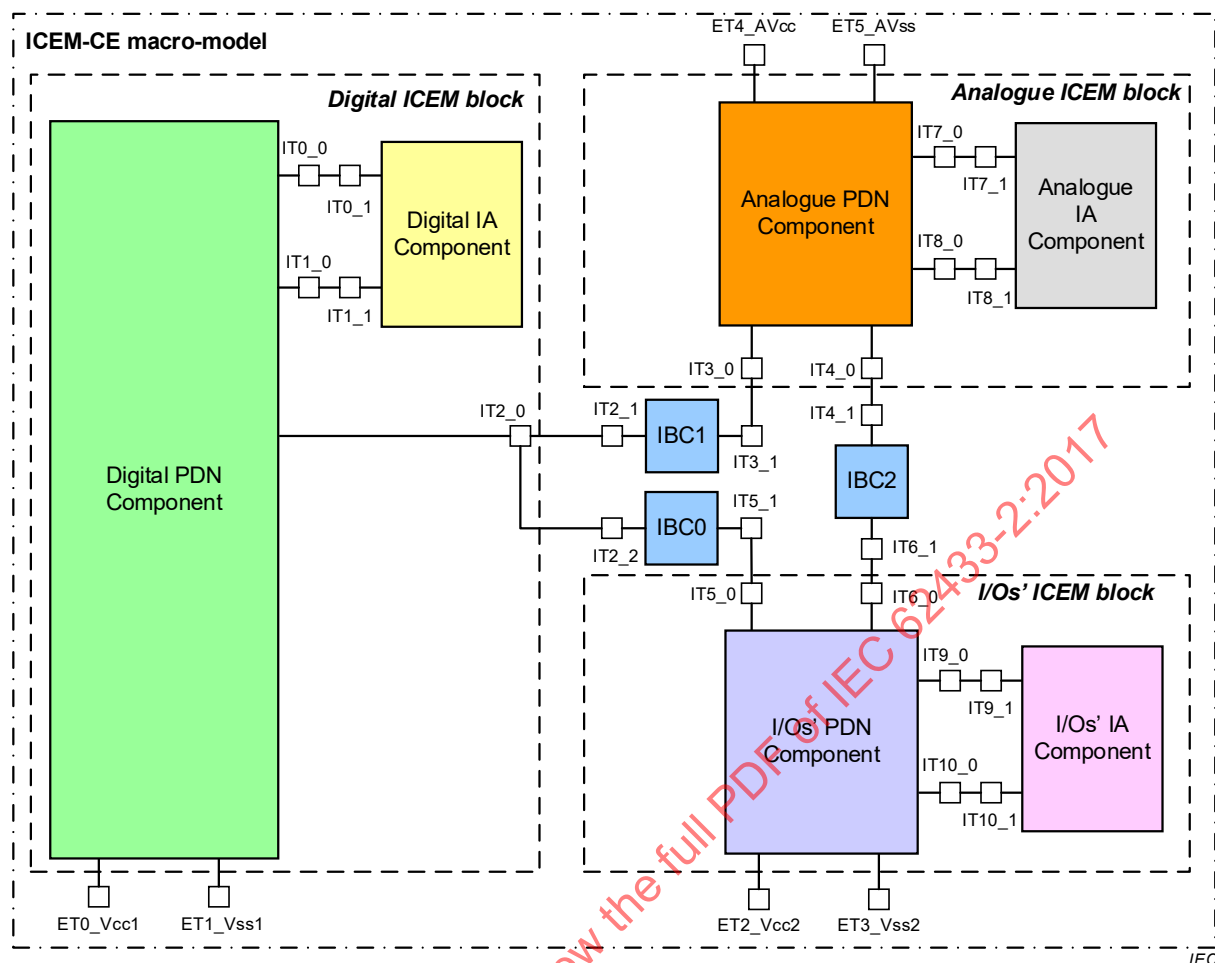


Figure 14 – Example of block-based IC macro-model

6.4 Sub-model-based IC macro-model

6.4.1 Sub-model component

The sub-model in Figure 16 represents the electromagnetic behaviour of specific functional circuits of IC. Some examples include an IP block, functional circuits such as embedded memory, Central Processing Unit (CPU) core, etc. These blocks can be modelled using the sub-model; it can be repeatedly used in the IC and/or other ICs.

The sub-model consists of a single PDN and one or more IAs. This PDN is a PDN of the specific functional circuits. The sub-model is equipped with internal terminals but does not have any external terminals. An example of a simple sub-model is shown in Figure 15.

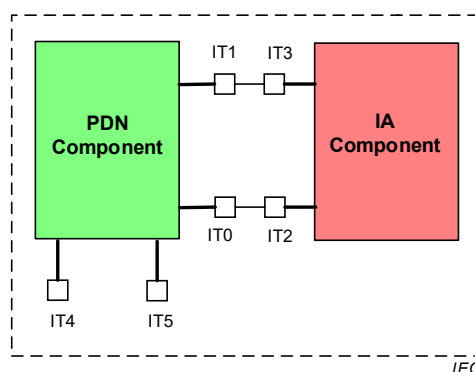


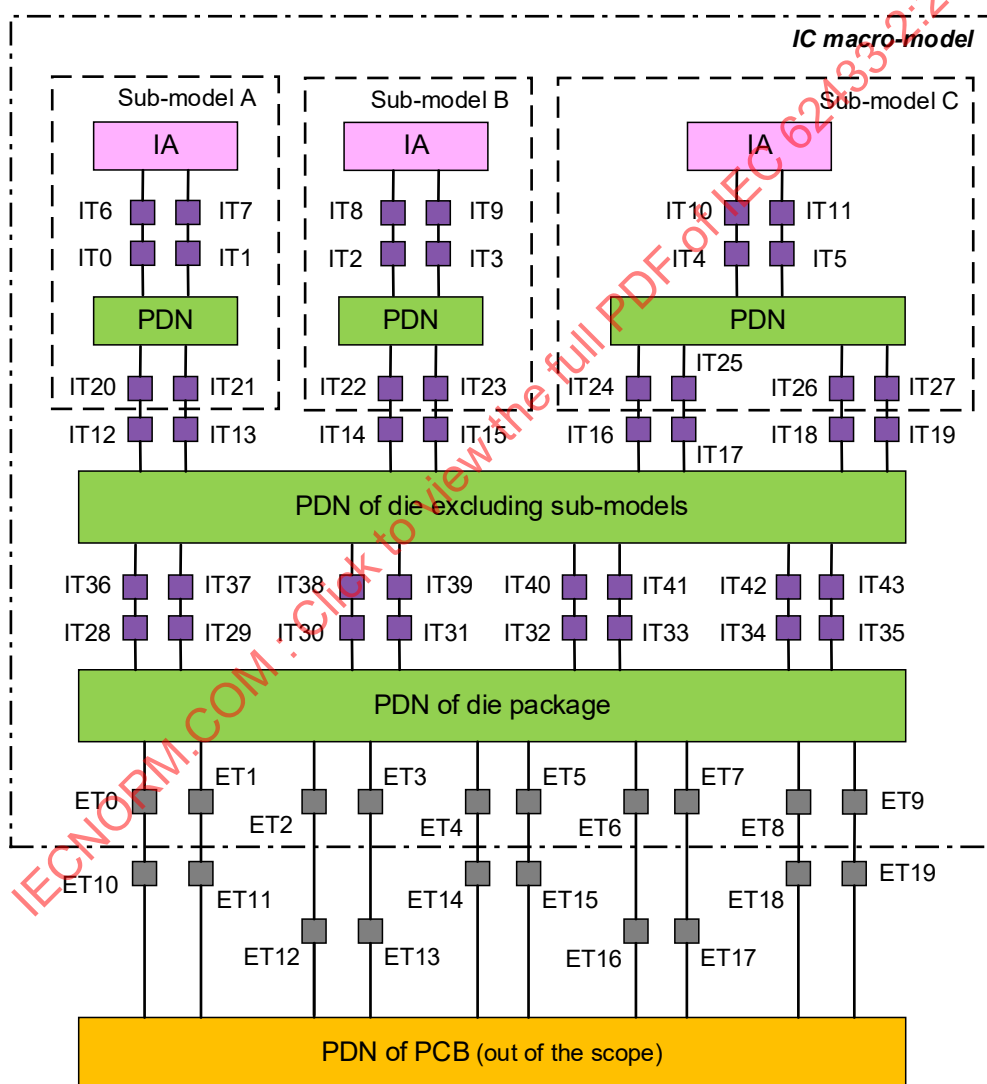
Figure 15 – Example of simple sub-model

The sub-model description shall contain the following information.

- Name of the sub-model
- Names of its internal terminals
- Names of the basic components that are included in the sub-model
- Connections of the internal terminals of basic components

6.4.2 Sub-model-based IC macro-model structure

The sub-model-based structure is shown in Figure 16. It consists of one or more sub-models, a PDN of the die, and a PDN of the package. The PDN of the die includes the global power/ground network of the IC die and the on-chip decoupling capacitor if it exists, but does not include the PDNs that belong to sub-models. Some IAs such as IAs for standard cell circuits can be directly connected to the PDN of the die (not shown in the figure).



IEC

Figure 16 – Sub-model-based IC macro-model

NOTE 1 This structure is suitable for modelling using IC design information.

NOTE 2 By combining PDNs of the whole IC macro-model, a sub-model-based IC macro-model can be converted into a general IC macro-model.

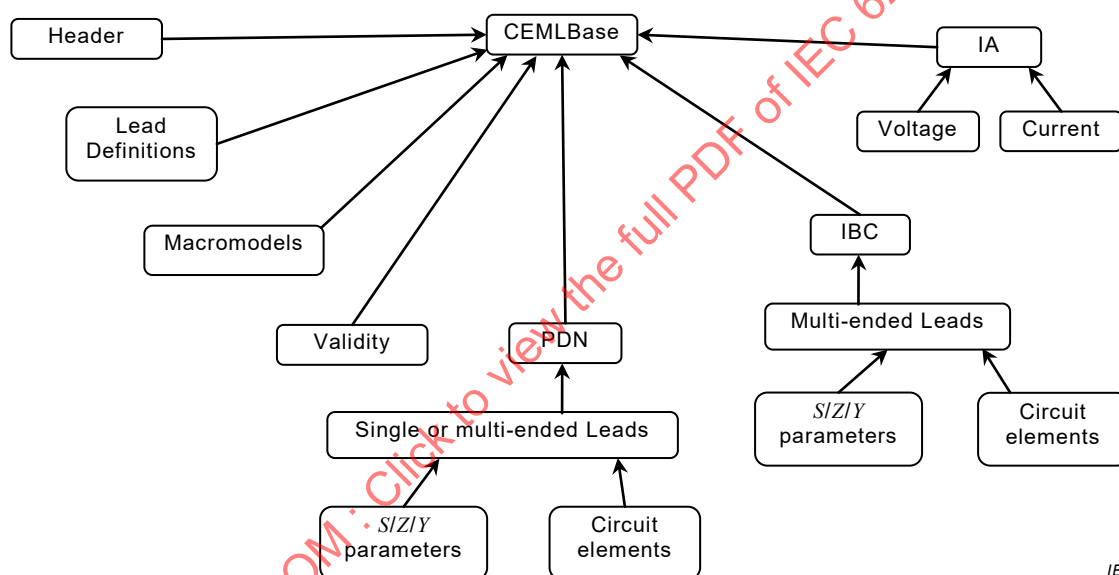
7 CEML format

7.1 General

Data of the ICEM-CE macro-model is arranged in xml format, henceforth called as Conducted Emissions Markup Language (CEML). The model data is separated into seven parts:

- Header containing general information
- Description of IC's Lead or Node definitions
- Description of macro-models for use within the PDN and IBC data
- Description of the model validity conditions
- Description of the PDN
- Description of the IBC
- Description of the IA

The inheritance hierarchy is depicted in Figure 17.



IEC

Figure 17 – CEML inheritance hierarchy

The top level of a CEML model definition simply consists of these components:

```

beginning of model definition
  model header definition
  DUT lead definitions
  SPICE macro-models
  model validity conditions
  model PDN
  model IBC
  model IA
end of model definition

```

This exchange format uses eXtensible Markup Language (XML) 1.0 (Fourth Edition) to structure the information. XML is derived from the Standard Generalized Markup Language (SGML) [3].

The rules below ensure that the file can be correctly parsed by a Conducted Emissions Model (CEM) parser. An example CEML file is shown in Annex C.

7.2 CEML structure

Considering all the rules defined in Annex A, and being consistent with model description requirements in IEC TS 62433-1, a typical ICEM-CE macro-model file is represented as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<CEmodel>
<!-- HEADER -->
  <Header>
    ...
  </Header>
<!-- DUT LEAD DEFINITIONS-->
  <Lead_definitions>
    ...
  </Lead_definitions>
<!-- SPICE MACROMODEL DEFINITIONS -->
  <Macromodels>
    ...
  </Macromodels>
<!-- MODEL VALIDITY CONDITIONS -->
  <Validity>
    ...
  </Validity>
<!-- MODEL PDN DATA -->
  <Pdn>
    ...
  </Pdn>
<!-- MODEL IBC DATA -->
  <Ibc>
    ...
  </Ibc>
<!-- MODEL IA DATA -->
  <Ia>
    ...
  </Ia>
</CEmodel>
```

The *Header*, *Lead_definitions*, *Validity*, *Pdn*, and *Ia* sections are minimum and mandatory information of the ICEM-CE macro-model. The *Macromodels* and *Ibc* sections are optional. Each section of the file is presented below.

7.3 Global keywords

Documentation, *Notes*, *Unit* sections are global keywords and can be placed anywhere in the file except within an element containing a value. See B.4 for more information on usage.

7.4 Header section

Readers may wonder about the motivations for an independent Header section. A simpler approach to creating the header information would be to place them all directly at the top level under <CEmodel> ... </CEmodel>. It is instead chosen to group them within the XML element <Header>, because such an approach would help to organize the components in a better format and make visual reading of model definitions easier. It is thus proposed to define header information within the *Header* tag. The minimum details are the model version number, filename and the file version number. Other header contents are freely dimensioned as per IEC TS 62433-1, giving information such as:

- DUT reference
- Authors' name

- Date
- Measurement method
- Copyright
- Disclaimer
- Documentation

EXAMPLE An example *Header* section may be defined as shown

```
<Header>
  <Cem_ver>1.0</Cem_ver>
  <Filename>ExampleICEMCE_file.CEML</Filename>
  <File_ver>1.0</File_ver>
  <Author>Valeo 1</Author>
  <Dut>MCYYY/Dut>
  <Date>April 3, 2013</Date>
  <Meas_method> PDN using Netlist, IA in time-domain </Meas_method>
</Header>
```

A detailed list of valid keywords under the *Header* section is available in Annex B.

7.5 Lead definitions

This section describes the various leads or terminals of the IC under test. Each lead in the *Lead_definitions* section is made using the *Lead* tag, whose definition is shown in Table 1. Several *Lead* structures are listed one after another to form the *Lead_definitions* structure.

Table 1 – Attributes of Lead keyword in the *Lead_definitions* section

Lead
Id: pin identity as a valid string (required) Name: Name of the pin as designated in the datasheet (optional). Default = "None" Mode: Mode in which the pin is used for ICEM-CE ("GND") (optional). Default="None" Type: Type in which the pin is used for ICEM-CE ("internal", "external") (optional). Default="external"

Every *Lead* structure in the *Lead_definitions* section has one required field, *Id*, representing the identity of the lead. The lead may be additionally defined with its *Name*, *Mode* and *Type* fields. If the *Name* field is undefined or absent, it defaults to "None". The *Mode* field is considered as "None" under the following conditions:

- If *Mode* for a particular pin is absent
- If explicitly set as "None" since the respective pin is not GND (reference return path).

The *Type* field is optional and can be "external" or "internal". See 3.1.1 and 3.1.2 for the definition of these terms.

Leads with *Type*="external" are used to interface the model to the external environment of the IC, such as power supply pins and I/O pins. The "internal" terminals are used for interfacing different blocks of the block-based or sub-model-based ICEM-CE macro-model (as explained in 6.3 and 6.4). The "internal" terminals shall be defined with *Mode*="None". These leads are dedicated for interfacing the different blocks of the block-based or sub-model-based ICEM-CE components.

Leads that are defined with *Mode*="GND" are considered as "external" terminals.

NOTE Internal ground terminals can be specified as any other internal node. i.e., *Type*="internal" with *Mode*="None".

Table 2 lists the compatibility between the *Mode* and *Type* fields of the *Lead* structure for correct CEML annotation by the CEM parser.

Table 2 – Compatibility between the Mode and Type fields for correct CEML annotation

Mode	Type	
	external	internal
GND	Yes	No
None	Yes	Yes

An example of the *Lead_definitions* section, representing Figure 5, is shown below.

```
<Lead_definitions>
  <Lead Id="1" Name="ETVdd" Type="external"/>
  <Lead Id="2" Name="IT1" Type="internal" Mode="None"/>
  <Lead Id="3" Name="IT0" Type="internal"/>
  <Lead Id="4" Name="ETVss" Mode="GND"/>
</Lead_definitions>
```

7.6 SPICE macro-models

This section described the various SPICE (Simulation Program with Integrated Circuit Emphasis) macro-models in netlist format [2].

These sub-circuits are references within the *Pdn* tag under the Netlist section as explained in 7.8.3.4 and 7.8.3.5. Each SPICE macro-model in the *Macromodels* section is defined using the *Subckt* tag, whose definition is shown in Table 3. This section is optional for the ICEM-CE macro-model.

Table 3 – Subckt definition

<i>Subckt</i>
Name: Name of the macro-model (required)
Nodes: External Nodes connecting to the main circuit (required)
Kind: SPICE netlist format (optional) default: "SPICE3"
Data_files: Macro-model defined in an external file (optional)

The *Subckt* keyword has two required fields: *Name* and *Nodes*. The *Name* field consists of letters and numbers defined in A.2.5.4. Since these definitions share the same namespace within the CEML format, every sub-circuit shall carry a unique name

The *Nodes* field defines the external nodes that connect outside the sub-circuit, i.e., the nodes through which the sub-circuit element connects to the main circuit. The different external nodes are defined in sequence separated by a comma (",") and these nodes are strictly local to the *Subckt* definition. These *Nodes* can be identified with either numbers and/or characters [2].

The optional attribute *Kind* tells the CEM parser that the defined sub-circuit follows a specific syntax. CEML version 1 supports industry-standard SPICE-like netlist syntaxes [2]. The *Kind* field defaults to generic "SPICE3" if absent.

An example *Subckt* element is shown in Example 1.

EXAMPLE 1 An example of a SPICE-like *Subckt* element.

```
<Subckt Name="PDN_pin1" Nodes="Node1,Node2" Kind="SPICE3">
  L1 Node1 int1 2.2e-9
  R1 int1 int2 10e-3
  C1 int2 int3 800e-12
  L1 int3 Node2 2.2e-9
</Subckt>
```

For compatibility reasons between several SPICE kinds, "0" is not allowed as an external node. The different sub-circuit elements are listed one after another to form the *Macromodels* section (see Example 2).

EXAMPLE 2 An example of a *Macromodels* section with several *Subckt* elements.

```
<Macromodels>
  </Subckt Name="PDN_example1" Nodes="Node1,Node2">
  ...
</Subckt>
  </Subckt Name="PDN_example2" Nodes="Node1,Node2,Node3">
  ...
</Subckt>
  ...
</Macromodels>
```

Once defined, the various sub-circuits are usable anywhere within the *Pdn* and *Ibc* tags using an identifier starting with the character "X". The call shall be made under the *Netlist* section and is referenced using the *Name* field (see 7.8.3.4 and 7.8.3.5). The number of nodes on the call line shall match the number listed in the *Nodes* attribute line of the specific *Subckt*. For example the above defined "PDN_pin1" (in Example 1) can be called:

```
<Pdn>
  ...
  <Lead Id="1" Type="Ckt">
    ...
    <Netlist>
      Xpin1 1 4 PDN_pin1
    </Netlist>
  </Lead>
  ...
</Pdn>
```

If one or more sub-circuit models is (are) defined in an external library file, then the file(s) is (are) referenced using the *Data_files* tag as shown in Example 3.

EXAMPLE 3 An example of a *Macromodels* section with several *Subckt* elements defined in external data files.

```
<Macromodels>
  <Subckt Kind="SPICE3">
    <Data_files>
      subckt_example1.lib
      subckt_example2.lib
      ...
    </Data_files>
  </Subckt>
</Macromodels>
```

A typical sub-circuit definition is shown in Figure 18.

```

*Netlist file created on Fri 27 Nov 2012
*Time: 12:01:27
*PDN macro-model model of PIN1 with respect to GND1
.Subckt Pin1_PDN N1 N2
C1 N1 int1 20e-9
L1 int1 int2 9e-9
R1 int2 N2 230e-3
.ends

```

IEC

Figure 18 – Example of a netlist file defining a sub-circuit

While using external data files containing the SPICE model libraries, to avoid ambiguity, the CEM parser parses only the data statements defined within the SPICE keywords ".SUBCKT" and ".ENDS". These keywords are not case sensitive [2]. If multiple ".SUBCKT" sections are found, then they are parsed as independent sub-circuit elements.

7.7 Validity section

7.7.1 General

The *Validity* section shall be used to represent the conditions in which the ICEM-CE macro-model is defined as per IEC TS 62433-1. This section is strictly informative to the user and independent of all other sections in the CEML file.

Table 4 lists the various recognized keywords in the *Validity* section.

Table 4 – Definition of the *Validity* section

<i>Validity</i>
Power_supply: Power supply range as a string (required)
Temperature_range: Temperature range in which the model is extracted with units (required)
Frequency_range: Frequency range as a string with units (required for frequency-domain model data)
Phase_origin: Origin of phase for frequency-domain IA (required for frequency-domain model data)
Time_range: Time range as a string with units (required for time-domain model data)
Time_origin: Origin of time for time-domain IA (required for time-domain model data)

This definition is not exhaustive and is open for progress and improvement.

Documentation pertaining to the IC such as datasheets, test-reports and ICEM-CE extraction report shall be specified with the global keyword *Documentation* in the form of a valid string that specifies the path to the corresponding documentation. Many file paths' can be listed one below another. See B.4 for more information on the usage of the *Documentation* keyword.

Any other specific details pertaining to the complete ICEM-CE macro-model definition, that may be needed to understand the model, shall be defined with the *Notes* section in the form of a valid string. See B.4 for more information on the usage of the *Notes* keyword.

This section is mandatory and is defined directly under the *CEmodel* root element as shown in Example 1 and Example 2.

EXAMPLE 1 An example of the *Validity* section of a frequency-domain ICEM-CE macro-model is shown below.

```

<CEmodel>
...
<Validity>

```

```

    <Power_supply>5V</Power_supply>
    <Temperature_range>[20Celsius-40Celsius]</Temperature_range>
    <Frequency_range>[1MHz - 1GHz]</Frequency_range>
    <Phase_origin>10degrees</Phase_origin>
    <Notes>Only PLL is activated</Notes>
    <Documentation>
        test_report.pdf
        ICdatasheet.pdf
    </Documentation>
</Validity>
...
</CEmodel>

```

EXAMPLE 2 An example of the *Validity* section of a time-domain ICEM-CE macro-model is shown below:

```

<CEmodel>
...
<Validity>
    <Power_supply>5V</Power_supply>
    <Temperature_range>[20Celsius-40Celsius]</Temperature_range>
    <Time_range>from 1us to 101us</Time_range>
    <Time_origin>1us</Time_origin>
    <Notes>Only PLL is activated</Notes>
    <Documentation>
        test_report.pdf
        ICdatasheet.pdf
    </Documentation>
</Validity>
...
</CEmodel>

```

7.7.2 Attribute definitions

7.7.2.1 Power_supply

The *Power_supply* attribute is used to define the supply conditions for which ICEM-CE is valid. This attribute informs the user that the ICEM-CE macro-model is extracted in the specified supply range. This is a required field.

There is no particular format for defining the value for this attribute; it shall be easily comprehensible for proper model usage. The value shall be a data string containing valid text string and/or numerical values with units. See A.2.5.5 for valid units.

A few examples are shown in Examples 1 to 3.

EXAMPLE 1 The following syntax specifies that the model is defined for a supply voltage of 5 V.

```
<Power_supply>5V</Power_supply>
```

EXAMPLE 2 The following syntax specifies that the digital blocks of the model data are defined with a supply voltage of 5 V and the analogue blocks are defined for a supply voltage of 12 V.

```
<Power_supply>5V for digital blocks, 12V for analogue blocks</Power_supply>
```

EXAMPLE 3 The following syntaxes specify that the model is defined for supply voltages between 2,5 V and 18 V.

```
<Power_supply>[2.5V-18V]</Power_supply>
```

or

```
<Power_supply>between 2.5V and 18V</Power_supply>
```

7.7.2.2 Temperature_range

The *Temperature_range* attribute is used to define the temperature range in which ICEM-CE is extracted. This is a required field.

There is no particular format for defining the value for this attribute; it shall be easily comprehensible for proper model usage. The value shall be a data string containing valid text string and/or numerical values with units. See A.2.5.5 for valid units.

Two examples are shown in Example 1 and Example 2.

EXAMPLE 1 The following syntaxes specify that the model is valid in the temperature range between 20 °C and 40 °C.

```
<Temperature_range>[20Celsius-40Celsius]</Temperature_range>
or
<Temperature_range>from 20Celsius to 40Celsius</Temperature_range>
```

EXAMPLE 2 The following syntax specifies that model is defined only at 298,15 K.

```
<Temperature_range>298.15K</Temperature_range>
```

The temperature range of validity of the ICEM-CE macro-model shall be the common temperature range of the PDN and IA components.

7.7.2.3 Frequency_range

The *Frequency_range* attribute is used define the frequency range in which ICEM-CE is valid. This attribute informs the user that the ICEM-CE macro-model is extracted in the frequency domain, is specified in the given frequency range and is usable in the same range. This is a required field for frequency-domain ICEM-CE macro-model.

There is no particular format for defining the value for this attribute; it shall be easily comprehensible for proper model usage. The value shall be a data string containing valid text string and/or numerical values with units (see Example 1).

EXAMPLE 1 The following syntaxes specify that the model is valid in the frequency range from 1 MHz to 1 GHz.

```
<Frequency_range>[1MHz-1GHz]</Frequency_range>
or
<Frequency_range>from 1MHz to 1GHz</Frequency_range>
```

The frequency range of validity of the ICEM-CE macro-model shall be the common frequency range of the PDN and IA components.

7.7.2.4 Phase_origin

The *Phase_origin* attribute is used define the origin of phase in which ICEM-CE is valid. This information is specific to the IA sub-model. This is a required field for frequency-domain ICEM-CE macro-model.

There is no particular format for defining the value for this attribute; it shall be easily comprehensible for proper model usage. The value of this attribute is a data string containing valid text string and/or numerical values with units (see Examples 1 and 2).

EXAMPLE 1 The following syntax specifies that the phase origin of the model data (IA) is 10 degrees.

```
<Phase_origin>10degrees</Phase_origin>
```

EXAMPLE 2 The following syntax specifies that the phase origin of the IA data of the digital block is 5 degrees and that of the analogue block is 22 degrees.

```
<Phase_origin>5degrees for digital block, 22 degrees for analogue block</Phase_origin>
```

7.7.2.5 Time_range

The *Time_range* attribute is used to define the time range in which ICEM-CE is valid. This attribute informs the user that the ICEM-CE macro-model is extracted in the time-domain, is specified in the given time range and is usable in the same range. This is a required field for time-domain ICEM-CE macro-model.

There is no particular format for defining the value for this attribute; it shall be easily comprehensible for proper model usage. The value of this attribute is a data string containing valid text string and/or numerical values with units (see Example 1).

EXAMPLE 1 The following syntax specifies that the model is valid in the time range from 100 ps to 1 ms.

```
<Time_range>[100ps-1ms]</Time_range>
or
<Time_range>from 100ps to 1ms</Time_range>
```

The time range of validity of the ICEM-CE macro-model shall be the common time range of the PDN and IA components.

7.7.2.6 Time_origin

The *Time_origin* attribute is used to define the origin of time and cycle-time for the operational mode in which ICEM-CE is valid. This information is specific to the IA sub-model. This is a required field for time-domain ICEM-CE macro-model.

There is no particular format for defining the value for this attribute; it shall be easily comprehensible for proper model usage. The value of this attribute is a data string containing valid text string and/or numerical values with units (see Example 1).

EXAMPLE 1 The following syntax specifies that the time origin of the IA data is 100 ns.

```
<Time_origin>100ns</Time_origin>
```

EXAMPLE 2 The following syntax specifies that the time origin of the IA data of the digital block is 100 ps and that of the analogue block is 1 μ s.

```
<Time_origin>100ps for digital block, 1us for analogue block</Time_origin>
```

7.8 PDN

7.8.1 General

The *Pdn* section of the ICEM-CE macro-model contains the PDN data that describes the model. The data shall be defined within the *Pdn* keyword as follows:

```
<Pdn>
    PDN data
</Pdn>
```

A PDN data is defined for a particular IC and thus the definition shall be done within a *Lead* tag. See Figure 17 for the structural hierarchy. Table 5 lists the various recognized fields of the *Lead* keyword:

Table 5 – Definition of the Lead keyword for *Pdn* section

<i>Lead</i>
Id : pin identity as a valid string (required)
Ground_id : return pin identity as a valid string (required if Type="S", "Z", "Y", else optional)
Blockname : PDN block name as a valid string (optional)
Type : PDN source parameter ("S", "Z", "Y", "Ckt")
Param_order : Order in which PDN parameters are defined
Format : Data format ("RI", "MA", "DB")
Meas_type : Method implemented for performing PDN measurements
Reference_impedance : Reference impedance used in performing PDN measurements
Use : Parameter that is to be specifically used
Netlist : PDN definition using standard netlist format (required if Type="Ckt")
Unit_freq : Unit definition of the frequency terms
Unit_param : Unit of the PDN parameters
Power_level : Measurement power level during PDN extraction
Data_files : PDN source parameter defined in an external file (required if not List)
List : PDN parameter list (required if not Data_files)

The PDN data can be defined as a single or multiple PDN blocks (interconnected or disconnected). As explained in 5.3.1, the PDN (or PDN blocks) can be represented using single or multi-port *S*-, *Y*- or *Z*- parameters (network parameters) or using SPICE type netlists [2]. In both the cases, data can be defined inline or using external data files.

The frequency range or time range of the PDN information shall be specified in the *Validity* section under the *Frequency_range* tag or the *Time_range* tag respectively.

NOTE The frequency range (or time range) of validity of the ICEM-CE macro-model is the common frequency range (or time range) of the PDN and IA components.

As per IEC TS 62433-1, any other relevant information required for correct understanding (and usage) of the PDN shall be optionally defined within the *Notes* and *Documentation* tags. Details such as IC's operating mode, decoupling capacitors on supply lines, power-supply ranges, activated functions, load conditions, grounding details, datasheets and test-reports can be defined. See B.4 for more information on the usage of the *Documentation* and *Notes* keywords.

7.8.2 Attribute definitions

7.8.2.1 Id

The lead identities (Ids) used in this PDN definition should have been previously defined in the *Lead_definitions* tag as described in 7.5. Both "internal" and "external" leads can be defined.

This is a required field.

7.8.2.2 Ground_id

The return lead identity (Ground_id) used in this PDN definition should have been previously defined in the *Lead_definitions* tag as described in 7.5. This lead defines the return signal path for defining the PDN.

This attribute is defined only if network parameters are used for defining the PDN (Type="S" or "Z" or "Y"). *Ground_id* attribute is optional if Type="Ckt", that is, if a netlist is used for representing the PDN. See 7.8.2.4 for more information on the *Type* attribute.

When used, only one unique *Ground_id* is permitted per *Lead* definition. The ground terminal shall be the ideal reference ground used for obtaining the network parameters as stated in 5.3.2.

This is an optional field and is required if PDN is represented using network parameters.

7.8.2.3 Blockname

The *Blockname* field is used to define the name of the PDN for block-based or sub-model-based ICEM-CE modelling. If both the PDN and the IA share the same name, then they are regrouped together by the CEM parser as a sub-model.

This field is optional.

7.8.2.4 Type

The *Type* attribute is used to represent the type of the PDN data. Valid types are:

- "S": S-parameter data
- "Z": Z-parameter data. These parameters are not normalized to the reference impedance.
- "Y": Y-parameter data. These parameters are not normalized to the reference impedance.
- "Ckt": Circuit description using netlists

This field is optional. When absent, the default value is "S".

7.8.2.5 Param_order

The *Param_order* attribute tells the CEM parser how the PDN data is represented. The following strings are dedicated for specifying the parameter order:

- "Freq" and "Frequency": Frequency quantity used for parameters' definition
- "Time": Time quantity used for parameters' definition
- "Sij": S-parameters, i and j are integers representing measurement ports/pins (example: "S11", "S21", "S31", etc.)
- "Zij": Z-parameters, i and j are integers representing measurement ports/pins (example: "Z11", "Z21", "Z31", etc.)
- "Yij": Y-parameters, i and j are integers representing measurement ports/pins (example: "Y11", "Y21", "Y31", etc.)
- All: All defined terms with respect to the number of lead Id's are considered.

The different terms of *Param_order* shall be separated by a comma character (",").

This field is optional. It is not defined if the PDN is represented as a Circuit model as discussed in 7.8.3.4 and 7.8.3.5. If absent, the default value is "Freq,All".

7.8.2.6 Format

The *Format* attribute decides the data format. It is not defined if the PDN is represented as a Circuit model as discussed in 7.8.3.4 and 7.8.3.5. Valid data formats are:

- "RI": real/imaginary format (frequency-domain PDN)
- "DB": Magnitude in decibel scale with phase angle in degrees (frequency-domain PDN)

- "MA": Magnitude in linear scale with phase angle in degrees (frequency-domain PDN)
- "DBMAG": magnitude in decibel scale without phase data (time-domain PDN)
- "MAG": Magnitude data in linear scale without phase data (time-domain PDN)

This field is optional. If absent, the default value is "RI".

7.8.2.7 Meas_type

The *Meas_type* attribute is used by the CEM parser in order to compute the impedance of the specific lead *Ids* when *S*-parameters (Type= "S") are used. Depending on the type of the PDN (single-ended, differential or multi-ended), the *Meas_type* attribute takes multiple definitions. CEM version 1 supports the following *Meas_type* values for single-ended PDN with *S*-parameters:

- "0" for standard or conventional measurement configuration (default)
- "1" when the DUT is in parallel to the measurement ports (shunt connection)
- "2" when the DUT is in series to the measurement ports (series connection)

For multi-ended PDN, *Meas_type* is always "0".

The different measurement configurations and set-up for single-ended PDN extraction shall conform to the CISPR 17 specifications. Some basic aspects are discussed in Annex E.

This field is defined only when Type="S" and is optional. When absent, it defaults to "0".

7.8.2.8 Reference_impedance

The *Reference_impedance* attribute is used by the CEM parser in order to compute the impedance of the specific lead *Id* when network parameters are used for PDN definition.

This field is defined only when Type="S" and is optional. Unless defined with a numerical value, the default is "50ohm" as per CISPR 17.

7.8.2.9 Use

The *Use* attribute tells the CEM parser to use one of the values in the *Param_order* attribute. When explicitly defined, the corresponding parameter is used as PDN. If undefined, the first "Sij" or "Zij" or "Yij" term is used by default.

This field is optional.

7.8.2.10 Netlist

The Netlist keyword is used to define the PDN as a SPICE netlist that represents the electrical connectivity of the PDN elements. The electrical connectivity flow shall be as per SPICE specification [2]. A detailed description of the circuit model is presented in 7.8.3.4 and 7.8.3.5.

This field is required if Type="Ckt".

7.8.2.11 Unit_freq and Unit_param

The parameter units are defined under the *Unit_freq* and *Unit_param* tags for frequency and data respectively. If absent, *Unit_freq* defaults to basic units, "Hz". Table 6 below shows the default value of the *Unit_param* attribute as a function of data type and format when undefined:

Table 6 – Valid data formats and their default units in the *Pdn* section

Data type	Data format	Default Unit_param values
S	RI	1
	MA or MAG	1
	DB or DBMAG	dB
Z	RI	ohm
	MA or MAG	ohm
	DB or DBMAG	dBohm
Y	RI	S
	MA or MAG	S
	DB or DBMAG	dBS

See A.2.5.5 for valid units.

These fields are optional.

7.8.2.12 Power_level

The measurement power level used for PDN extraction is specified in the *Power_level* field. For simplicity reasons, the power level along with units should be defined together as discussed in A.2.5.3.

This is an optional field. If absent, it defaults to "0dBm".

7.8.2.13 Data_files and List

The *Data_files* tag is used to differentiate between inline and external data. If the PDN data is defined in an external file, the link to the file shall be specified within the *Data_files* tag. Inline data is directly defined under the top level *Lead* tag within a *List* tag if needed. Files with extensions listed in Table 7 are valid.

Table 7 – Valid file extensions in the *Pdn* section

File extension	Common name
dat or DAT	Data file
csv or CSV	Comma separated values
txt or TXT	Text file
snp or SnP	Touchstone, n is an integer (1,2,...)
cir or CIR	Circuit file (Netlist)
lib or LIB	Library file (Netlist)
net or NET	Netlist file

List is not defined if circuit model is used for the PDN model. *Data_files* is defined for network parameters (*S/Z/Y*) and *Netlist* definitions.

This is a required field; a unique *List* or *Data_files* element shall be used to define the PDN data.

7.8.3 Description

7.8.3.1 General

As stated in 6.1, the PDN of the ICEM-CE macro-model can either be a single block (generic IC macro model) or consist of multiple blocks (and sub-models) for block-based ICEM-CE modelling. Depending on the simulation tool used, the PDN could be represented with different parameters, typically S -, Z -, Y - parameters, or as a circuit/netlist.

Table 8 lists the valid fields of a Lead structure PDN definition and their usage.

Table 8 – Valid fields of the Lead keyword in the *Pdn* section

Field	Description	Usage	Rules
Id	Identifier or pin number	Required	One or more Id's, external or internal nodes previously defined in the Lead-definitions section
Ground_id	Identifier of the ground pin	Required if Type="S" or "Z" or "Y"	One unique Id previously defined in the Lead-definitions section
Blockname	Name of the PDN block	Optional	Valid string (see 7.8.1)
Type	PDN source parameter (S/Z/Y/Circuit)	Optional	Valid String: "S" (default) or "Z" or "Y" or "Ckt"
Param_order	Order in which parameters are defined	Optional	Valid string (see 7.8.1)
Format	Data format	Optional	"RI" (default) or "DB" or "MA"
Meas_type	Method implemented for performing PDN measurements	Optional	"0" or "1" or "2"
Reference_impedance	Reference impedance used in PDN measurements	Optional	Valid numerical value with units: "50ohm"
Use	Parameter that is to be specifically used	Optional	One of the values in Param_order other than the Frequency term. Default: First S or Z or Y term
Netlist	SPICE type netlist	Optional	See 7.8.3.4 and 7.8.3.5
Unit_freq	Frequency units	Optional	See A.2.5.5
Unit_param	Parameter units	Optional	See A.2.5.5
Power_level	Measurement power level	Optional	See A.2.5.3
Data_files	PDN source parameter defined in an external file	Required if not List	Valid for external netlist file or external network parameter file.
List	PDN data entries in the form of a list defined inline	Required if not Data_files	Valid numerical values with or without units in the specified order

The compositions of *Id*, *Type*, *Param_order*, *Meas_type*, *Reference_impedance* and *Format* fields have been previously discussed in 7.8.1. For single-ended PDN, *Id* is a unique string and for multi-ended PDN, *Id* contains multiple pin Id's referenced together and separated by a comma (",") character. A single-ended PDN is shown in Example 1 and a multi-ended PDN definition is illustrated in Example 2.

EXAMPLE 1 A single-ended PDN of lead "1" with respect to the ground lead "5", defined using S -parameters, is represented as:

```
<Lead Id="1" Ground_id="5" Type="S".../>
```

EXAMPLE 2 A multi-ended PDN of leads "1", "2", "3", "4" etc. with respect to the ground lead "5", defined using *S*-parameters, is represented as:

```
<Lead Id="1,2,3,4,..." Ground_id="5" Type="S" ... />
```

For block-based or sub-model-based IC macro-modelling, the name of the PDN sub-block is specified using the *Blockname* field. This field is intended for informational purposes only. The CEM parser does not interpret this field. The PDN, IBC and the IA sub-models share the same namespace. Therefore, if both a PDN (and IBC) and an IA hold the same value (name string) in their respective *Blockname* element, they are regrouped together as a single ICEM-CE sub-model component. An example PDN in a block-based ICEM-CE macro-model is illustrated in Example 3.

EXAMPLE 3 A single-ended PDN representing the "Digital_block_A" is represented as

```
<Lead Id="1" Blockname="Digital_block_A".../>
```

Data can either be defined directly within the CEML file or can be defined using external data files for both network parameter and circuit (netlist) definition.

7.8.3.2 Inline data – network parameter

In common practice, network parameters such as *S*-, *Z*- and *Y*- parameters are extracted and defined in the frequency domain. If they are defined for a single frequency, then they can be defined within the *Lead* keyword directly. If the PDN is defined for multiple frequencies (more than one frequency), then data shall be written within the *List* keyword. It is also possible to represent *S*-parameters in the time-domain.

The different leads (internal and external), whose PDN is to be defined, shall be defined using the *Id* attribute and the ground lead (lead with Mode="GND" in *Lead_definitions* section) used as return signal path (ideal reference ground) shall be specified using the *Ground_id* attribute.

The frequency units shall be defined within the *Unit_freq* keyword and data units shall be specified using *Unit_param* keyword. The power level used for measurement shall be defined using *Power_level* tag and the reference impedance using the *Reference_impedance* tag. See 7.8.2 for more information on these attributes and their default values.

NOTE The lead *Ids*' value need not be coherent with the port numbers used for measurements. The values are matched with the port numbers in sequence, from left to right, in the touchstone format.

Example 1 illustrates a multi-ended PDN of leads "1","2","3","4" with respect to the ground lead "5". *S*-parameter data (frequency-domain), extracted with power of "10dBm", is defined in Real/Imaginary format and the *Param_order* field is defined as "Freq,All" which signifies that all the *S*-parameters are defined in sequence (touchstone format): Frequency, Re{S11}, Im{S11}, Re{S12}, Im{S12}, Re{S13}, Im{S13}, Re{S14}, Im{S14}, and so on...

EXAMPLE 1 Example of a frequency-domain multi-ended PDN.

```
<Pdn>
  <!-- S Parameters inline data, multi-frequencies -->
  <Lead Id="1,2,3,4";Ground_id="5" Type="S"; Param_order="Freq,All"; Format="RI">
    <Unit_freq>kHz</Unit_freq>
    <Unit_param>1</Unit_param>
    <Power_level>10dBm</Power_level>
    <List>
100 0.004 0.002 0.0018 0.0023 -0.0013 -0.0021 0.99 -0.0037 0.0019 0.002 0.0042 0.0024
0.998 -0.0038 -0.0011 -0.002 -0.0013 -0.0021 0.998 -0.0037 0.0041 0.0027 0.0013
0.0026 0.998 -0.0036 -0.001 -0.0019 0.0013 0.0027 0.0022 0.0027
...
    </List>
  </Lead>
</Pdn>
```

7.8.3.3 External file data – network parameter

For directing the CEM parser to use the necessary external data file in ASCII format, definition shall be made using *Data_files* keyword. If the external file has a touchstone extension (*.snp or *.SnP where n is an integer 1, 2, ...), *Unit_param*, *Unit_freq*, *Param_order*, *Reference_impedance* and *Type* definitions are not required (ignored if explicitly defined). For all other file extensions (*.dat, *.csv, *.txt) the defined values are used. If absent, default values are automatically considered. See Table 8 for default properties. Similar to inline definition seen in 7.8.3.2, the *Ground_id* field is required and the *Meas_type* field is optional.

Measurement power level should always be a part of the external file as a comment. Comment lines in the PDN file shall start with an exclamation symbol ("!") as shown below in Example 1.

EXAMPLE 1 !This is a comment line

Figure 19 shows an example of the four-port *S*-parameter file in Touchstone format, in the frequency-domain, with *data* in dB format. This file is used to define the PDN of leads "1","2","3","4" with respect to ground lead "5" (see Example 2). The model is block based with a name defined.

NOTE The measurement power level is included as a comment in the PDN file.

```
!Agilent Technologies,E5071C,MY46100659,A.07.02
!Date: Wed Jan 23 09:49:50 2013
!Data & Calibration Information:
!Freq
!Power_level: 10dBm
# Hz S RI R 50
100000 0.004274546 0.00241179 0.001886209 0.002311387 -0.001362228 -0.002127369 0.9983465 -0.003734921
0.00190445 0.00233215 0.004235411 0.002405719 0.998264 -0.003837676 -0.001106818 -0.002012462
-0.001369703 -0.002143684 0.9981545 -0.003736369 0.004101523 0.002755493 0.001347644 0.002668388
0.9983041 -0.003624817 -0.001178987 -0.001959733 0.001348257 0.002710212 0.002291472 0.002748889
100679 0.00422729 0.002403554 0.001871881 0.002334189 -0.001349262 -0.002154236 0.998417 -0.003759051
0.001902756 0.002356657 0.004170153 0.002522967 0.9982947 -0.003870883 -0.001127652 -0.002029074
-0.001356177 -0.002142675 0.9980684 -0.003715199 0.00410353 0.002777282 0.001371339 0.002686834
0.9982006 -0.003591272 -0.001147017 -0.001990634 0.001357759 0.002703257 0.002186219 0.002865791
101359 0.004252984 0.002479831 0.001827719 0.002357767 -0.001379005 -0.002139817 0.9983823 -0.00378314
0.001916786 0.002348517 0.004238612 0.002526162 0.9983067 -0.003854939 -0.001140214 -0.002014498
-0.00136173 -0.002137753 0.9980044 -0.003766839 0.004121085 0.002838985 0.001346626 0.00268385
0.9981698 -0.003693599 0.001171574 -0.00198756 0.001361229 0.00271512 0.002269425 0.002912863
```

IEC

Figure 19 – PDN represented as *S*-parameters in Touchstone format

EXAMPLE 2 Describing a PDN using an external data file

```
<Pdn>
  <Lead Id="1,2,3,4;Ground_id="5">
    <Power_level>0dBm</Power_level>
    <Blockname>memory_block</Blockname>
    <Data_files>
      Pin1234_PDN.s4p
    </Data_files>
  </Lead>
</Pdn>
```

7.8.3.4 Inline circuit definition

Another way to define the PDN is in the form of a schematic (circuit). CEM parser understands that the PDN has a circuit definition when the *Type* attribute of a particular lead or port is set to "Ckt". The circuit definition shall be directly declared within the *Netlist* tag under the *Lead* section. The *Netlist* section shall contain the following definitions:

- *Power_level* definition
- SPICE like data statements defining the electrical connectivity of components [2]

All SPICE recognised elements can be defined in this section and the representation shall follow SPICE syntax. The data statements have a free format and consist of fields separated by a blank. If one wants to continue a statement to the next line, one uses a "+" sign (continuation sign) at the beginning of the next line. Numbers can be integers, or floating points [2].

EXAMPLE 1 A resistor R1 with a value of 1 kΩ connecting nodes "In" and "Out" can be represented as:

R1 In Out 1e3

All nodes used in the data statements share the same namespace with the lead Ids defined in the *Lead_definition* section. The *Ground_id* field is optional (ignored by the CEM parser if defined) when defining the PDN using a netlist. The *Id* attribute of the *Lead* keyword shall carry the Id information of the internal, external and ground pins (as defined in the *Lead_definitions* section).

Within the several data statements used to define the netlist, at least one node shall correspond to the lead defined as internal or external and at least one node shall be GND in the *Lead_definition* section. All other interior nodes used for circuit definition shall not carry the lead Id's defined in the *Lead_definition*. For compatibility reasons between several SPICE kinds, "0" is not allowed as node number.

The *Netlist* keyword definition is shown in Table 9.

Table 9 – Netlist definition

<i>Netlist</i>
Kind: SPICE netlist format (optional) default: "SPICE3"
Data_files: SPICE netlist in external file (optional)

The optional attribute *Kind* tells the CEM parser that the defined sub-circuit follows a specific syntax. CEM version 1 supports industry-standard SPICE like netlist syntaxes.

The *Kind* field defaults to generic "SPICE3" if absent.

Example 1 illustrates an example PDN using a SPICE netlist. The same can also be expressed as a SPICE macro-model as illustrated in Example 2. In Example 2, it is assumed that "PDN_pin12345" is defined in the *Macromodels* section (see 7.6).

EXAMPLE 1 An example of PDN using inline netlist definition. In this case, lead with Id="5" is used as ground lead.

```
<Pdn>
  <!-- General circuit model -->
  <Lead Id="1,2,3,4,5" Type="Ckt">
    <Power_level>10dBm</Power_level>
    <Netlist Kind="SPICE3">
      L1 1 n00 1.820e-09
      R1 n01 n02 2.408e-01
      C1 n00 5 1e-12
      L2 2 n10 1.828e-09
      R2 n11 n12 2.434e-01
      C2 n10 5 1e-12
      ...
    </Netlist>
  </Lead>
</Pdn>
```

EXAMPLE 2 An example of a PDN using SPICE macro-models. In this case, lead with Id="5" is used as ground lead. XPDN1 is the sub-circuit's identifier in the main circuit.

```

<Pdn>
  <!-- General circuit model using macro-model-->
  <Lead Id="1,2,3,4,5" Type="Ckt">
    <Power_level>10dBm</Power_level>
    <Netlist>
      XPDN1 1 2 3 4 5 PDN_pin12345
    </Netlist>
  </Lead>
</Pdn>

```

7.8.3.5 External file containing the netlist

Instead of the inline netlist definition (see 7.8.3.4), it is possible to define the PDN in an external netlist file. The definition is done with the *Data_files* tag under the *Netlist* section. All rules with respect to the node numbers used in the netlist definition are the same as defined in 7.8.3.4.

In some cases, a netlist file could contain the main circuit definition and sub-circuit definitions together. To avoid ambiguity, the CEM parses only the data statements defined outside the SPICE keywords: ".SUBCKT" and ".ENDS". These keywords are not case sensitive. For parsing the sub-circuit element, it is mandatory to define the same file within the *Macromodels* section (see 7.6).

Example 1 illustrates the use of an external netlist file and the PDN is defined as a block.

EXAMPLE 1 An example of a PDN using external file containing the netlist definition. In this case, lead with Id="5" is used as ground lead.

```

<Pdn>
  <!-- General circuit model using netlist -->
  <Lead Id="1,2,3,4,5" Type="Ckt">
    <Power_level>0dBm</Power_level>
    <Blockname>block2</Blockname>
    <Netlist>
      <Data_files>Pin12345_PDN_Ckt.lib</Data_files>
    </Netlist>
  </Lead>
</Pdn>

```

7.9 IBC

7.9.1 General

Similar to the PDN, different IBC complexity levels can be considered. The simplest configuration level is modelling the IBC with two internal terminals linking two internal terminals of different PDN blocks. IBC configuration interfacing many PDN blocks and many internal terminals can become very complex.

The *Ibc* section of the ICEM-CE macro-model contains the IBC data that describes the ICEM-CE PDN. The data shall be defined within the *Ibc* keyword as follows:

```

<Ibc>
  IBC data
</Ibc>

```

An IBC data is defined for interfacing different PDN blocks and thus the definition shall be done within a *Lead* tag (similar to the PDN). See Figure 17 for the structural hierarchy. Many *Lead* elements can be listed one below another within the *Ibc* section.

7.9.2 Attribute definitions

The general structure of the *Lead* keyword is coherent with that listed in Table 5. However, some attribute specifications are different for the IBC network. These differences are tabulated in Table 10. All other fields in Table 5 remain unchanged.

Table 10 – Differences between the *Pdn* and *Ibc* section fields

Field	Description	Usage		Rules	
		PDN	IBC	PDN	IBC
Id	Identifier or pin number	Required	Required	<u>One or more Ids</u> previously defined in the Lead-definitions section as <u>external or internal terminals</u>	<u>Two or more Id's</u> previously defined in the Lead-definitions section as <u>internal terminals</u>
Ground_id	Identifier or pin number of the return signal pin	Required for Type="S" or "Z" or "Y"	NA	One unique Id previously defined as GND in the Lead-definitions section	NA

Table 11 lists the valid fields of a *Lead* structure for an IBC definition and their usage.

Table 11 – Valid fields of the *Lead* keyword for IBC definition

Field	Description	Usage	Rules
Id	Identifier or pin number	Required	Two or more Id's previously defined in the Lead-definitions section as internal terminals
Blockname	Name of the IBC component	Optional	Valid string (see 7.8.1)
Type	IBC source parameter (S/Z/Y/Circuit)	Optional	Valid String: "S" (default) or "Z" or "Y" or "Ckt"
Param_order	Order in which IBC parameters are defined	Optional	Valid string (see 7.8.1)
Format	Data format	Optional	"RI" (default) or "DB" or "MA"
Meas_type	Method implemented for performing IBC measurements	Optional "0"	Valid numerical value: "0" or "1" or "2" Default: "0"
Reference_impedance	Reference impedance used in IBC measurements	Optional	Valid numerical value with units: "50ohm"
Use	Parameter that is to be specifically used	Optional	One of the values in <i>Param_order</i> other than the Frequency term. Default: First S or Z or Y term
Netlist	SPICE type netlist	Optional	See 7.8.3.4 and 7.8.3.5
Unit_freq	Frequency units	Optional	See A.2.5.5
Unit_param	Parameter units	Optional	See A.2.5.5
Power_level	Measurement power level	Optional	See A.2.5.3
Data_files	IBC source parameter defined in an external file	Required if not List	Valid for external netlist file or external network parameter file.
List	IBC data entries in the form of a list defined inline	Required if not Data_files	Valid numerical values with or without units in the specified order

Any relevant information required for correct understanding (and usage) of the IBC shall be optionally defined within the *Notes* and *Documentation* tags.

7.10 IA

7.10.1 General

The *Ia* section of the ICEM-CE macro-model contains the IA data that describes the model. The data shall be defined within the *Ia* keyword as follows:

```
<Ia>
    IA data
</Ia>
```

Similar to the PDN and IBC definition, IA data is defined for IC pin or terminal and thus the definition shall be done within a *Lead* tag (see Figure 17 for the structural hierarchy). Table 12 lists the various recognized fields of the Lead keyword in the IA section:

Table 12 – Definition of the *Lead* keyword in the *Ia* section

<i>Lead</i>
Id: pin identity as a valid string (required)
Type: IA source parameter -"time" for time-domain data and "freq" for frequency-domain data (required)
Blockname: IA block name as a valid string (optional)
Voltage: IA data defined as a voltage source (required if not Current)
Current: IA data defined as a current source (required if not Voltage)

The conditions of validity (frequency or time range, power-supply conditions, time or phase origin, etc.) of the IA sub-model shall be specified in the *Validity* section (see 7.7).

NOTE The frequency range (or time range) of validity of the ICEM-CE macro-model is the common frequency (or time) range of the PDN and IA.

As per IEC TS 62433-1, any other relevant information required for correct understanding (and usage) of the IA shall be optionally defined within the *Notes* and *Documentation* tags. Details such as IC's operating mode, decoupling capacitors on supply lines, activated functions, grounding details, datasheets and test-reports can be defined.

7.10.2 Attribute definitions

7.10.2.1 Id

The identity, *Id*, of the leads used in the IA definition should have been previously defined as internal pins in the *Lead_definitions* tag as described in 7.5. Only two Ids are permitted per Lead definition since the IA source is always connected between two terminals. The first Id terminal is assumed to be the positive node and the second Id terminal is the negative node. Current flows from the positive node to the negative node.

This is a required field.

7.10.2.2 Type

The *Type* attribute is used to represent the type of the IA data. CEML version 1 supports two types: "time" for time-domain IA data and "freq" for IA data in the frequency domain.

This is a required field.

7.10.2.3 Blockname

The *Blockname* field is used to define the name of the IA block. This field is required for the IA. If both the PDN and the IA share the same block name, then they are regrouped together by the CEM parser as a sub-block.

This is an optional field.

7.10.2.4 Voltage and Current

The main IA source is defined under the *Voltage* and/or *Current* sections.

At least one definition (*Voltage* field or *Current* field) is required in the IA section. Voltage data is defined within the *Voltage* keyword and the current data is defined within the *Current* keyword.

The common definition of the *Voltage* and *Current* sections is shown in Table 13.

Table 13 – Voltage and Current definition

<i>Voltage / Current</i>
List: IA data list (required if not <i>Data_files</i> or <i>Pulse</i>)
Data_files: IA source parameter defined in an external file (required if not <i>List</i> or <i>Pulse</i>)
Pulse: IA data as a simulated waveform (required if not <i>List</i> or <i>Data_files</i> , valid only if <i>Type</i> ="time")
Param_order: Order in which IA parameters are defined (optional)
Format: IA Data format (optional)
Unit_voltage: Unit of the voltage terms (optional)
Unit_current: Unit of the current terms (optional)
Unit_freq: Unit of the frequency terms (optional)
Unit_time: Unit of the time terms (optional)

These sections have one required child element: *List* or *Data_files* or *Pulse*; either one shall be used to define the IA data. All other attributes and elements in Table 13 are optional. When absent, default values and units are used.

7.10.2.5 Data_files and List

The definitions of the *Data_files* and *List* keywords are similar to that seen with the PDN definition (see 7.8.2.13). The *Data_files* tag is used to differentiate between inline and external data. If the PDN data is defined in an external file, the link to the file(s) shall be specified within the *Data_files* tag. Inline data is directly defined under the top level *Voltage* or *Current* tag within a *List* tag if needed. Files with extensions listed in Table 14 are allowed for defining the IA.

Table 14 – Valid file extensions in the *Ia* section

File extension	Common name
dat or DAT	Data file
csv or CSV	Comma separated values
txt or TXT	Text file

7.10.2.6 Pulse

This keyword is dedicated for representing the IA sub-model using a simulated waveform, such as triangular or pulse waveform; it is permitted to do so only for defining the IA in the time-domain, i.e. Type="time" (see 7.10.2.2). The *Pulse* section is directly defined under the top level *Voltage* or *Current* tag.

This section is required if no *List* or *Data_files* element has been used to define the voltage or current data defining the IA. Table 15 lists the various recognized fields of the *Pulse* keyword in the *Voltage* or *Current* section.

Table 15 – Definition of the *Pulse* keyword in the *Voltage* or *Current* section

<i>Pulse</i>
Initial_value: Initial or minimum amplitude (required)
Final_value: Final or maximum amplitude (required)
Period: Time period of the signal (required if not Frequency)
Frequency: Frequency of the signal (required if not Period)
Time_delay: Time delay before the leading edge of the signal (optional) default:"0"
Rise_time: Time taken by the signal to rise from the initial value to reach the final value (optional) default:"0"
Fall_time: Time taken by the signal to fall from final value to the initial value (optional) default:"0"
On_time: Time duration for which the signal maintains its final value (optional) default: half of period
Ncycles: Number of repetition cycles (optional) default: infinite

A *Pulse* section has three required fields: *Initial_value*, *Final_value* and *Period* or *Frequency* (inverse of period). All other attributes in Table 15 are optional. The value of all the fields is a numerical value with or without units. See A.2.5.5 for valid units.

The base units of these fields are described in Table 16; see A.2.5.5 for valid scaling factors. Parameter units may also be set using the keywords defined in 7.10.2.9.

Table 16 – Base units of the *Pulse* section's fields

Attribute	Base units	
	Voltage section	Current section
Initial_value	V	A
Final_value	V	A
Period	s	
Frequency	Hz	
Time_delay	s	
Rise_time	s	
Fall_time	s	
On_time	s	
Ncycles	1	

A typical waveform defined by the *Pulse* keyword, with the corresponding fields, is shown in Figure 20.

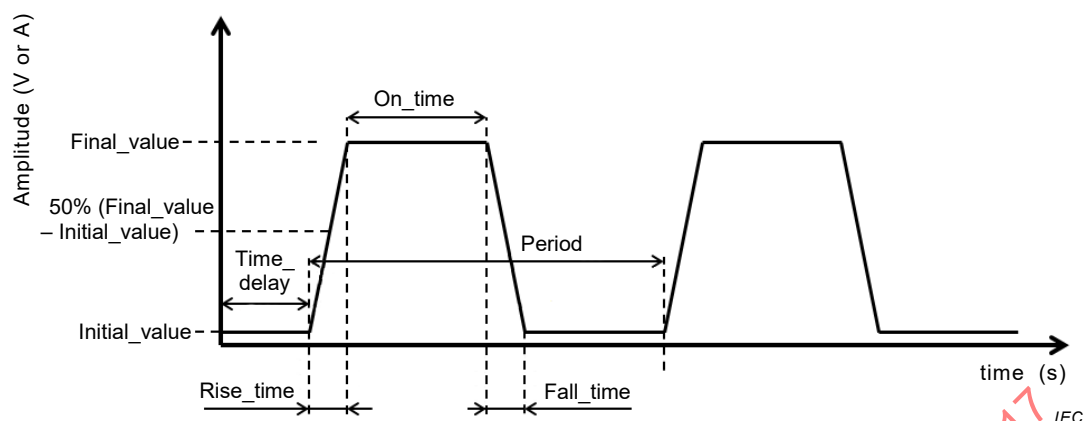


Figure 20 – Simulated IA waveform with corresponding parameters

An example of IA data using a *Pulse* tag is given in the Example below.

EXAMPLE An example *Pulse* section may be defined as shown.

```
<Ia>
  <Lead Id="1,3" Type="time">
    <Current>
      <Pulse>
        <Initial_value>0</Initial_value>
        <Final_value>50e-3</Final_value>
        <Period>125n</Period>
        <Rise_time>2n</Rise_time>
        <Fall_time>4n</Fall_time>
        <On_time>0.1n</On_time>
      </Pulse>
    </Current>
  </Lead>
</Ia>
```

7.10.2.7 Param_order

The *Param_order* attribute tells the CEM parser on how the IA data is represented. The following strings are dedicated for specifying the parameter order:

- "Freq" and "Frequency": Frequency quantity used for parameters' definition
- "Time": Time quantity used for parameters' definition
- "Volt" and "Voltage": Voltage used for parameters' definition
- "Curr" and "Current": Current used for parameters' definition

If absent, it defaults to:

- "Freq,Volt" in the *Voltage* section and "Freq,Curr" in the *Current* section if Type="freq"
- "Time,Volt" in the *Voltage* section and "Time,Curr" in the *Current* section if Type="time"

7.10.2.8 Format

The *Format* attribute decides the data format. Valid data formats for IA are:

- "RI": real/imaginary format. This is valid for frequency-domain IA.
- "DB": Magnitude in decibel scale with phase angle in degrees. This is valid for frequency-domain IA.
- "MA": Magnitude in linear scale with phase angle in degrees. This is valid for frequency-domain IA.
- "DBMAG": Magnitude in decibels without phase information. This is valid for frequency-domain and time-domain IA.

- "MAG": Magnitude in linear scale without phase information. This is valid for frequency-domain and time-domain IA.

This field is optional. If absent, the default value is "MAG".

7.10.2.9 Unit_freq, Unit_time, Unit_voltage and Unit_current

The parameter units are defined under the *Unit_freq*, *Unit_time*, *Unit_voltage* and *Unit_current* tags for frequency, time, voltage and current quantities respectively. Depending on the parent element, only the corresponding units are considered, i.e., *Unit_voltage* is allowed under the *Voltage* section and so on. Only *Unit_freq* and *Unit_time* attributes are defined for multiple sections.

If absent, *Unit_freq* defaults to "Hz" and *Unit_time* defaults to "s". See Table 17 for default values of *Unit_voltage* and *Unit_current* tags as a function of data format.

Table 17 – Valid data formats and their default units for the *Voltage* and *Current* elements

Parameter	Data format	Default Unit_voltage values	Default Unit_current values
Voltage	RI	V	-
	MA or MAG	V	-
	DB or DBMAG	dBV	-
Current	RI	-	A
	MA or MAG	-	A
	DB or DBMAG	-	dB A

If complex values are used to represent the IA (Format="MA" or "DB"), the default units of phase quantities is "degrees". See A.2.5.5 for valid units.

7.10.3 Description

7.10.3.1 Inline data

If the IA data (voltage and current) are defined for a single frequency, then they can be defined within the corresponding section (*Voltage* and *Current*) keyword directly. For multi-frequency IA, the data shall be written within the *List* keyword under the corresponding *Current* or *Voltage* section. The frequency units shall be defined within the *Unit_freq* keyword, time units within the *Unit_time* keyword, voltage units shall be specified using *Unit_voltage* keyword and that of current using *Unit_current* keyword. When absent, default values are used (see 7.10.2.9).

Example 1 illustrates the IA data of a frequency-domain ICEM-CE macro-model using inline data definition. A current quantity is used to define the IA in decibel format ("DB"). The current is specified in "dBuA" units and the frequency quantities are defined in "kHz".

EXAMPLE 1 An example of an IA section in CEML is shown.

```

<Ia>
  <Lead Id="10,11" Type="freq">
    <Current Param_order ="Freq,Current" Format="DB">
      <Unit_freq>kHz</Unit_freq>
      <Unit_current>dBuA</Unit_current>
      <List>
        7.700018e+03      -2.207281e+01
        7.800018e+03      -1.900327e+01
        7.900019e+03      -1.330357e+01
        8.000019e+03      6.099079e+01
        8.100019e+03      -1.338519e+01
        8.200019e+03      -1.912021e+01
        8.300020e+03      -2.216923e+01
        ...
      </List>
    </Current>
  </Lead>
</Ia>

```

7.10.3.2 External file data

For directing the CEM parser to use the necessary external IA data file in ASCII format, definition shall be made using *Data_files* keyword under the *Voltage* or *Current* section. All other rules are the same as that seen with inline data definition in 7.10.3. Note that all comment lines, if present, are indicated by "!" symbol in the beginning, but they not parsed by the CEM parser.

8 Requirements for parameter extraction

8.1 General

ICEM-CE model parameters can be extracted from either design information or measurements. Detailed methodology for model parameter extractions are not the purpose of this standard. This clause gives basic requirements for model parameter extractions from measurements.

NOTE Annex A gives examples of parameter extractions from design information and from measurements.

8.2 Environmental extraction constraints

The ICEM macro-model parameter extractions have to be performed under normal room temperature conditions: $23\text{ °C} \pm 5\text{ °C}$. There are no additional requirements on air pressure and humidity.

When open silicon is used, the lights shall be dimmed or switched off in order not to generate photonic effects.

NOTE Some substrate materials are hygroscopic which might affect the permittivity of the material and its loss tangent.

8.3 IA parameter extraction

IA can be derived by design data or by calculating by measurements under the following conditions:

- Nominal power supply voltages and typical loadings shall be applied to the device under test.
- Input signals such as specific test vector which correspond to the specific operational mode shall be applied.

8.4 PDN parameter extraction

The PDN parameters shall be derived by analyzing impedances between the terminals under the nominal power supplies.

The derived parameters are only suited and re-useable for conducted emission simulations when the PVT (process, voltage and temperature) conditions are the same.

NOTE 1 PDN is static, but due to the N-well and gate-oxide capacitances that are involved, the power supply voltage will affect it.

NOTE 2 Most of the impedance parameters can be derived from measurements using an impedance analyzer or a VNA (Vector Network Analyzer).

8.5 IBC parameter extraction

The IBC impedance can be derived from the Vdd to Vss impedance by subtracting the impedance part that belongs to the PDN. A detailed method should be elaborated in the future.

IECNORM.COM : Click to view the full PDF of IEC 62433-2:2017

Annex A (normative)

Preliminary definitions for XML representation

A.1 XML basics

A.1.1 XML declaration

Although the XML declaration is optional in an XML file, the ICEM-CE macro-model file should include an XML declaration, dedicated to basic XML parsers.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

The XML declaration shall be the first line of the file.

A.1.2 Basic elements

All information is saved in the form of XML elements. Each element starts with a start-tag and ends with an end-tag. The start-tag consists of a keyword enclosed in triangular brackets, `<Keyword>`. The end-tag consists of the same keyword prefixed by the character "/" and enclosed in brackets, `</Keyword>`. Content in the form of text is enclosed by a start-tag and an end-tag.

An example of an element is given below:

```
<Keyword>          <!-- start-tag -->
    text           <!-- content -->
</Keyword>         <!-- end-tag -->
```

It is also allowed to write an element on the same line, for example, to include short content:

```
<Keyword>text</Keyword>
```

The contents of an element may consist of one or more other elements or a value (numerical, or alphanumerical). For clarity, tab characters may be used for indenting. Except when used for surrounding keywords, triangular brackets "<" and ">" shall not be part of content.

The text shall use characters from the UTF-8 set (space, " ", "<", ">", "&" are not included).

An empty element may be included to indicate that a particular keyword exists, but has no content:

```
<empty_element/>
```

A.1.3 Root element

The XML file shall contain one, and only one, root element. It encloses all the other elements and is therefore the sole parent element to all the other elements. The start-tag of the root element is placed at the beginning of the file or after the XML declaration when present. The end-tag of the root element is at the last entry of the file.

The root element keyword shall not be used for any other purposes in the XML file.

A.1.4 Comments

Comments may be inserted into the file between "<!--" and "-->". An example is given below:

```
<!-- this line is a comment -->
```

Comments can be inserted anywhere in the file, except inside start- and end-tags, and written on a single line or on several lines. All text enclosed by comment brackets is considered as a comment and may be ignored.

A.1.5 Line terminations

In order to facilitate readability, it is usual to organise the file into lines. The line termination sequence shall be either a linefeed character or a carriage return character followed by a linefeed character.

A.1.6 Element hierarchy

The order of the elements is not important, but their hierarchy shall be respected. A layout example is shown below:

```
<Keyword1> ... </Keyword1>
<Keyword2>
  <Keyword21> ... </Keyword21>
  <Keyword22> ... </Keyword22>
</Keyword2>
<Keyword3> ... </Keyword3>
```

A.1.7 Element attributes

XML keyword elements can have attributes. Attributes provide additional information about elements that are not a part of the data. Attribute values shall always be quoted. Either single or double quotes can be used. Multiple attributes are either separated by a space or by a semi-colon (";") character. In contrast to elements, attributes usually cannot contain multiple values and tree structures, except for certain exceptions in the XML. An example of attribute definition is shown below:

```
<Parent Type="example"; Format="string"/>
```

In the above syntax *Type* and *Format* are attributes to the keyword *Parent*. The same terms can be defined as elements as shown below:

```
<Parent>
  <Type>example</Type>
  <Format>string</Format>
</Parent>
```

A.2 Keyword requirements

A.2.1 General

Keywords, placed in start- and end-tags, are used to introduce descriptions, values and sections that are specific to the file. Some keywords, such as Unit, List, etc., may be present in several sections. A parent keyword is required whenever a child keyword is present. The rules below ensure that the file can be correctly parsed by a specific XML parser.

A.2.2 Keyword characters

Only ASCII characters, as defined in ANSI Standard X4-1986, may be used in the files [4]. The use of characters with codes greater than hexadecimal 07E is not allowed. Also, ASCII control characters (those numerically less than hexadecimal 20) are not allowed, except for tabs or in a line termination sequence.

Only alphabetical or numerical characters can be used to write keywords. Spaces are not permitted. If needed, the underscore “_” character can separate the parts of a multi-word keyword.

A.2.3 Keyword syntax

The content of the files is case sensitive. All keywords shall be written in lower case starting with an upper case letter.

A.2.4 File structure

A.2.4.1 General

The information to be exchanged may be stored in a single XML file or in several XML and data files. The following rules and guidelines ensure that the files can be correctly located by a XML parser.

A.2.4.2 File names

To facilitate portability between operating systems, file names should have a base name of no more than forty characters followed by a period “.”, followed by a filename extension of no more than four characters. The file name and extension shall use characters from the set (space, “ ”, 0x20 is not included) seen previously in A.1.2. File name shall be defined within the “Filename” keyword as shown below:

```
<Filename>ExampleXML_file.xml</Filename>
```

A.2.4.3 File Paths

In order to ensure portability and compressibility, only relative paths can be used to define a path name. An absolute path is not exportable and is not permitted. The relative path shall start with “./” to indicate that the path name of the picture file will be appended to the path of the current XML file. It is not permitted to browse to a higher level from the current XML path (e.g. by using “../”). A file name without “./” is assumed to be located in the same directory as the current XML file.

A.2.4.4 Single XML file

When the information is contained in a single XML file, it shall conform to the rules and guidelines applicable to XML files as described previously.

Data is included in the PDN or IA section of the file within the XML element using the keyword “List” whenever required.

A.2.4.5 Multiple XML files

The XML document is divided into several sections having the root element as parent. The model’s information is then defined under the root section using keywords such as *Pdn*, *Ia* etc. Each XML file may contain one or more sections and shall conform to the rules and guidelines applicable to XML files as described previously. A section shall be present in only one of the XML files.

In order to ensure portability and compressibility, all the XML files shall be placed in the same directory, as shown in Figure A.1. The XML parser shall parse all the files that are in the main directory.

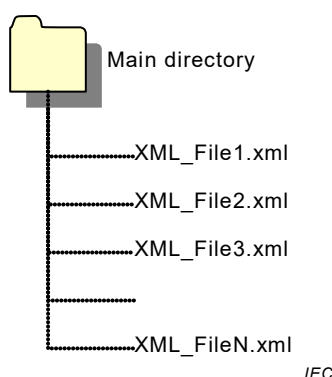
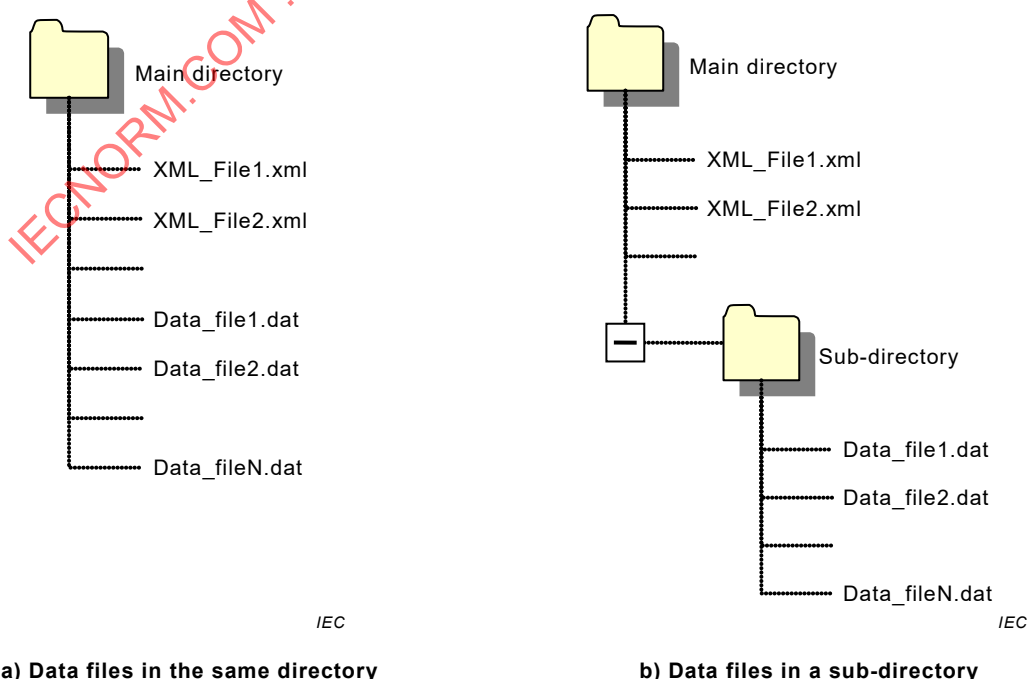


Figure A.1 – Multiple XML (CEML) files

A.2.4.6 Separate data files

The information may be contained a single or multiple XML files and the data contained in one or more additional data files. The XML files shall conform to the rules and guidelines applicable to XML files as described above. The data files can contain lines of data and header information shall be specified with exclamation character ("!") as the first character of each header line (non-data line). As an exception, for data files with a touchstone extension (*.snp, n = 1,2,...) used for PDN definition, the lines starting with an hash character("#") is treated as the "option line" that specifies data format and units pertaining to touchstone data. Only one "option line" is allowed per touchstone data file. The names and paths of the data files are defined by the keyword *Data_files* and shall conform to A.2.4.2 and A.2.4.3.

In order to ensure portability and compressibility, the data files shall be placed either in the same directory as the XML files or in a sub-directory located at the same level or a lower level as the XML files, as shown in Figure A.2. It is not permitted to locate the additional files at a higher level than the XML files.



a) Data files in the same directory

b) Data files in a sub-directory

Figure A.2 – XML files with data files (*.dat)

A.2.4.7 Additional files

A XML file may contain references to other files such as document files (Keyword: *Documentation*). In order to ensure portability and compressibility, these additional files shall be placed either in the same directory as the single XML file or in a sub-directory located at the same level or a lower level as the XML files, as shown in Figure A.3. It is not permitted to locate the additional files at a higher level than the XML files.

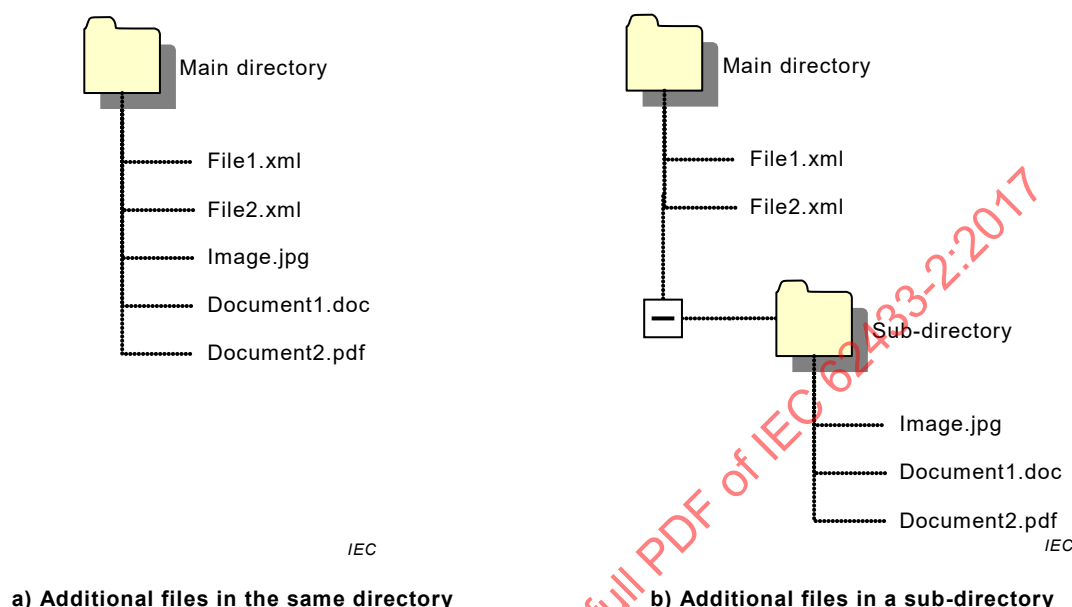


Figure A.3 – XML files with additional files

A.2.4.8 File compression

When compressing the file system, care shall be taken to include the paths of the various XML and data files in the compressed file. This ensures that, when decompressed, the file structure is conserved. The paths are not required when all files are stored in the same directory.

A.2.5 Values

A.2.5.1 General

When an element contains a value, this may be a numerical value (e.g. 123.45), or a numerical value with units (e.g. 123.45MHz).

A.2.5.2 Numerical syntax

Numerical values may be expressed in decimal form with the period (".") as the decimal separator (e.g. 123.45) or in scientific form (e.g. 1.2345e2). Spaces " " and commas ",", which are often used as thousand separators, and other characters are not allowed.

In cases where several numerical values are required, they shall be separated by spaces " " or tab characters.

A.2.5.3 Numerical with units syntax

The numerical value (see A.2.5.2) is followed by valid units, as described in A.2.5.5 (e.g. 123.45 MHz). Spaces are not allowed between the numerical value and the units.

A.2.5.4 Text string

A text string may represent a word recognized by the XML parser or it may be a file name, a description, etc. A text string may contain any of the alphanumerical characters given in A.1.2.

A.2.5.5 Valid units

Units shall be expressed in SI units or derived SI units. Valid units are:

V for Volt	ohm for Ohm	S for Siemens =1/Ohm
A for Ampere	Hz for Hertz	m for metre
W for Watt	s for second	H for Henry
F for Farads	ohm.m for resistivity	Celsius for degree Celsius
K for Kelvin	1 for dimensionless quantity	

Units are case sensitive.

Angles are expressed in degrees. The symbol "°" is not required.

Percentage is expressed as "%".

Valid scaling factors are:

T = tera: 1e12	c = centi: 1e-2	p = pico: 1e-12
G = giga: 1e9	m = milli: 1e-3	f = femto: 1e-15
M = mega: 1e6	u = micro: 1e-6	a = atto: 1e-18
k = kilo: 1e3	n = nano: 1e-9	

When no scaling factors are specified, the appropriate base units are assumed. These are volts, Amperes, Watts, Ohms, Siemens, Hertz, metres, seconds, Henry, Farads, ohm.m, Celsius and Kelvin. Abbreviations for the units (e.g., pV, nA, ms, MHz) shall be used, except Ohm, which shall be written in full. A list of valid logarithmic units is shown in Table A.1.

Table A.1 – Valid logarithmic units

Usage	Symbol	Unit	Reference
Ratio	dB	Decibel	1
Power	dBW	Decibel Watt	1W
Power	dBmW	Decibel milli-Watt	1mW
Power	dBuW	Decibel micro-Watt	1μW
Voltage	dBV	Decibel Volt	1V
Voltage	dBmV	Decibel milli-Volt	1mV
Voltage	dBuV	Decibel micro-Volt	1μV
Ampere	dBA	Decibel Ampere	1A
Ampere	dBmA	Decibel milli-Ampere	1mA
Ampere	dBuA	Decibel micro-Ampere	1μA
Impedance	dBohm	Decibel Ohms	1ohm
Admittance	dBS	Decibel Siemens	1S

Unit definitions can be made in any section and with the *Unit* keyword. A typical unit description is shown below:

```
<Unit>  
    kHz  
</Unit>
```

IECNORM.COM : Click to view the full PDF of IEC 62433-2:2017

Annex B (normative)

CEML valid keywords and usage

B.1 Root element keywords

The following keywords are placed at the beginning of the file after the XML definition:

Keyword	Parent	Description	Presence	Example
Header	Root element	Specifies a header section	Required	<Header> </Header>
Lead_definitions	Root element	Specifies a lead definitions section. The details defined in this section is used for in all other sections	Required	<Lead_definitions> </Lead_definitions >
Validity	Root element	Specifies the validity section	Required	<Validity> ... </Validity>
Macromodels	Root element	Specifies a spice macromodels section	Optional	<Macromodels> </Macromodels>
Pdn	Root element	Specifies the PDN data	Required	<Pdn> </Pdn>
Ia	Root element	Specifies the IA data	Required	<Ia> </Ia>
Ibc	Root element	Specifies the IBC data	Optional	<Ibc> </Ibc>

B.2 File header keywords

The following keywords are placed at the beginning of the file after the root element start-tag:

Keyword	Parent	Description	Presence	Example
Cem_ver	Header element	Specifies the version of file format (1.0). Should follow the header element. Informs parsers of the version of XML exchange format used and allows them to know what keywords to expect.	Required	<Cem_ver>1.0</Cem_ver>
Filename	Header element	Specifies the file name. Normally follows the keyword: Cem_ver.	Required	<Filename> My_file.xml </Filename>
File_ver	Header element	Tracks the revision level of a particular .xml file. Revision level is set at the discretion of the originator of the file.	Required	<File_ver>1.0</File_ver>
Date	Header element	The value can contain blanks, be of any format, but should be limited to a maximum of 20 characters. The month should be spelled out for clarity. The parser considers this information as a data string and does not interpret it.	Optional	<Date> January 1, 2013 </Date>
Author	Header element	The value can contain blanks and be of any format. The parser considers this information as a data string and does not interpret it.	Optional	<Author> Name 1 Name2 </Author>
Dut	Header element	The value can contain blanks and be of any format. The parser considers this information as a data string and does not interpret it.	Optional	<Dut> 32-bit microcontroller </Dut>
Meas_method	Header element	The value can contain blanks and be of any format. The parser considers this information as a data string and does not interpret it.	Optional	<Meas_method> VNA, S param </Meas_method>
Disclaimer	Header element	The value can contain blanks and be of any format. The parser considers this information as a data string and does not interpret it.	Optional	<Disclaimer> This file contains results of model extraction. Other use is not guaranteed </Disclaimer>
Copyright	Header element	The value can contain blanks and be of any format. The parser considers this information as a data string and does not interpret it.	Optional	<Copyright> Copyright 2013, XYZ Corp., All Rights Reserved </Copyright>

B.3 Validity section keywords

The following keywords are placed in the root element start-tag:

Keyword	Parent	Description	Presence	Example
Power_supply	Validity element	Specifies the power supply conditions in which ICEM-CE macro-model is extracted and defined. Should follow the validity element. Informs the user of the ICEM-CE validity criteria. A valid numerical value with or without units is allowed. See A.2.5.3 for more information.	Required	<Power_supply> 9V </Power_supply>
Frequency_range	Validity element	Specifies the valid frequency range in which the frequency-domain ICEM-CE macro-model is extracted and defined. Should follow the validity element. Informs the user of the ICEM-CE validity criteria. A valid numerical value with or without units is allowed. See A.2.5.3 for more information.	Required	<Frequency_range> [10kHz-100MHz] </Frequency_range>
Time_range	Validity element	Specifies the valid time range in which the time-domain ICEM-CE macro-model is extracted and defined. Should follow the validity element. Informs the user of the ICEM-CE validity criteria. A valid numerical value with or without units is allowed. See A.2.5.3 for more information.	Required	<Time_range> [0ms-1ms] </Time_range>
Temperature_range	Validity element	Specifies the valid temperature range in which ICEM-CE macro-model is extracted and defined. Should follow the validity element. Informs the user of the ICEM-CE validity criteria. A valid numerical value with units is allowed. See A.2.5.3 for more information.	Optional	<Temperature_range> [25Celsius-55Celsius] </Temperature_range>
Phase_origin	Validity element	Specifies the origin of phase corresponding to the definition of the ICEM-CE IA. Should follow the validity element. Informs the user of the ICEM-CE validity criteria. A valid numerical value with units is allowed. See A.2.5.3 for more information.	Required	<Phase_origin> 0degrees </Phase_origin>
Time_origin	Validity element	Specifies the origin of time corresponding to the definition of the ICEM-CE IA. Should follow the validity element. Informs the user of the ICEM-CE validity criteria. A valid numerical value with units is allowed. See A.2.5.3 for more information.	Required	<Time_origin> 100ps </Time_origin>

B.4 Global keywords

The following keywords may be placed anywhere in the file, except within an XML element containing a value:

Keyword	Parent	Description	Presence	Example
Notes	Any element except those containing a value.	Optionally adds information about the setup, data, etc. The value can contain blanks, and be of any format. A notes section can be inserted anywhere in the file and the number of notes sections in the file is not limited. The parser considers this information as a data string and it does not interpret it.	Optional	<pre><Notes> Use this section for any special notes </Notes></pre>
Documentation	Any element except those containing a value.	Optionally adds the paths to files containing documentation on the project.	Optional	<pre><Documentation> Project doc.pdf Model_descr.doc </Documentation></pre>

B.5 Lead Keyword

The following keyword may be placed in the *Lead_definitions*, *Pdn*, *Ibc* and the *Ia* sections as the parent element:

Keyword	Parent	Description	Presence	Example
Lead	Any element: Lead_definitions, Pdn, Ia and Ibc	Specifies the lead element. The accepted attributes could vary from section to section	Required	<pre><Lead> ... </Lead></pre>

B.6 Lead_definitions section attributes

The Lead keyword in the *Lead_definitions* section has the following valid attributes:

Keyword	Parent	Description	Presence	Example
Id	Lead element	Specifies the identity or number of the Lead element. Only one value is allowed.	Required	<Lead Id="1"> ... </Lead>
Name	Lead element	Specifies the name of the lead element. Can be any valid string	Optional	<Lead Id="1" Name="ET1"> ... </Lead>
Mode	Lead element	Specifies the mode in which a particular lead is used: GND, None	Optional	<Lead Id="1" Name="ET1" Mode="GND"> ... </Lead>
Type	Lead element	Specifies the type in which a particular lead is used: internal, external	Optional	<Lead Id="1" Name="ET1" Mode="GND" Type="external"> ... </Lead>

B.7 Macromodels section attributes

The Subckt keyword in the *Macromodels* section has the following valid attributes:

Keyword	Parent	Description	Presence	Example
Name	Subckt element	Specifies the name of the macro-model.	Required	<Subckt Name="MD1" ...> ... </Subckt>
Nodes	Subckt element	Specifies the external nodes of the macro-model through which the sub-circuit connects to the main circuit.	Required	<Subckt Name="MD1" Nodes="N1,N2,N3"> ... </Subckt>
Kind	Subckt element	Specifies the kind of netlist used for defining the sub-circuit. See 7.6 for more information.	Optional	< Subckt Name="MD1" Nodes="N1,N2,N3" Kind="SPICE3"> ... </Subckt>
Data_files	Subckt element	Specifies the path names of the files containing a sub-circuit netlist. The path names are separated by a space character (" ") or a line termination. The file names and paths shall conform to A.2.4.2 and A.2.4.3. Only one Data_files keyword or one List keyword shall be included in the Subckt section.	Optional	< Subckt Kind="SPICE3"> ... <Data_files> PDN_Pin1_macro.lib </Data_files> ... </Subckt>

B.8 Pdn section keywords

B.8.1 Lead element keywords

The PDN data of each tested lead defined within the *Lead* tag. The Lead keyword in the *Pdn* section has the following valid attributes:

Keyword	Parent	Description	Presence	Example
Id	Lead element	Specifies the identity or number of the Lead element. One or many Ids are allowed for differentiating between single, differential or multi-ended pins. For differential or multi-ended pins, Ids shall be separated by a "," character	Required	<pre><Lead Id="1"...> ... </Lead> <Lead Id="1,2,3"...> ... </Lead></pre>
Ground_id	Lead element	Specifies the identity or number of the Lead element used as reference pin. Only one Id is allowed	Required for Type="S" or "Z" or "Y"	<pre><Lead Id="1" Ground_id="10"...> ... </Lead></pre>
Blockname	Lead element	Specifies the name of the sub block when using block based PDN modelling	Optional	<pre><Lead Id="1" Ground_id="10" Blockname="Block1"> ... </Lead></pre>
Type	Lead element	Specifies the type of PDN data: "S" for S-parameters, "Z" for Z-parameters, "Y" for Y-parameters, "Ckt" for circuit definition. See 7.8.2.4	Optional	<pre><Lead Id="1" Type="Z"> ... </Lead></pre>
Param_order	Lead element	Specifies the order in which the PDN parameters are defined. See 7.8.2.5 for detailed information. Undefined if using Data_files in touchstone format and Netlist	Optional	<pre><Lead Id="1" Type="Z" Param_order="Freq,Z 11"> ... </Lead></pre>
Format	Lead element	Specifies the PDN data format. See 7.8.2.6. Undefined if using Data_files in touchstone format and Netlist	Optional	<pre><Lead Id="1" Type="Z" Param_order="Freq,Z 11" Format="MA"> ... </Lead></pre>
Meas_type	Lead element	Specifies the method implemented for performing PDN measurements. See 7.8.2.7. Undefined if using Netlist	Optional	<pre><Lead Id="1" Type="S" Param_order="Freq,S 21" Format="MA" Meas_type="1"> ... </Lead></pre>
Reference_impedance	Lead element	Specifies the reference impedance used in performing PDN measurements. Undefined if using Netlist	Optional	<pre><Lead Id="1" Type="S" Param_order="Freq,S 11" Format="MA" Reference_impedance="50ohm"> ... </Lead></pre>

Keyword	Parent	Description	Presence	Example
Use	Lead element	Specifies one of the parameters defined in Param_order, except the frequency term, that is to be specifically used. See 7.8.2.9. Undefined if using Netlist	Optional	<pre> <Lead Id="1" Type="S" Param_order="Freq,S 21" Format="MA" Meas_type="1" Use="S21"> ... </Lead> </pre>
Netlist	Lead element	Specifies the electrical connectivity of the PDN elements in circuit form using SPICE like data statements.	Required if Type="Ckt"	<pre> <Lead Id="1" Type="Ckt"> ... <Netlist> ... </Netlist> ... </Lead> </pre>
Unit_freq	Lead element	Specifies the units of the frequencies used for specifying the PDN. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units. Undefined if using Netlist and Data_files in touchstone format	Optional	<pre> <Lead Id="1" Type="S"> <Unit_freq> MHz </Unit_freq> ... </Lead> </pre>
Unit_param	Lead element	Specifies the units of the parameters used for specifying the PDN. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units. Undefined if using Netlist and Data_files in touchstone format	Optional	<pre> <Lead Id="1" Type="S"> <Unit_param> dB </Unit_param> ... </Lead> </pre>
Power_level	Lead element	Specifies the power level used for obtaining the PDN data. The value shall conform to A.2.5.5 and should carry both the value and units together. If no units are found, SI units are assumed. Undefined if using Netlist and Data_files.	Optional	<pre> <Lead Id="1" Type="S"> <Power_level> -10dBm </Power_level> ... </Lead> </pre>
Data_files	Lead element	Specifies the path names of the files containing a list of PDN data corresponding to the Pdn section. The path names are separated by a space character (" ") or a line termination. The file names and paths shall conform to A.2.4.2 and A.2.4.3. Only one Data_files keyword or one List keyword shall be included in the Pdn section.	Required if not List	<pre> <Lead Id="1" Type="S"> ... <Data_files> Pdn_pin1_S11.txt </Data_files> ... </Lead> </pre>
List	Lead element	Specifies a list of PDN data entries in the model.	Required if not Data_files	<pre> <Lead Id="1" Type="S"> ... <List> ... </List> ... </Lead> </pre>

B.8.2 Netlist section keywords

The following keywords may be placed in the *Netlist* section under the *Pdn* tag as the parent element:

Keyword	Parent	Description	Presence	Example
Kind	Netlist element	Specifies the kind of netlist used for defining the PDN. See 7.8.3.4 for more information.	Optional	<pre><Netlist Kind="SPICE3"> R1 1 10 1e3 ... </Netlist></pre>
Data_files	Netlist element	Specifies the path names of the files containing a netlist. The path names are separated by a space character (" ") or a line termination. The file names and paths shall conform to A.2.4.2 and A.2.4.3. Only one Data_files keyword or one List keyword shall be included in the Netlist section.	Optional	<pre>< Netlist Kind="SPICE3"> ... <Data_files> PDN_Pin1.net </Data_files> ... </Netlist></pre>

B.9 Ibc section keywords

B.9.1 Lead element keywords

The IBC data of each tested lead defined within the *Lead* tag, similar to the PDN. The *Lead* keyword in the *Ibc* section has the following valid attributes:

Keyword	Parent	Description	Presence	Example
Id	Lead element	Specifies the identity or number of the Lead element. One or many Ids are allowed for differentiating between single, differential or multi-ended pins. For differential or multi-ended pins, Ids shall be separated by a "," character. See 7.8.2.1.	Required	<pre><Lead Id="1,2"...> ... </Lead> <Lead Id="1,2,3"...> ... </Lead></pre>
Blockname	Lead element	Specifies the name of the IBC block. See 7.8.2.3.	Optional	<pre><Blockname> Block1_IBC </Blockname></pre>
Type	Lead element	Specifies the type of IBC data: "S" for S-parameters, "Z" for Z-parameters, "Y" for Y-parameters, "Ckt" for circuit definition. See 7.8.2.4.	Optional	<pre><Lead Id="1,2" Type="Z"> ... </Lead></pre>
Param_order	Lead element	Specifies the order in which the IBC parameters are defined. See 7.8.2.5 for detailed information. Undefined if using Data_files in touchstone format and Netlist.	Optional	<pre><Lead Id="1,2" Type="Z" Param_order="Freq, Z11"> ... </Lead></pre>
Format	Lead element	Specifies the IBC data format. See 7.8.2.6. Undefined if using Data_files in touchstone format and Netlist.	Optional	<pre><Lead Id="1,2" Type="Z" Param_order="Freq, Z11" Format="MA"> ... </Lead></pre>

Keyword	Parent	Description	Presence	Example
Meas_type	Lead element	Specifies the method implemented for performing IBC measurements. See 7.8.2.7. Undefined if using Netlist.	Optional	<pre> <Lead Id="1,2" Type="S" Param_order="Freq, S21" Format="MA" Meas_type="1"> ... </Lead> </pre>
Reference_impedance	Lead element	Specifies the reference impedance used in performing PDN measurements. Undefined if using Netlist. See 7.8.2.8.	Optional	<pre> <Lead Id="1,2" Type="S" Param_order="Freq, S11" Format="MA" Reference_impedance="50ohm"> ... </Lead> </pre>
Use	Lead element	Specifies one of the parameters defined in Param_order, except the frequency term that is to be specifically used. See 7.8.2.9. Undefined if using Netlist	Optional	<pre> <Lead Id="1,2" Type="S" Param_order="Freq, S21" Format="MA" Meas_type="1" Use="S21"> ... </Lead> </pre>
Netlist	Lead element	Specifies the electrical connectivity of the IBC elements in circuit form using SPICE like data statements. See 7.8.2.10.	Required if Type="Ckt"	<pre> <Lead Id="1,2" Type="Ckt"> ... <Netlist> </Netlist> ... </Lead> </pre>
Unit_freq	Lead element	Specifies the units of the frequencies used for specifying the IBC. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units. Undefined if using Netlist and Data_files in touchstone format	Optional	<pre> <Lead Id="1,2" Type="S"> <Unit_freq> MHz </Unit_freq> ... </Lead> </pre>
Unit_param	Lead element	Specifies the units of the parameters used for specifying the IBC. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units. Undefined if using Netlist and Data_files in touchstone format	Optional	<pre> <Lead Id="1,2" Type="S"> <Unit_param> dB </Unit_param> ... </Lead> </pre>
Power_level	Lead element	Specifies the power level used for obtaining the IBC data. The value shall conform to A.2.5.5 and should carry both the value and units together. If no units are found, SI units are assumed. Undefined if using Netlist and Data_files.	Optional	<pre> <Lead Id="1,2" Type="S"> <Power_level> -10dBm </Power_level> ... </Lead> </pre>

Keyword	Parent	Description	Presence	Example
Data_files	Lead element	Specifies the path names of the files containing a list of IBC data corresponding to the <i>lbc</i> section. The path names are separated by a space character (" ") or a line termination. The file names and paths shall conform to A.2.4.2 and A.2.4.3. Only one Data_files keyword or one List keyword shall be included in the <i>lbc</i> section.	Required if not List	<pre><Lead Id="1,2" Type="S"> ... <Data_files> lbc_pin1_S11.txt </Data_files> ... </Lead></pre>
List	Lead element	Specifies a list of IBC data entries in the model.	Required if not Data_files	<pre><Lead Id="1,2" Type="S"> ... <List> ... </List> ... </Lead></pre>

B.9.2 Netlist section keywords

The following keywords may be placed in the *Netlist* section under the *lbc* tag as the parent element:

Keyword	Parent	Description	Presence	Example
Kind	Netlist element	Specifies the kind of netlist used for defining the IBC. See 7.8.3.4 for more information.	Optional	<pre><Netlist Kind="SPICE3"> R1 1 10 1e3 ... </Netlist></pre>
Data_files	Netlist element	Specifies the path names of the files containing a netlist. The path names are separated by a space character (" ") or a line termination. The file names and paths shall conform to A.2.4.2 and A.2.4.3. Only one Data_files keyword or one List keyword shall be included in the Netlist section.	Optional	<pre>< Netlist Kind="SPICE3"> ... <Data_files> IBC_IT1_IT0.net </Data_files> ... </Netlist></pre>

B.10 Ia section keywords

B.10.1 Lead element keywords

The IA data of each tested lead defined within the *Lead* tag. The *Lead* keyword in the *Ia* section has the following valid attributes:

Keyword	Parent	Description	Presence	Example
Id	Lead element	Specifies the identity or number of the Lead element. Only two Ids are permitted per Lead definition since the IA source is always connected between two nodes. Ids shall be separated by a "," character	Required	<pre><Lead Id="1,2"...> ... </Lead> <Lead Id="1,2,3"...> ... </Lead></pre>
Type	Lead element	Specifies the type of IA data: "freq" for frequency domain data and "time" for time domain data.	Required	<pre><Lead Id="1,2" Type="freq"> ... </Lead></pre>
Blockname	Lead element	Specifies the name of the sub-block when using block based IA modelling	Optional	<pre><Lead Id="1,2" Type="freq" > <Blockname> Block1 </Blockname> ... </Lead></pre>
Voltage	Lead element	Specifies the IA data as a voltage quantity. See B.10.2	Required if not Current	<pre><Lead Id="1,2" Type="freq" > <Voltage...> ... </Voltage> ... </Lead></pre>
Current	Lead element	Specifies the IA data as a current quantity. See B.10.3	Required if not Voltage	<pre><Lead Id="1,2" Type="freq" > <Current...> ... </Current> ... </Lead></pre>

B.10.2 Voltage section keywords

The following keywords may be placed in the *Voltage* section under the *Lead* tag as the parent element (within the *Ia* section):

Keyword	Parent	Description	Presence	Example
Param_order	Voltage element	Specifies the order in which the IA voltage quantity is defined. See 7.10.2.5 for detailed information.	Optional	<Voltage Param_order="Freq, voltage"> ... </Voltage>
Format	Voltage element	Specifies the data format for the voltage parameter. See 7.10.2.8.	Optional	<Voltage Param_order="Freq, voltage" Format="MAG"> ... </Voltage>
Unit_freq	Voltage element	Specifies the units of the frequencies used for specifying the IA voltage quantity. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units.	Optional	<Voltage Param_order="Freq, voltage" Format="MAG"> <Unit_freq> kHz </Unit_freq> ... </Voltage>
Unit_time	Voltage element	Specifies the units of the time quantities used for specifying the IA voltage quantity. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units.	Optional	<Voltage Param_order="Freq, voltage" Format="MAG"> <Unit_time> ms </Unit_time> ... </Voltage>
Unit_voltage	Voltage element	Specifies the voltage units of IA data. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units.	Optional	<Voltage Param_order="Freq, voltage" Format="MAG"> <Unit_voltage> dB </Unit_voltage> ... </Voltage>
Data_files	Voltage element	Specifies the path names of the files containing a list of IA data as a voltage quantity. The path names are separated by a space character (" ") or a line termination. The file names and paths shall conform to A.2.4.2 and A.2.4.3. Only one Data_files keyword or one List keyword shall be included in the Voltage section.	Required if not List	<Voltage Param_order="Freq, voltage" Format="MA"> ... <Data_files> la_voltage_pin1_2.d at </Data_files> ... </Voltage>

Keyword	Parent	Description	Presence	Example
List	Voltage element	Specifies a list of IA voltage data entries in the model.	Required if not Data_files or Pulse	<pre><Voltage Param_order="Freq, voltage" Format="MA"> ... <List> ... </List> ... </Voltage></pre>
Pulse	Voltage element	Specifies a simulated waveform for defining the IA voltage data. Allowed only for time-domain IA data.	Required if not Data_files or List	<pre><Voltage> ... <Pulse> ... </Pulse> </Voltage></pre>

B.10.3 Current section keywords

The following keywords may be placed in the *Current* section under the *Lead* tag as the parent element (within the *Ia* section):

Keyword	Parent	Description	Presence	Example
Param_order	Current element	Specifies the order in which the IA current quantity is defined. See 7.10.2.5 for detailed information.	Optional	<pre><Current Param_order="Freq, voltage"> ... </Current></pre>
Format	Current element	Specifies the data format for the current parameter. See 7.10.2.8.	Optional	<pre><Current Param_order="Freq, Current" Format="MAG"> ... </Current></pre>
Unit_freq	Current element	Specifies the units of the frequencies used for specifying the IA current quantity. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units.	Optional	<pre><Current Param_order="Freq, Current" Format="MAG"> <Unit_freq> kHz </Unit_freq> ... </Current></pre>
Unit_time	Current element	Specifies the units of the time quantities used for specifying the IA current quantity. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units.	Optional	<pre><Current Param_order="Freq, Current" Format="MAG"> <Unit_time> us </Unit_time> ... </Current></pre>

Keyword	Parent	Description	Presence	Example
Unit_current	Current element	Specifies the current units of IA data. The value shall conform to A.2.5.5. If this keyword is omitted, the units are assumed to be SI units.	Optional	<pre> <Current Param_order="Freq, Current" Format="MAG"> <Unit_current> dB </Unit_current> ... </Current> </pre>
Data_files	Current element	Specifies the path names of the files containing a list of IA data as a current quantity. The path names are separated by a space character (" ") or a line termination. The file names and paths shall conform to A.2.4.2 and A.2.4.3. Only one Data_files keyword or one List keyword shall be included in the Current section.	Required if not List or Pulse	<pre> <Current Param_order="Freq, Current" Format="MA"> ... <Data_files> Ia_current_pin1_2.d at </Data_files> ... </Current> </pre>
List	Current element	Specifies a list of IA current data entries in the model.	Required if not Data_files or Pulse	<pre> <Current Param_order="Freq, Current" Format="MA"> ... <List> ... </List> ... </Current> </pre>
Pulse	Current element	Specifies a simulated waveform for defining the IA current data. Allowed only for time-domain IA data.	Required if not Data_files or List	<pre> <Current> ... <Pulse> ... </Pulse> </Current> </pre>

Annex C (informative)

Example of ICEM-CE macro-model in CEML format

C.1 General

An example ICEM-CE macro-model of a microcontroller is illustrated. The IC package and the modelled pins are shown in Figure C.1.

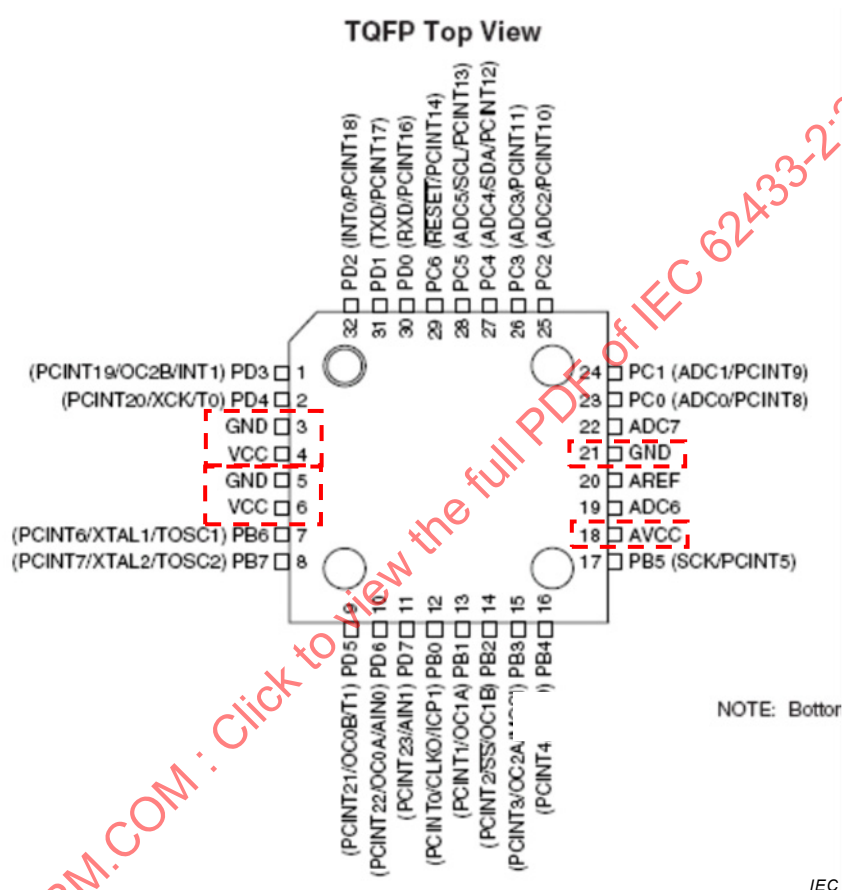


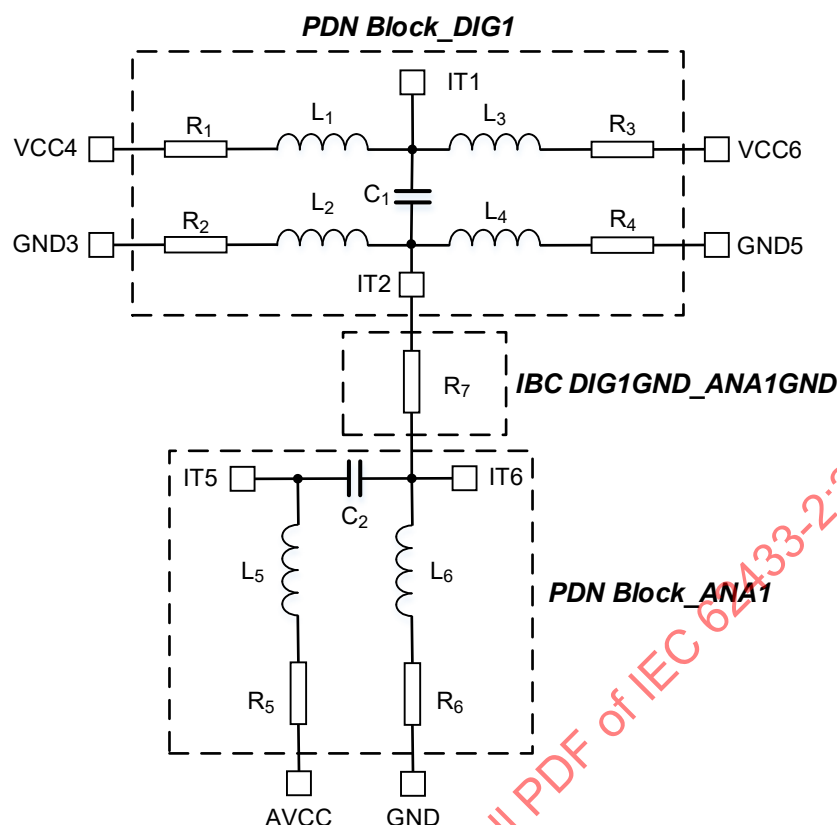
Figure C.1 – Example pin-out of a microcontroller and the modelled pins

The ICEM-CE macro-model of the following parts of the IC is built:

- Digital supply 1: VCC (pin 4) and GND (pin 3)
- Digital supply 2: VCC (pin 5) and GND (pin 5)
- Analogue supply 1: AVCC (pin 18) and GND (pin 21)

C.2 PDN and IBC sub-model

The PDN or IBC structure is shown in Figure C.2.



IEC

Key

VCC4, VCC6	IC pin interface for the digital supply at package level (external terminals)
GND3, GND5	IC pin interface for the digital ground at the package level (external terminals)
R_1, R_2, R_3, R_4	Parasitic resistance of package on the digital supply lines
L_1, L_2, L_3, L_4	Parasitic inductance of package on the digital supply lines
C_1	Parasitic capacitance of die (digital section)
AVCC	IC pin interface for the analogue supply at package level
GND	IC pin interface for the analogue ground at the package level
R_5, R_6	Parasitic resistance of package on the analogue supply lines
L_5, L_6	Parasitic inductance of package on the analogue supply lines
C_2	Parasitic capacitance of die (analogue section)
R_7	Parasitic resistance between the analogue and digital grounds (IBC)

Figure C.2 – PDN sub-model topology

For representative purposes, the PDN is separated into two blocks: Block_DIG1 and Block_ANA1. An IBC block (IBC_DIG1GND_ANA1GND) is used to model the ground isolation between the digital and analogue modules.

The PDN and IBC sub-models are extracted using VNA *S*-parameter measurements.

C.3 IA sub-model

The IA structure is shown in Figure C.3.

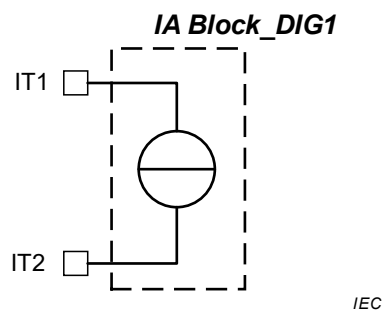


Figure C.3 – IA sub-model topology

The IA sub-model is modelled only for the digital block: Block_DIG1. The IA data is obtained through measurements. The frequency-domain and time-domain representation of the IA data of the digital block is shown in Figure C.4 and Figure C.5 respectively.

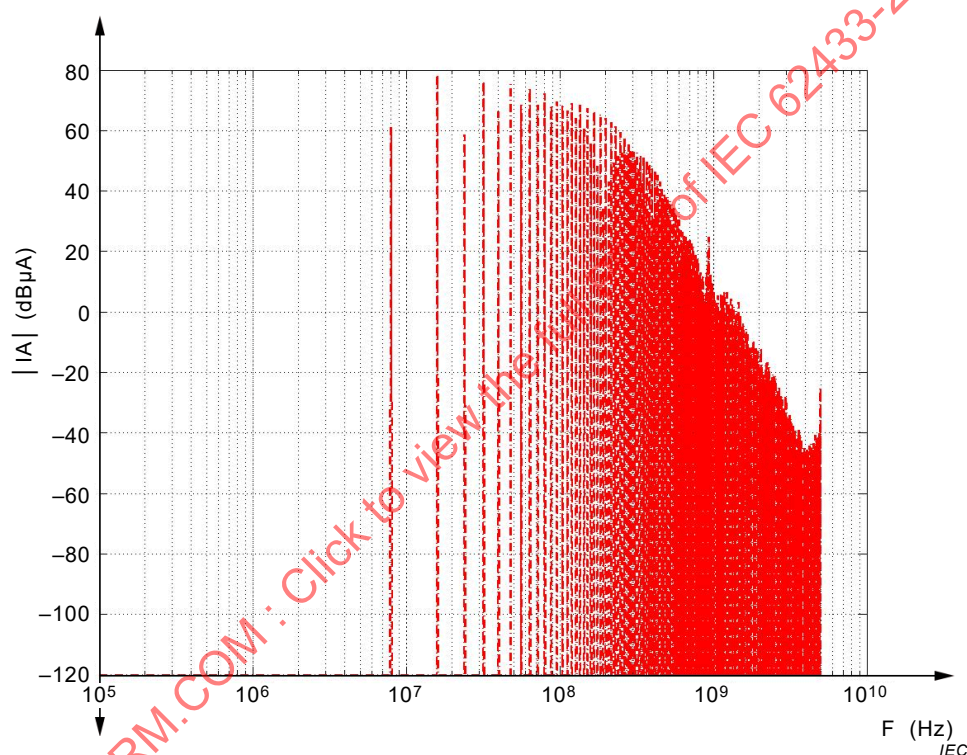


Figure C.4 – IA of digital block in frequency domain

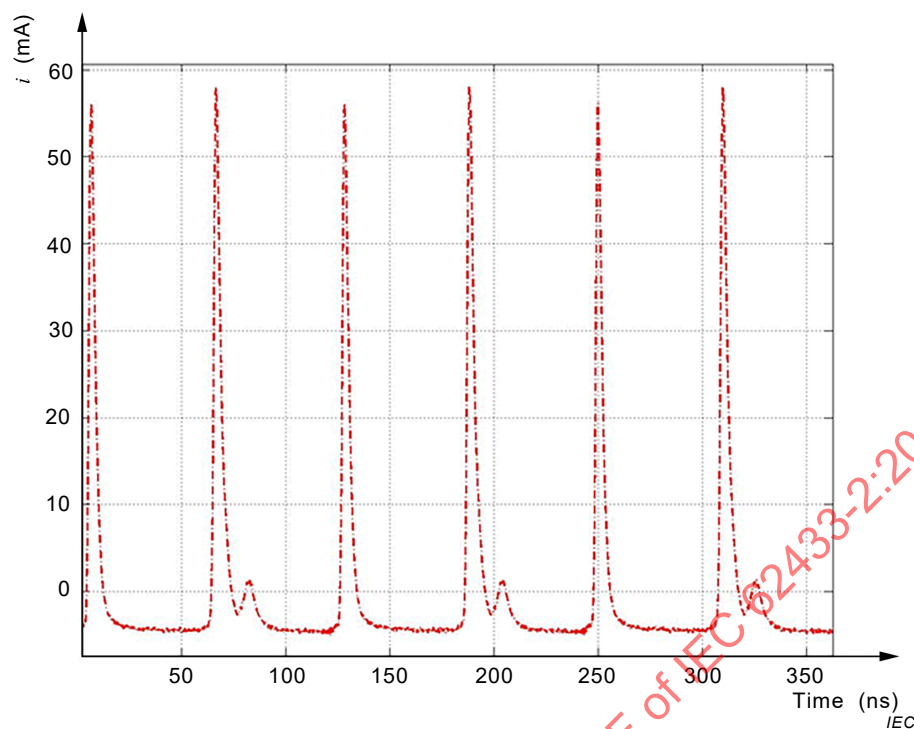


Figure C.5 – IA of digital block in time domain

C.4 Frequency domain ICEM-CE in CEML

With the PDN, IBC and IA sub-models obtained in C.2 and C.3, the frequency domain ICEM-CE of the microcontroller is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- root element -->
<CEmodel>
  <!-- Header section -->
  <Header>
    <Cem_ver>1.0</Cem_ver>
    <Filename>ICEM_Micro88_FD.ceml</Filename>
    <File_ver>1.0</File_ver>
    <Author>AR,JLL</Author>
    <Dut>MicroAt88</Dut>
    <Date>July 17, 2014</Date>
    <Meas_method>S for PDN</Meas_method>
  </Header>
  <!-- Lead definitions section -->
  <Lead_definitions>
    <Lead Id="1" Name="VCC4" Mode="None" Type="external"/>
    <Lead Id="2" Name="GND3" Mode="GND" Type="external"/>
    <Lead Id="3" Name="VCC6" Mode="None" Type="external"/>
    <Lead Id="4" Name="GND5" Mode="GND" Type="external"/>
    <Lead Id="5" Name="AVCC" Mode="None" Type="external"/>
    <Lead Id="6" Name="GND" Mode="GND" Type="external"/>
    <Lead Id="7" Name="IT1" Mode="None" Type="internal"/>
    <Lead Id="8" Name="IT2" Mode="None" Type="internal"/>
    <Lead Id="9" Name="IT5" Mode="None" Type="internal"/>
    <Lead Id="10" Name="IT6" Mode="None" Type="internal"/>
  </Lead_definitions>
  <!-- Validity section -->
  <Validity>
    <Power_supply>5V</Power_supply>
  </Validity>
</CEmodel>
```

```

    <Frequency_range>[100kHz - 5GHz]</Frequency_range>
    <Phase_origin>0degrees</Phase_origin>
    <Temperature_range>25Celsius</Temperature_range>
    <Notes>Loopless program for IA extraction</Notes>
</Validity>
<!-- Pdn section -->
<Pdn>
    <!-- PDN of digital block -->
    <Lead Id="1,2,3,4,7,8" Type="Ckt" >
        <Blockname>PDN_Block_DIG1</Blockname>
        <Netlist>
            L1 1 tmp1 2.2n
            R1 tmp1 7 0.18
            L3 3 tmp3 2.2n
            R3 tmp3 7 0.18
            L2 2 tmp2 2.2n
            R2 tmp2 8 0.175
            L4 4 tmp4 2.2n
            R4 tmp4 8 0.175
            k1 L1 L2 0.5
            k2 L3 L4 0.5
            C1 7 8 1.32n
        </Netlist>
    </Lead>
    <!-- PDN of analog block -->
    <Lead Id="5,6,9,10" Type="Ckt">
        <Blockname>PDN_Block_ANA1</Blockname>
        <Netlist>
            L5 5 tmp5 2.64n
            R5 tmp5 9 2.56
            L6 6 tmp6 2.2n
            R6 tmp6 10 0.335
            k2 L5 L6 0.091
            C2 9 10 531p
        </Netlist>
    </Lead>
</Pdn>
<!-- Ibc section -->
<Ibc>
    <!-- IBC of isolation between digital and analog block -->
    <Lead Id="8,10" Type="Ckt" >
        <Blockname>IBC_DIG1GND_ANA1GND</Blockname>
        <Netlist>
            R7 8 10 0.376
        </Netlist>
    </Lead>
</Ibc>
<!-- Ia section -->
<Ia>
    <!-- IA of digital block -->
    <Lead Id="7,8" Type="Freq" >
        <Blockname>IA_Block_DIG1</Blockname>
        <Current Param_order="Freq,Current" Format="DB">
            <Unit_freq>kHz</Unit_freq>
            <Unit_current>dBuA</Unit_current>
            <Data_files>
                IA_Dig_At88.txt
            </Data_files>
        </Current>
    </Lead>
</Ia>
</CEmodel>

```

C.5 Time domain ICEM-CE in CEML

With the PDN, IBC and IA sub-models obtained in C.2 and C.3, the time domain ICEM-CE of the microcontroller is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- root element -->
<CEmodel>
  <!-- Header section -->
  <Header>
    <Cem_ver>1.0</Cem_ver>
    <Filename>ICEM_Micro88_TD.ceml</Filename>
    <File_ver>1.0</File_ver>
    <Author>AR, JLL</Author>
    <Dut>MicroAt88</Dut>
    <Date>July 17, 2014</Date>
    <Meas_method>S for PDN</Meas_method>
  </Header>
  <!-- Lead definitions section -->
  <Lead_definitions>
    <Lead Id="1" Name="VCC4" Mode="None" Type="external"/>
    <Lead Id="2" Name="GND3" Mode="GND" Type="external"/>
    <Lead Id="3" Name="VCC6" Mode="None" Type="external"/>
    <Lead Id="4" Name="GND5" Mode="GND" Type="external"/>
    <Lead Id="5" Name="AVCC" Mode="None" Type="external"/>
    <Lead Id="6" Name="GND" Mode="GND" Type="external"/>
    <Lead Id="7" Name="IT1" Mode="None" Type="internal"/>
    <Lead Id="8" Name="IT2" Mode="None" Type="internal"/>
    <Lead Id="9" Name="IT5" Mode="None" Type="internal"/>
    <Lead Id="10" Name="IT6" Mode="None" Type="internal"/>
  </Lead_definitions>
  <!-- Validity section -->
  <Validity>
    <Power_supply>5V</Power_supply>
    <Time_range>[0 - 10us]</Time_range>
    <Temperature_range>25Celsius</Temperature_range>
    <Time_origin>0us</Time_origin>
    <Notes>Loopless program for IA extraction</Notes>
  </Validity>
  <!-- Pdn section -->
  <Pdn>
    <!-- PDN of digital block -->
    <Lead Id="1,2,3,4,7,8" Type="Ckt" >
      <Block_name>PDN_Block_DIG1</Blockname>
      <Netlist>
        L1 1 tmp1 2.2n
        R1 tmp1 7 0.18
        L3 3 tmp3 2.2n
        R3 tmp3 7 0.18
        L2 2 tmp2 2.2n
        R2 tmp2 8 0.175
        L4 4 tmp4 2.2n
        R4 tmp4 8 0.175
        k1 L1 L2 0.5
        k2 L3 L4 0.5
        C1 7 8 1.32n
      </Netlist>
    </Lead>
    <!-- PDN of analog block -->
    <Lead Id="5,6,9,10" Type="Ckt">
      <Block_name>PDN_Block_ANA1</Blockname>
      <Netlist>
```

```

        L5 5 tmp5 2.64n
        R5 tmp5 9 2.56
        L6 6 tmp6 2.2n
        R6 tmp6 10 0.335
        k2 L5 L6 0.091
        C2 9 10 531p
    </Netlist>
</Lead>
</Pdn>
<!-- Ibc section -->
<Ibc>
    <!-- IBC of isolation between digital and analog block -->
    <Lead Id="8,10" Type="Ckt">
        <Block_name>IBC_DIG1GND_ANA1GND</Blockname>
        <Netlist>
            R7 8 10 0.376
        </Netlist>
    </Lead>
</Ibc>
<!-- Ia section -->
<Ia>
    <!-- IA of digital block -->
    <Lead Id="7 8" Type="Time">
        <Block_name>IA_Block_DIG1</Blockname>
        <Current Param_order="Time,Current" Format="MAG">
            <Unit_time>s</Unit_time>
            <Unit_current>A</Unit_current>
            <Data_files>
                IA_Dig_At88.txt
            </Data_files>
        </Current>
    </Lead>
</Ia>
</CEmodel>

```

IECNORM.COM : Click to view the full PDF of IEC 62433-2:2017

Annex D (informative)

Conversions between parameter types

D.1 General

When the PDN is represented using network parameters (S - or Z - or Y - parameters), conversion between them may be needed for implementation purposes. This conversion depends on whether the PDN is single-ended or *multi*-ended. As an illustration, the conversion formulae for one-port and two-port PDN are shown in D.2 and D.3 respectively.

D.2 Conversion for one-port PDN

Table D.1 shows the relationships to convert between S_{11} , Z_{11} and Y_{11} for a one-port network.

Table D.1 – One-port conversion

		To		
		S -parameters	Z -parameters	Y -parameters
From	S -parameters		$Z_{11} = Z_0 \cdot \left(\frac{1 + S_{11}}{1 - S_{11}} \right)$	$Y_{11} = Y_0 \cdot \left(\frac{1 - S_{11}}{1 + S_{11}} \right)$
	Z -parameters	$S_{11} = \frac{Z_{11} - Z_0}{Z_{11} + Z_0}$		$Y_{11} = \frac{1}{Z_{11}}$
	Y -parameters	$S_{11} = \frac{Y_0 - Y_{11}}{Y_0 + Y_{11}}$	$Z_{11} = \frac{1}{Y_{11}}$	
Z_0 is the characteristic impedance and Y_0 is the characteristic admittance used as the reference for the S -parameters (Z_0 is typically 50 Ω).				

D.3 Conversion for two-port PDN

Table D.2 shows the relationships between S -, Z - and Y - parameters for a two-port network [5].

Table D.2 – Two-port conversion

		To		
		S-parameters	Z-parameters	Y-parameters
From	S-parameters		$Z_{11} = Z_0 \frac{(1 + S_{11})(1 - S_{22}) + S_{12} S_{21}}{(1 - S_{11})(1 - S_{22}) - S_{12} S_{21}}$ $Z_{12} = Z_0 \frac{2 S_{12}}{(1 - S_{11})(1 - S_{22}) - S_{12} S_{21}}$ $Z_{21} = Z_0 \frac{2 S_{21}}{(1 - S_{11})(1 - S_{22}) - S_{12} S_{21}}$ $Z_{22} = Z_0 \frac{(1 - S_{11})(1 + S_{22}) + S_{12} S_{21}}{(1 - S_{11})(1 - S_{22}) - S_{12} S_{21}}$	$Y_{11} = Y_0 \frac{(1 - S_{11})(1 + S_{22}) + S_{12} S_{21}}{(1 + S_{11})(1 + S_{22}) - S_{12} S_{21}}$ $Y_{12} = Y_0 \frac{-2 S_{12}}{(1 + S_{11})(1 + S_{22}) - S_{12} S_{21}}$ $Y_{21} = Y_0 \frac{-2 S_{21}}{(1 + S_{11})(1 + S_{22}) - S_{12} S_{21}}$ $Y_{22} = Y_0 \frac{(1 + S_{11})(1 - S_{22}) + S_{12} S_{21}}{(1 + S_{11})(1 + S_{22}) - S_{12} S_{21}}$
	Z-parameters	$S_{11} = \frac{(Z_{11} - Z_0)(Z_{22} + Z_0) - Z_{12} Z_{21}}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12} Z_{21}}$ $S_{12} = \frac{2 Z_0 Z_{12}}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12} Z_{21}}$ $S_{21} = \frac{2 Z_0 Z_{21}}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12} Z_{21}}$ $S_{22} = \frac{(Z_{11} + Z_0)(Z_{22} - Z_0) - Z_{12} Z_{21}}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12} Z_{21}}$		$Y_{11} = \frac{Z_{22}}{Z_{11} Z_{22} - Z_{12} Z_{21}}$ $Y_{12} = \frac{-Z_{12}}{Z_{11} Z_{22} - Z_{12} Z_{21}}$ $Y_{21} = \frac{-Z_{11}}{Z_{11} Z_{22} - Z_{12} Z_{21}}$ $Y_{22} = \frac{Z_{11}}{Z_{11} Z_{22} - Z_{12} Z_{21}}$
	Y-parameters	$S_{11} = \frac{(Y_0 - Y_{11})(Y_0 + Y_{22}) + Y_{12} Y_{21}}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y_{12} Y_{21}}$ $S_{12} = \frac{-2 Y_0 Y_{12}}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y_{12} Y_{21}}$ $S_{21} = \frac{-2 Y_0 Y_{21}}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y_{12} Y_{21}}$ $S_{22} = \frac{(Y_0 + Y_{11})(Y_0 - Y_{22}) + Y_{12} Y_{21}}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y_{12} Y_{21}}$	$Z_{11} = \frac{Y_{22}}{Y_{11} Y_{22} - Y_{12} Y_{21}}$ $Z_{12} = \frac{-Y_{12}}{Y_{11} Y_{22} - Y_{12} Y_{21}}$ $Z_{21} = \frac{-Y_{21}}{Y_{11} Y_{22} - Y_{12} Y_{21}}$ $Z_{22} = \frac{Y_{11}}{Y_{11} Y_{22} - Y_{12} Y_{21}}$	

Z_0 is the characteristic impedance and Y_0 is the characteristic admittance used as the reference for the S-parameters (Z_0 is typically 50 Ω).

Annex E (informative)

Model parameter generation

E.1 General

The purpose of Annex E is to explain the methodologies used to extract the components. Three different ways are possible:

- Default parameters can be used when no other data is available.
- Parameters derived from parasitic element extractor tools, which can be used at the design phase of the IC and/or 3D electromagnetic field simulator.
- Parameters derived from measurements when the IC is already available.

E.2 Default structure and values

E.2.1 General

The PDN and IA components can be obtained using technological data coming from the IC and the packaging suppliers. The accuracy of default values is relatively low compared to values obtained by measurements or design information.

E.2.2 IA parameters

The structure of the IA component is shown in Figure 2. It is possible to quickly determine the IA component using the technological data, which can be obtained from IC suppliers. Table E.1 shows typical values of the parameters.

As an example, for 32 nm technology, cell density is around 3 500 k/mm². The probable number of cells in 0,5×0,5 mm² area is approximately 3 500 k × 0,25 = 875 k gates. Considering that in average 10 % of cells are switching simultaneously, the probable CPU current at each clock edge is 875 k gates × 10 % × 0,04 mA = 3 500 mA.

Table E.1 – Typical parameters for CMOS logic technologies

Technology	1st year of produc.	External Supply (V)	Internal supply (V)	Typical Frequency (MHz)	Gate density (x1000 /mm ²)	SRAM area (µm ²)	Gate current (mA)	Gate capa (fF)	Typ gate delay (ps)
0,8 µm	1990	5	5	1-10	15	80,0	0,9	40	180
0,5 µm	1993	5	5	2-50	28	40,0	0,75	30	130
0,35 µm	1995	5	3,3	4-100	50	20,0	0,6	25	100
0,25 µm	1997	5	2,5	8-150	90	10,0	0,4	20	75
0,18 µm	1999	3,3	1,8	16-300	160	5,0	0,3	15	50
0,12 µm	2001	2,5	1,2	40-500	240	2,4	0,2	10	35
90 nm	2004	2,5	1,0	100-1 000	480	1,4	0,1	7	25
65 nm	2006	2,5	1,0	150-1 500	900	0,6	0,07	5	22
45 nm	2008	1,8	1,0	150-2 000	2 000	0,35	0,05	3	18
32 nm	2010	1,8	0,9	200-3 000	3 500	0,20	0,04	3	14
28 nm	2012	1,5	0,9	200-4 000	4 800	0,15	0,03	2	10
20 nm	2014	1,2	0,8	200-4 500	8 000	0,10	0,02	1,5	8
14 nm	2016	1,2	0,8	300-5 000	15 000	0,07	0,015	1,0	6
10 nm	2018	1,0	0,6	300-5 000	30 000	0,03	0,011	0,8	4
7 nm	2020	1,0	0,5	300-5 000	50 000	0,02	0,008	0,6	3

Table E.2 gives the default number of logic gates for typical microcontrollers.

Table E.2 – Typical number of logic gates vs. CPU technology

CPU technology	Total number of logic cells	Synchronous switching on clock edge (10 %)
8 bits	3 000 to 10 000	300 to 1 000
16 bits	30 k to 100 k	3 k to 10 k
32 bits	100 k to 1 M	10 k to 100 k
64 bits	1 M to 10 M	100 k to 1 M
64 bit Octa	10 M to 100 M	1 M to 10 M

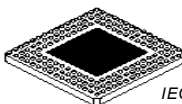
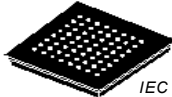
As an example, a 32-bit microcontroller is fabricated in 45 nm technology. The microcontroller has approximately 500 K gates. The probable CPU current at each clock edge is $500k \times 10 \% \times 0,05 \text{ mA} = 2,5 \text{ A}$. The rise and fall time of the peak current is 18 ps.

E.2.3 PDN parameters

The structure of a simple PDN is given in Figure 5.

The technological data given by the package suppliers enable to build quickly a PDN component. The typical values of the parasitic resistance (R), inductance (L) and capacitance (C) are summarized in Table E.3. These values are used for the frequency range from DC to approximately 1 GHz.

Table E.3 – R, L and C parameters for various package types

Package		Pin count	R	L	C
	Dual in Line (DIL)	64 pins	0,025 Ω to 0,075 Ω	2 nH to 15 nH	1 pF to 10 pF
	Shrink dual in line (SDIL)	64 pins	0,025 Ω to 0,1 Ω	1 nH to 10 nH	1 pF to 10 pF
	Small Outline Package (SOP)	64 pins	0,025 Ω to 0,05 Ω	1 nH to 7 nH	1 pF to 7 pF
	Quad Flat Pack (QFP)	400 pins	0,035 Ω to 0,55 Ω	3 nH to 7 nH	2 pF to 5 pF
	Ball Grid Array (BGA)	800 pins	0,05 Ω to 0,15 Ω	0,5 nH to 10 nH	1 pF to 10 pF
	Fine Pitch Ball Grid Array (FBGA)	1 500 pins	0,05 Ω to 0,2 Ω	0,5 nH to 10 nH	1 pF to 20 pF
	Mould Chip Scale Package (MCSP)	1 500 pins	0,025 Ω to 0,1 Ω	0,5 nH to 5 nH	1 pF to 15 pF

E.3 Model parameter generation from design information

E.3.1 General

ICs suppliers can extract parameters from their design information using IC design tools.

E.3.2 IA parameters

The current source is the main component in the ICEM-CE macro-model: it summarizes the contribution of all the logic gates on the current flowing through the power supply pins of the component. The behavioural simulation is a statistical approach that could be done at the early stage of the design flow (no need of the layout for example), and could take into account a very important numbers of gates. The way to build the current source is based on a statistical evaluation of the core elements: choice of the more representative gate (the most frequently-used gate in the design), choice of the typical load of the gate. From this information, a simulation can be done at the circuit level to extract the consumption current of the typical gate. For example, in a particular 16-bit micro-controller, the most popular gate is an inverter. Figure E.1 illustrates the test schematic to define the typical current waveform.

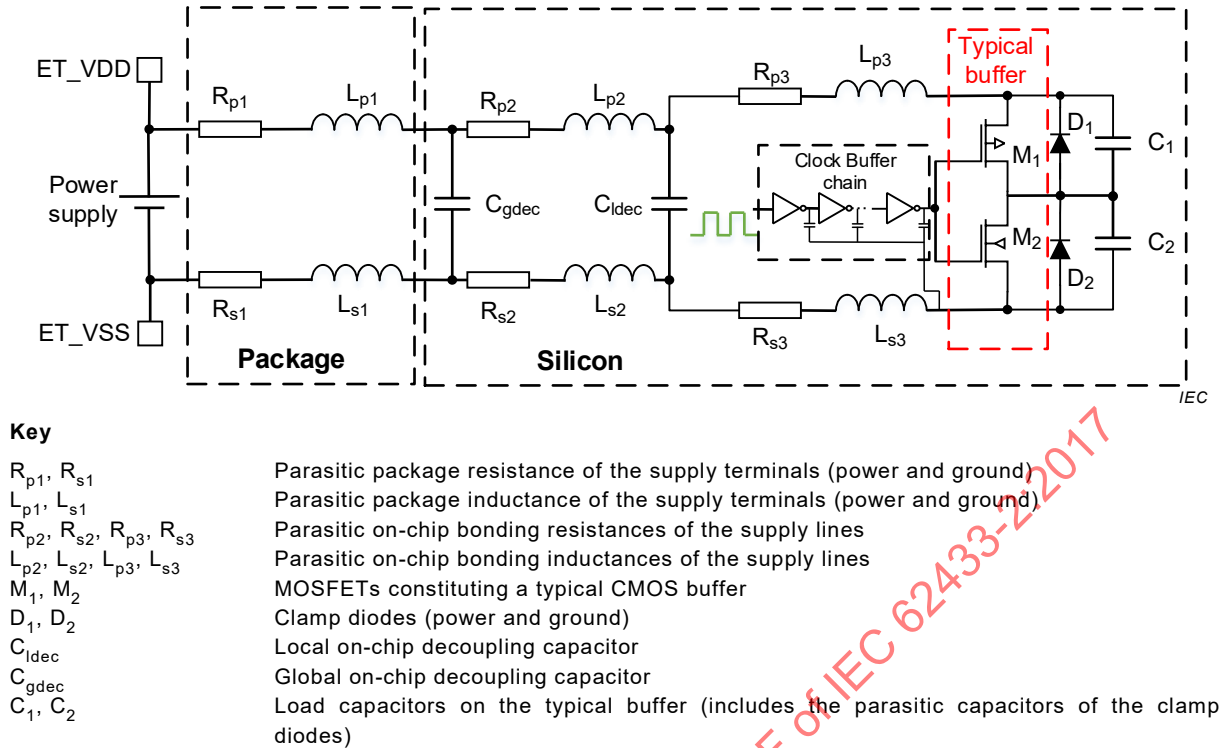


Figure E.1 – Typical characterization current gate schematic

This example shows the description of the PDN, local and distributed on-chip decoupling networks, and the description of the IA. As the inverter has a non-symmetric structure, the current peak is not the same for the switching-on and the switching-off: the average between these two current waveforms is therefore required in the approach given in Figure E.2.

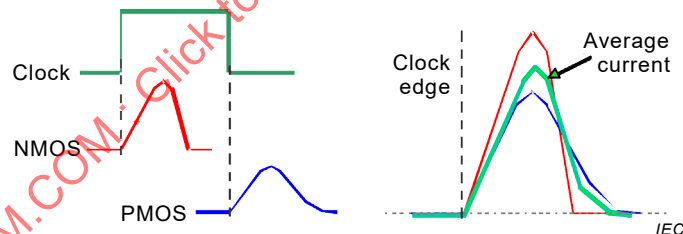


Figure E.2 – Current peak during switching transition

The second part of the methodology consists in multiplying this elementary current waveform by the number of gates that are switching at the same time. Figure E.3 illustrates this step. The aim is to obtain the number of logical nodes that change state in function of time for a given software implemented in the micro-controller. The feasibility of this operation is possible thanks to the Verilog source code, which models the micro-controller. Different types of analysis can be done, such as Best Case Simulation (BCS), Worst Case Simulation (WCS) and simulations at some corner cases of the technology.

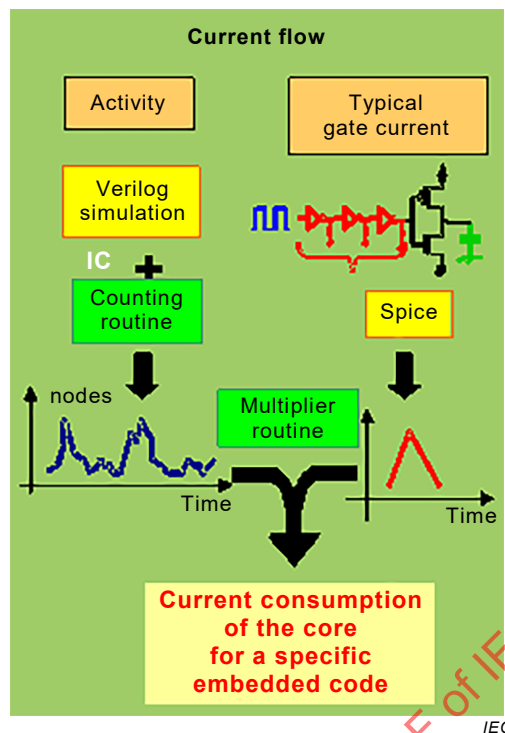


Figure E.3 – Example of IA extraction procedure from design

The comparison of the activity between these two cases is given in Figure E.4. The worst case peak is in fact determined to have the minimum delay in the signal propagation, and it is the opposite for the best case peak. The impact is in fact that the worst case peak presents a better synchronization of the signal and more gates are switching at the same time. The current peak is therefore sharper and the maximum value is higher than for the best case peak. The best case peak has therefore been chosen for the end of this study. In Figure E.4 the “RTL peak” item means Register Transfer Level.

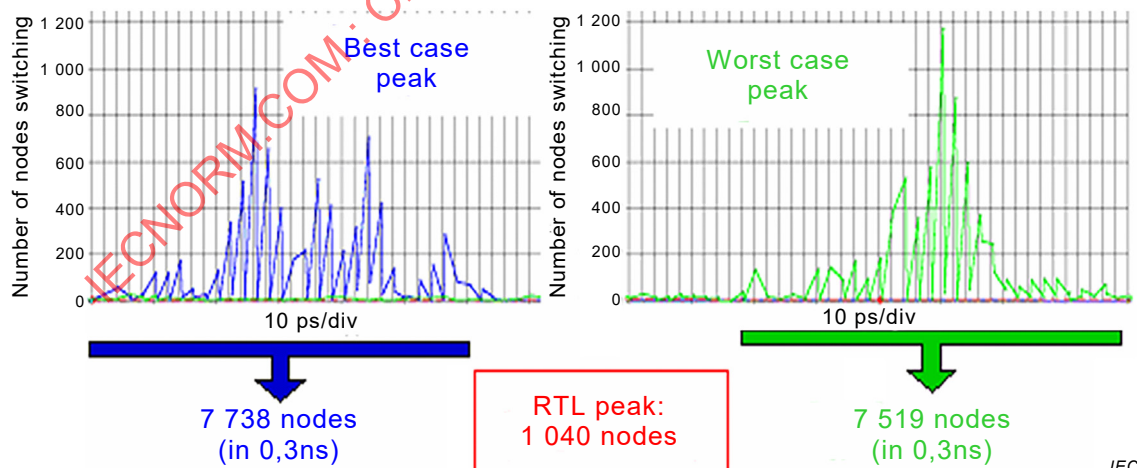


Figure E.4 – Technology Influence

The final waveform corresponding to a specific internal code embedded in the microcontroller is given in Figure E.5.

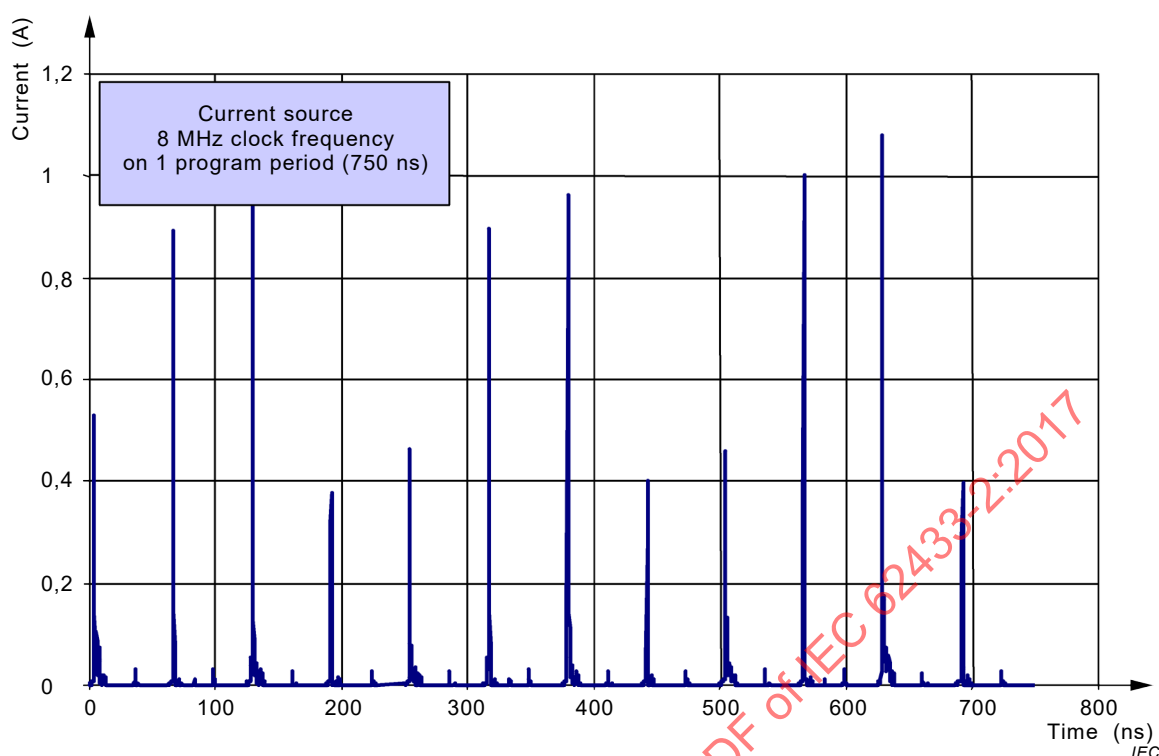


Figure E.5 – Final current waveform for a program period

The validation of this methodology has been done using the IEC 61967-4 measurement (conducted emission) on the power supply as described in Annex G. The current has been measured through the $1\ \Omega$ resistance inserted on the ground path of the IC [6]. The passive part of the ICEM-CE macro-model has been obtained by the parameters by default described in E.2. Indeed, the test PCB has also been modelled by an RLC block using the classical formula for identifying the RLC parameters.

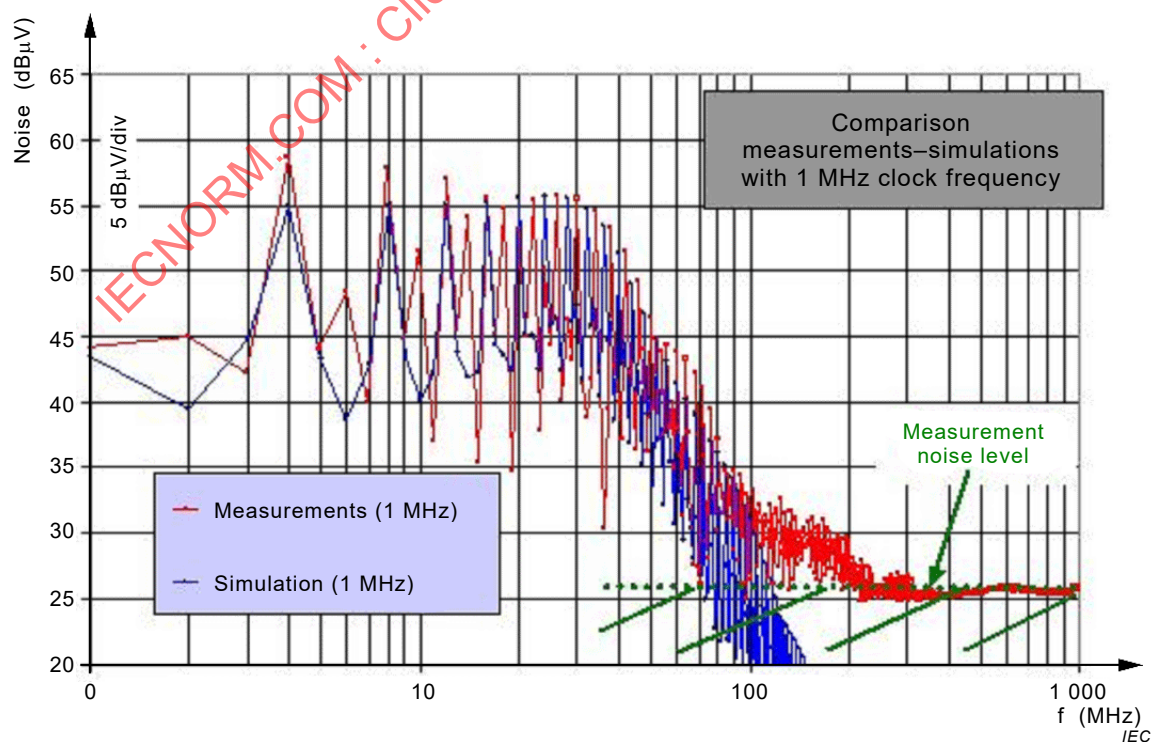
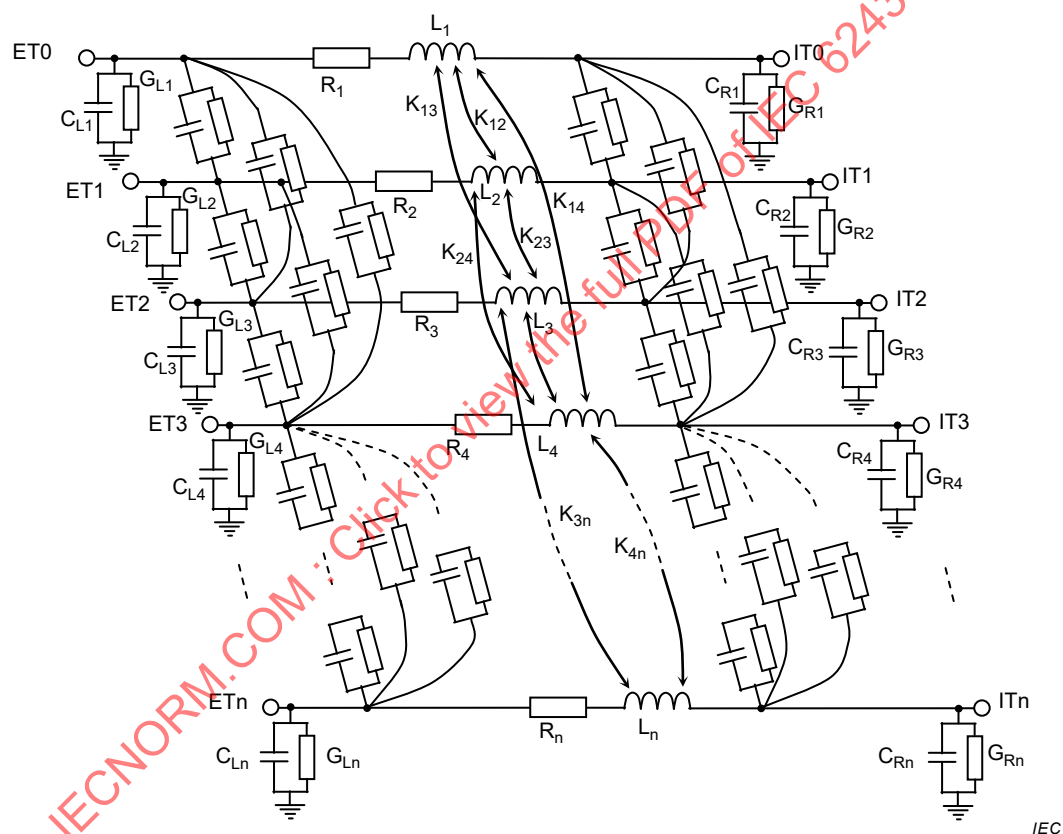


Figure E.6 – Comparison between measurement and simulation

It can be observed in Figure E.6 that it is possible to predict the harmonic current peak up to 100 MHz. More measurements with other 16-bit micro-controllers have been done and the same results were obtained. It can therefore be considered that the simulation methodology is correct but needs improvement to extend the frequency range of validity. As far as this methodology is concerned, many approximations are performed and especially for the modelling of the interconnections, package and PCB. The need to go higher in frequency will force to use RF methods for the passive, bonding and package extraction.

E.3.3 PDN parameters

Package and Bonding wire modelling can be determined by using a 3D EM solver. From this simulation, the S -parameter file of the whole path is obtained and then is reduced to a lumped element model depending of the accuracy needed for the simulation. Another solution is to use the mathematical approach, using solvers, in order to calculate the parameters of the lumped element model directly linked to the geometry of the package. The structure of the model is given in Figure E.7 and could be reduced by one of the order reduction methods depending on the accuracy needed for the simulation.



Key

ET0, ET1, ... ETn	External terminals
IT0, IT1, ... ITn	Internal terminals
$L_1, L_2, \dots L_n$	Parasitic inductors
$R_1, R_2, \dots R_n$	Parasitic resistors
$C_{L1}, C_{L2}, \dots C_{Ln}$	Parasitic capacitors
$C_{R1}, C_{R2}, \dots C_{Rn}$	Parasitic capacitors
$G_{L1}, G_{L2}, \dots G_{Ln}$	Parasitic conductance
$G_{R1}, G_{R2}, \dots G_{Rn}$	Parasitic conductance
$K_{12}, K_{13}, K_{24}, \text{etc.}$	Coupling coefficients between the parasitic inductors

Figure E.7 – Example lumped element model of a package

The information needed to extract package and bonding parameters is listed below:

- Package mechanical drawing
- Die size and cavity size
- Bonding diagram
- Type of bonding (diameter, material)
- Distance to the ground plane

An example of a package netlist is given below.

```
.subckt package ET0 ET1 ET2 ... ETn IT0 IT1 IT2 ... ITn gnd
#Definition of the leadframe part
L1    ET0  n00  1,820e-09
R1    n01  n02  2,408e-01
C1    n00  gnd  1e-12
L2    ET1  n10  1,828e-09
R2    n11  n12  2,434e-01
C2    n10  gnd  1e-12
L3    ET2  n20  1,820e-09
R3    n21  n22  2,408e-01
C3    n20  gnd  1e-12
...
K12   L1    L2    0,2597
K23   L2    L3    0,1089
K34   L3    L4    0,0612
...
C01   n00  n10  0,5e-12
C12   n10  n20  0,5e-12
C02   n00  n20  0,2e-12
...
# Definition of wire bonding part
L1wb   ET1    IT10  1,820e-09
R1wb   IT10   IT1    2,408e-01
L2wb   ET2    IT20  1,828e-09
R2wb   IT20   IT2    2,434e-01
...
K12wb  L1wb   L2wb   0,2597
K23wb  L2wb   L3wb   0,1089
K34wb  L3wb   L4wb   0,0612
...
.ends package
```

Figure E.8 illustrates the previous netlist and shows how the different elements of the package have been modelled. Some other architecture could also be taken into account, especially for the new packaging solutions.

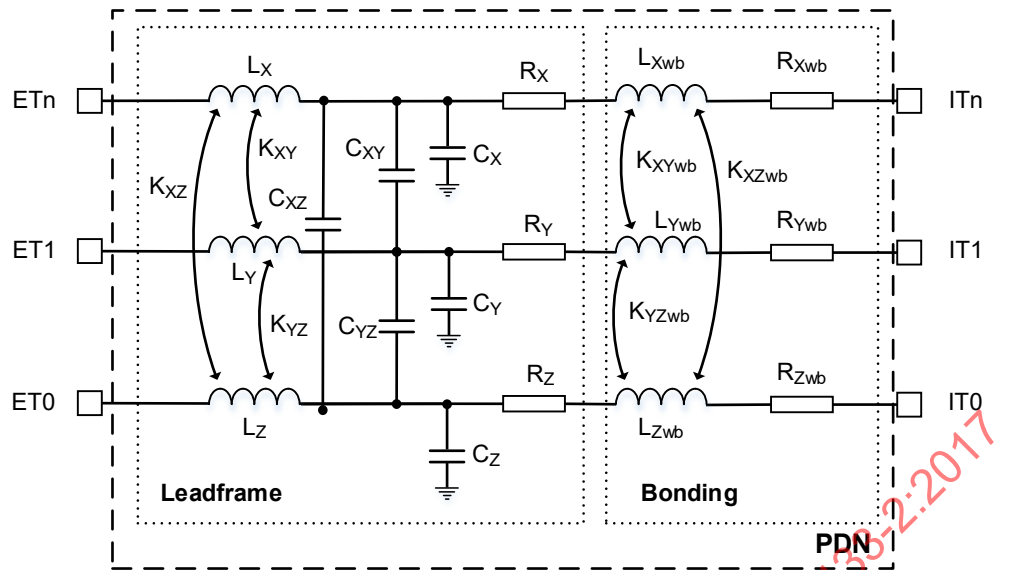


Figure E.8 – Circuit structure of the netlist

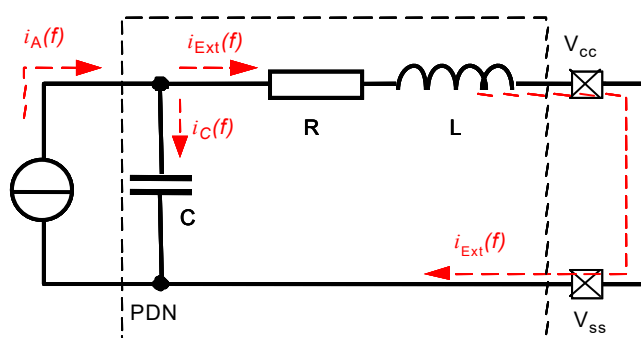
E.4 Model parameter generation from measurements

E.4.1 IA parameters

The IA component is described by a parameter called IA. Most of the time, this parameter is not accessible directly. The transfer function, PDN, has already been extracted and the current flowing externally, $i_{Ext}(t)$ or $i_{Ext}(f)$, has been measured. Depending on which analysis is needed, the IA can be described either in the time or in the frequency domain. It also depends on whether v_{Ext} is measured in the time or the frequency domain.

- Frequency domain

$i_{Ext}(f)$ is described in frequency and $i_A(f)$ is modelled directly by using the formula expressed in Figure E.9. In this figure it is assumed that the power-supply is equivalently shorted in the frequency band of interest.



$$i_A(f) = j\omega C \cdot \left(R + j\omega L + \frac{1}{j\omega C} \right) \cdot i_{Ext}(f) \quad IEC$$

Figure E.9 – Principle of the IA computation in the frequency domain

In case of multiple IAs, a matrix form can be used as shown in the following expression:

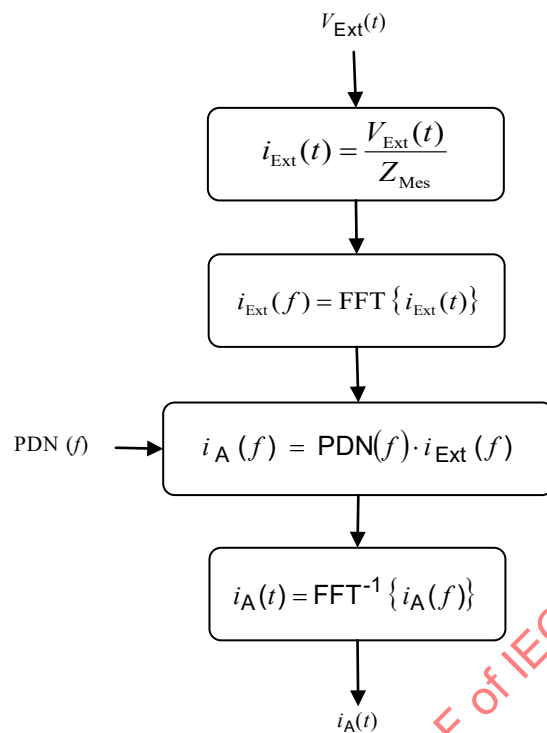
$$\begin{bmatrix} i_{An} \\ \vdots \\ i_{A1} \end{bmatrix} = [\text{PDN}] \times \begin{bmatrix} i_{Extn} \\ \vdots \\ i_{Ext1} \end{bmatrix} \quad (\text{E.1})$$

where PDN is the transfer function between i_{An} and i_{Extn} .

- Time domain

$i_{Ext}(t)$ is measured in the time domain and $i_A(t)$ has to be described in the time domain as well. In that case, the convolution of the discrete Fourier transform is used.

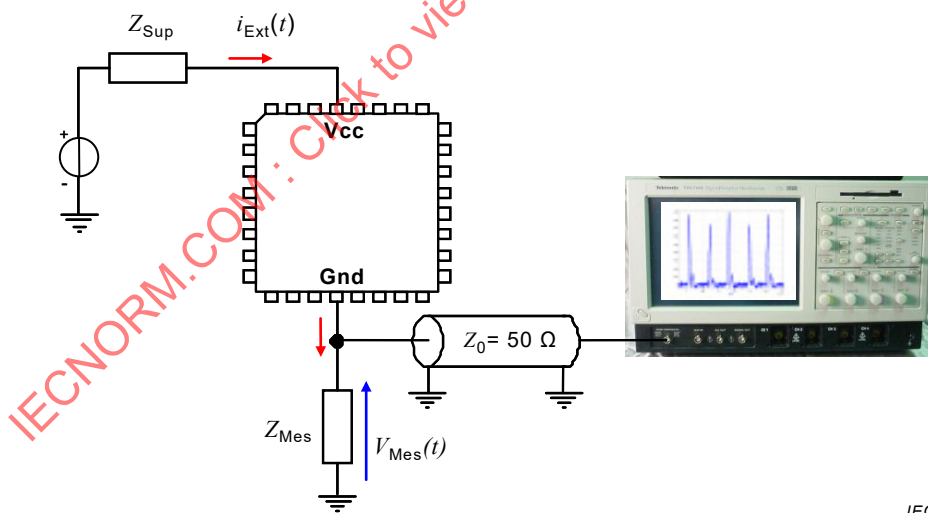
Figure E.10 shows the Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) and the inverse Fourier Transform process used to generate the IA component.



IEC

Figure E.10 – Process involved to model $i_A(t)$

First, $i_{Ext}(t)$ is measured using either the IEC 61967-4 proposal or other techniques (magnetic probe, ferrite probe, etc.). Figure E.11 presents an example in the time domain [6].



IEC

Figure E.11 – $i_{Ext}(t)$ measured using IEC 61967-4

Secondly, a DFT of FFT is applied to convert $i_{Ext}(t)$ in the frequency domain. Thirdly, the previous expression shown above is used to compute i_A in the frequency domain. Lastly, the inverse discrete Fourier transform is applied to express $i_A(f)$ in the time domain plots $i_A(t)$ and $i_{Ext}(t)$ descriptions, see Figure E.12. It should be noted that some precautions have to be taken when a Fourier Transform or convolution process is used, to avoid altering the model with unwanted information due to the Fourier transform algorithm such as spectral leakage. The observation window has to be chosen in order to have a periodic signal. The sampling period of the oscilloscope should be adjusted to give sufficient accuracy to the Fourier Transform, depending on what is to be observed.

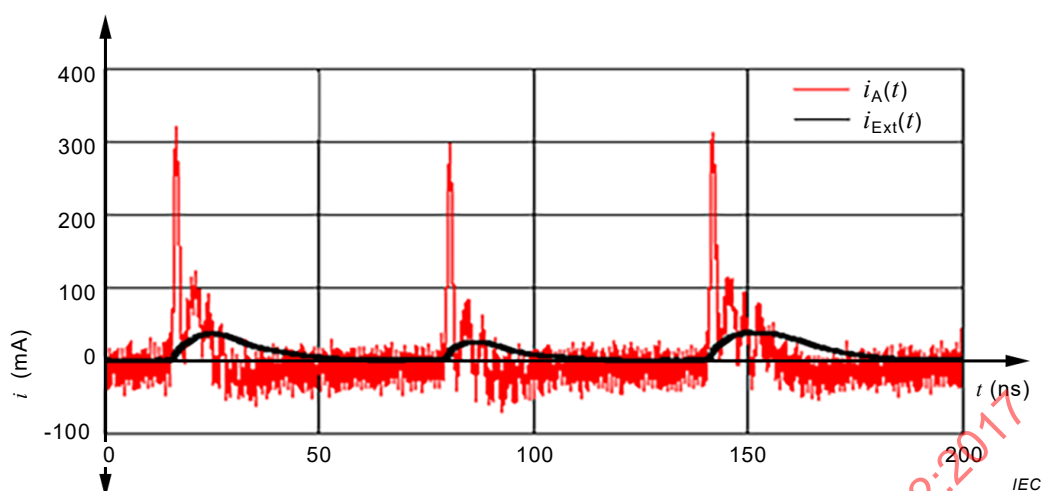


Figure E.12 – $i_A(t)$ and $i_{Ext}(t)$ profiles

In the case of multiple pairs of pins, the measurement can be performed separately on each pair and then added to obtain the total current. The methodology described above can be applied. The format of i_A depends on the simulation tools. A common format used by most of the tools is the ASCII format describing i_A by the frequency and its amplitude or by the time and its level.

E.4.2 PDN parameters

E.4.2.1 Overview

A VNA and a test board are used to extract by measurement the parameters of the PDN component. The test board has to be modelled first in order to de-embed the measurement and to obtain the parameters of the device. Certain procedures need to be followed rigorously for obtaining precise results as per CISPR17. These methods are described briefly below with the use of a two-port VNA. Nevertheless, these methods can be extended to four-port VNA.

E.4.2.2 Conventional one-port method

The simplest equivalent circuit model for one-port of a VNA connected to the pin under test is shown in Figure E.13. In general, this method is commonly used (recommended) if the impedance range of the DUT lies in the range $0,05 \Omega$ to 500Ω .

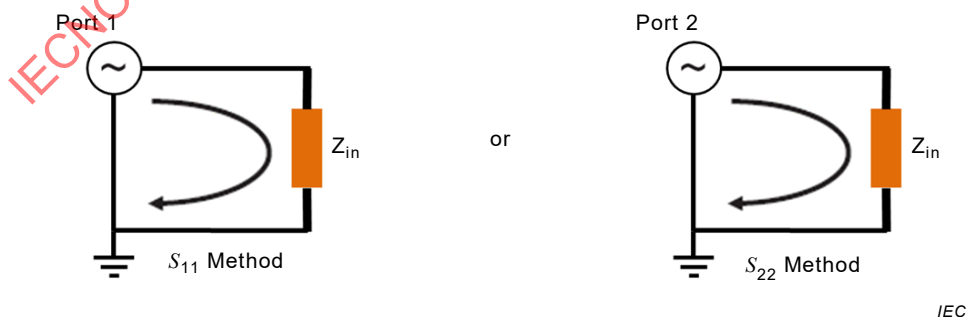


Figure E.13 – Conventional one-port S -parameter measurement

Depending on the exciting port (Port 1 or Port 2), S_{11} or S_{22} represents the impedance of the pin (Z_{in}). For example, Z_{in} can be obtained from S_{11} using:

$$Z_{in} = Z_0 \times \frac{1 + S_{11}}{1 - S_{11}} \quad (\text{E.2})$$

where Z_0 is the characteristic impedance (reference impedance) of Port 1, which is usually 50 Ω .

E.4.2.3 Two-port method for low impedance measurement

In order to measure low impedance accurately ($Z_{in} \ll Z_0$), the DUT shall be placed parallel to the two-VNA ports as shown in Figure E.14. This configuration is called shunt connection. In general, this method is commonly used (recommended) if the impedance range of the DUT is less than 0,05 Ω .

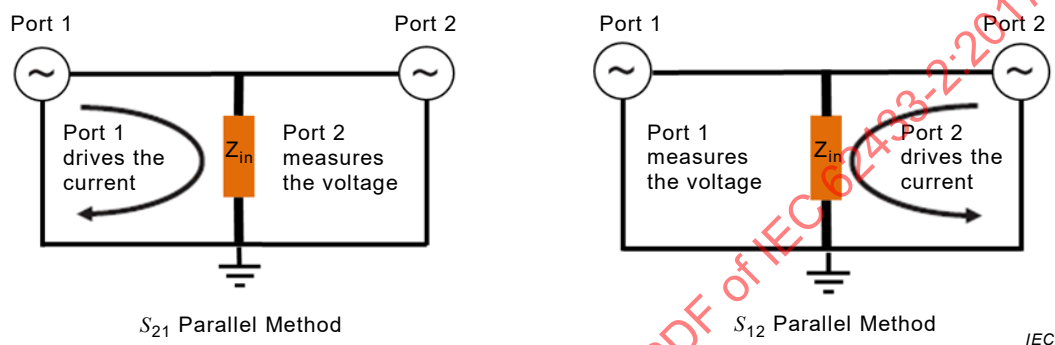


Figure E.14 – Two-port method for low impedance measurement

In this two-port method, S_{11} and S_{22} see the impedance of the DUT, with Z_0 shunting across it, from the other port. The information contained in the return loss measurements is clouded by the low impedance of the DUT and the additional Z_0 shunt from the other port. However, S_{21} or S_{12} has much more valuable information about the impedance of the DUT.

For example, Z_{in} can be obtained from S_{21} using:

$$Z_{in} = Z_0 \times \frac{S_{21}}{2 \times (1 - S_{21})} \quad (\text{E.3})$$

E.4.2.4 Two-port method for high impedance measurement

In order to measure high impedance ($Z_{in} \gg Z_0$) accurately, the DUT shall be placed in series to the two-VNA ports as shown in Figure E.15. This is also called series connection. In general, this method is commonly used (recommended) if the impedance range of the DUT is greater than 500 Ω .

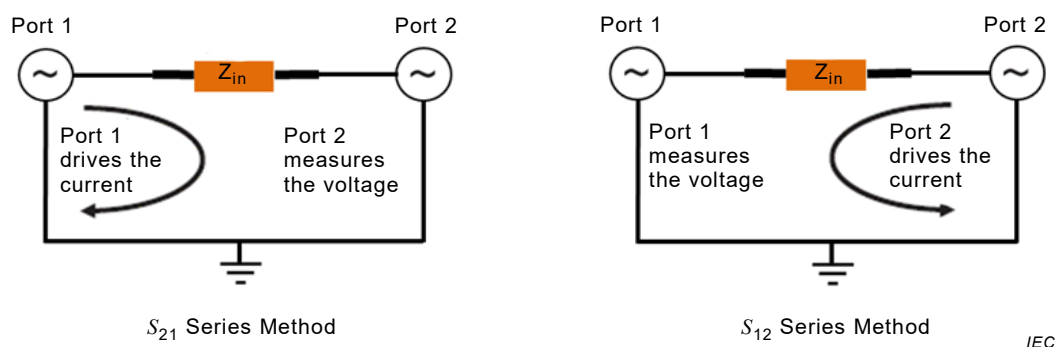


Figure E.15 – Two-port method for high impedance measurement

In this two-port method, S_{11} and S_{22} see the impedance of the DUT, with Z_0 in series, from the other port. However, S_{21} or S_{12} has much more valuable information about the impedance of the DUT.

$$Z_{in} = Z_0 \times \frac{2 \times (1 - S_{21})}{S_{21}} \quad (\text{E.4})$$

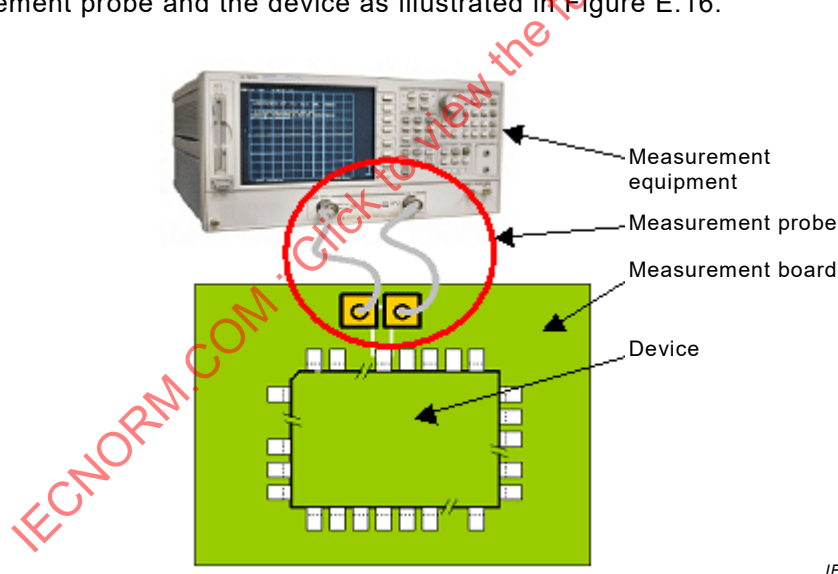
E.4.2.5 Bias conditions

Several configurations are required to extract all the PDN parameters and it is necessary to connect the power pins to ground or to the power supply rail. To achieve that, all the PDN parameters are extracted while the power supply is maintained off. The internal decoupling capacitors (digital, analogue, I/Os...) are the sum of all parasitic capacitors of CMOS transistors and the value depends on the power supply voltage. So only these PDN parameters are extracted while the power supply is on. In general when a VNA is used, a bias tee is incorporated and the VNA supplies the power to the chip under measurement. If another measurement instrument is used, an external bias tee has to be inserted between the power supply and the device.

E.4.2.6 Measurement technique and de-embedding process

E.4.2.6.1 General

Before extracting the PDN parameters and because a hardware measurement set-up is used, a de-embedding process removes all the parasitic elements of this set-up. The hardware set-up consists of measurement equipment such as a VNA, a RF impedance bridge, a measurement probe and the device as illustrated in Figure E.16.



IEC

Figure E.16 – Example of a hardware set-up used to extract the PDN parameters

E.4.2.6.2 Measurement probe, Z-Probe

The measurement probe is based on two miniature coaxial connectors as shown in Figures E.17 and E.18. A small PCB track connects both connectors to the device.

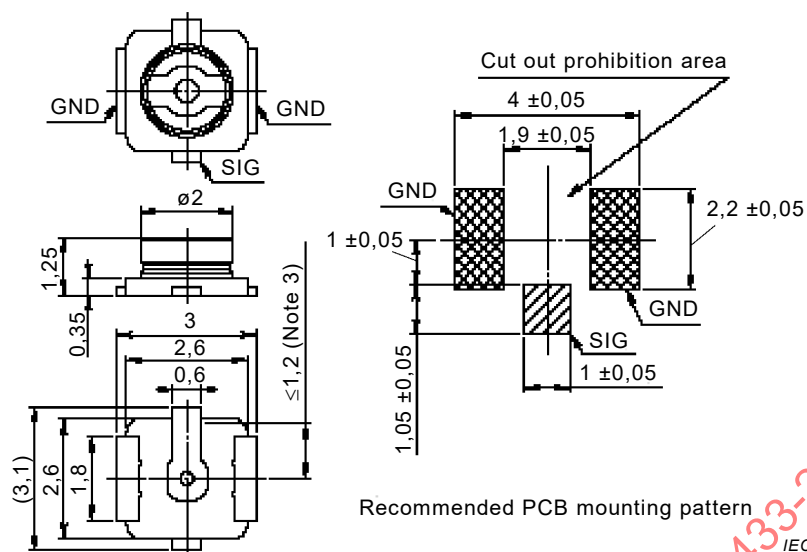
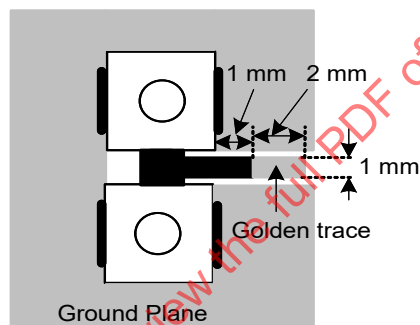


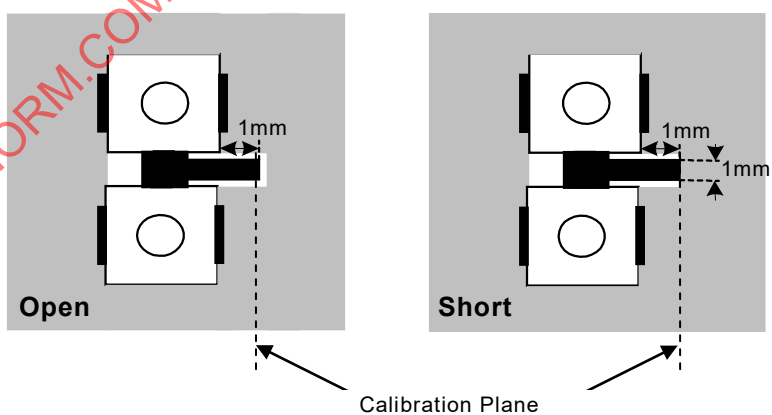
Figure E.17 – Miniature 50 Ω coaxial connectors



IEC

Figure E.18 – Impedance probe using two miniature coaxial connectors

The probe is calibrated using open- and short-circuit compensations as shown in Figure E.19.



IEC

Figure E.19 – Open and short terminations

Based on these two measurements the parasitic elements of this probe can be determined and finally the probe model is defined. An example is shown in Figure E.20.

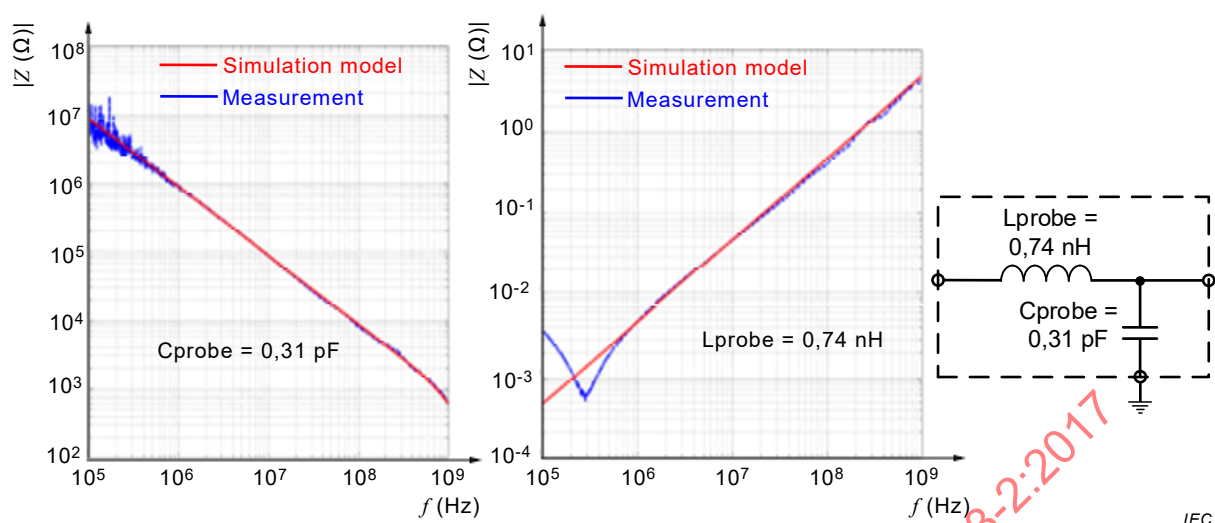


Figure E.20 – Measurement probe model

E.4.2.6.3 The measurement Board, Z_c

The measurement board can be modelled generally by a parasitic capacitor, which can be measured before mounting the device. Figure E.21 depicts the de-embedding principle and gives an example. The measurement in the upper left corner of Figure E.21 shows the measurement performed before the de-embedding process. The measurement in the upper right corner of the same figure shows the impedance profile after the probe measurement impedance has been removed exhibiting pure RLC impedance.

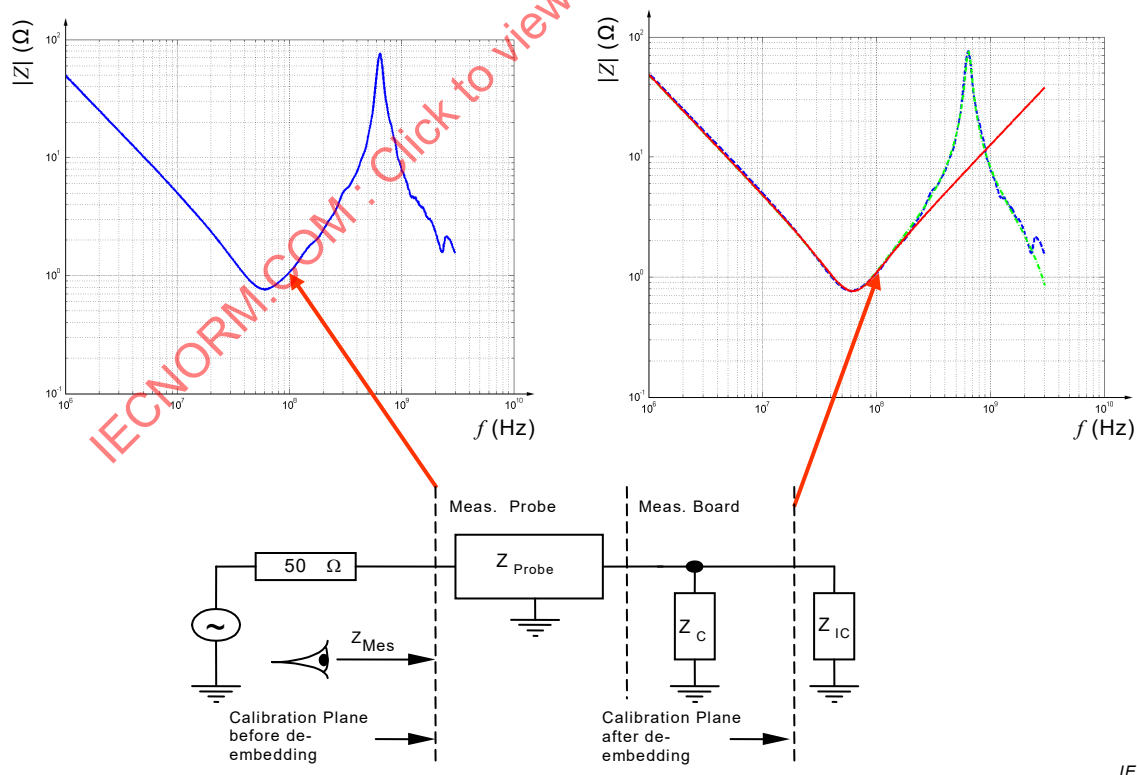


Figure E.21 – De-embedding principle

E.4.2.7 PDN parameters extraction process

E.4.2.7.1 General

The general process to build the PDN component is explained hereafter.

E.4.2.7.2 Choosing the structure of the PDN

Based on the number of pairs of power pins, a predefined structure is chosen and the number of unknown parameters is determined (example: R_{Vcc} , R_{Gnd} , R_{AGnd} , R_{AVcc} , L_{Gnd} , etc.). An example of predefined structure is given in Figure E.22 for a device having one pair of digital pins and one pair of analogue pins.

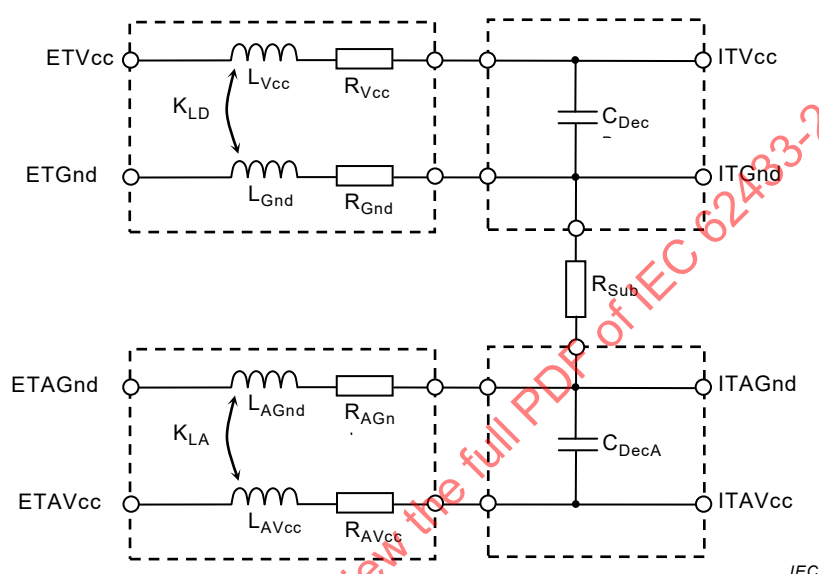


Figure E.22 – Example of a predefined PDN structure

The structure of the PDN depends on the number of power pins, the technology, the size of the circuit, etc. The correlation between the measurement and the PDN component allows validation of the predefined structure. If the number of parameters is insufficient to correctly describe the PDN, extra elements can be added to the model.

E.4.2.7.3 Definition of the number of impedance measurements to be performed

The number of measurements to be performed is at least equal to the number of unknown parameters. Table E.4 below shows the measurement configurations used to extract the R parameters.

Table E.4 – Measurement configurations and extracted RLC parameters

Measurement configurations	VNA (S11)
ZVccGnd_0v	C=1,45 nF, L=3,25 nH, RVccGnd=1,256
ZAvccAgnd	C=687pF, L=2,8 nH, RAgndAvcc=1,63
ZAvssGnd	L=5,66 nH, RSubAVccGnd=1,12
ZAgndVcc_0v	C=1,54 nF, L=5,2 nH, RVccSubAgnd=1,66
ZAvccGnd_0v	C=689 pF, L=4,91 nH, R=2,19

For example:

- **ZVccGnd_0v**: ZVcc is connected to the excitation port of the Vector Network Analyzer and the Gnd is connected to the ground.
- **ZAgndGnd**: ZAgnd is connected to the excitation port and the Gnd to the ground. This configuration allows extracting the model of the IBC component.

E.4.2.7.4 Basic measurement configurations

Three basic configurations can be met during the measurement phase. The first one is depicted in Figure E.23 and is for a series inductance and resistance. At low frequencies, the resistance is dominant and gives the R parameter. At high frequencies, the inductance dominates and the L parameter can be determined using the following equation:

$$X_L = j\omega L \Rightarrow L = \frac{X_L}{j\omega} \quad (\text{E.5})$$

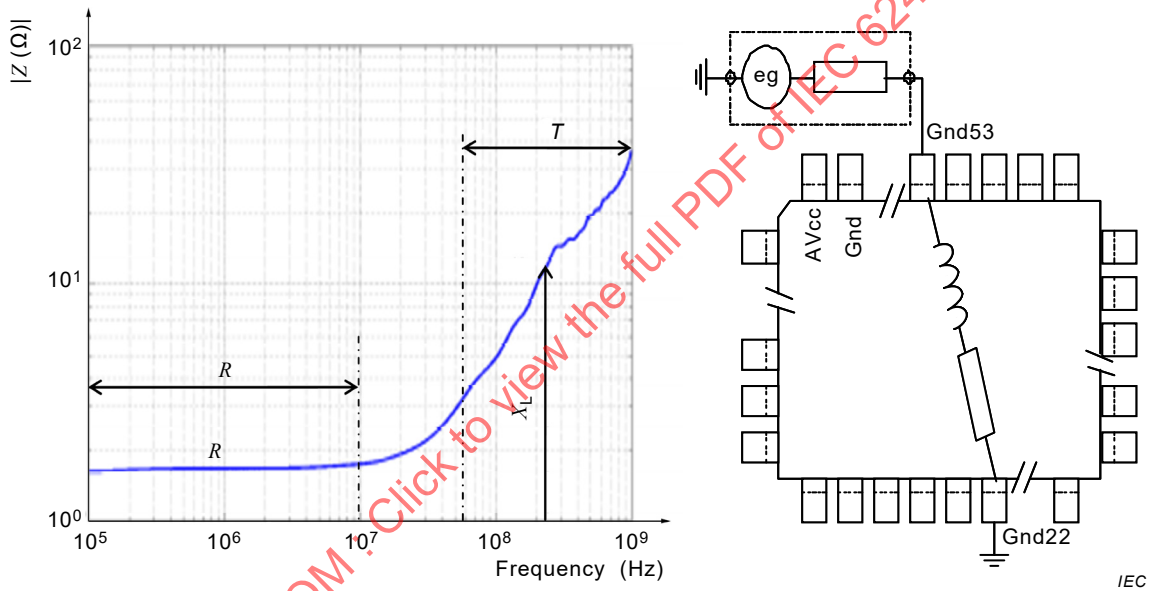


Figure E.23 – RL configuration

Figure E.24 depicts a second configuration seen when a ground and a VCC pin (Vcc52) are located on opposite sides and when there is no magnetic coupling between the two pins. The C parameter is determined in the low frequency range by using the following formula:

$$X_C = \frac{1}{j\omega C} \Rightarrow C = \frac{1}{j\omega X_C} \quad (\text{E.6})$$

The L parameter is determined in the high frequency range by using the following equation:

$$F_R = \frac{1}{2\pi\sqrt{L \cdot C}} \Rightarrow L = \frac{1}{(2\pi F_R)^2 \cdot C} \quad (\text{E.7})$$

The R parameter is determined at the anti-resonance frequency (F_R).

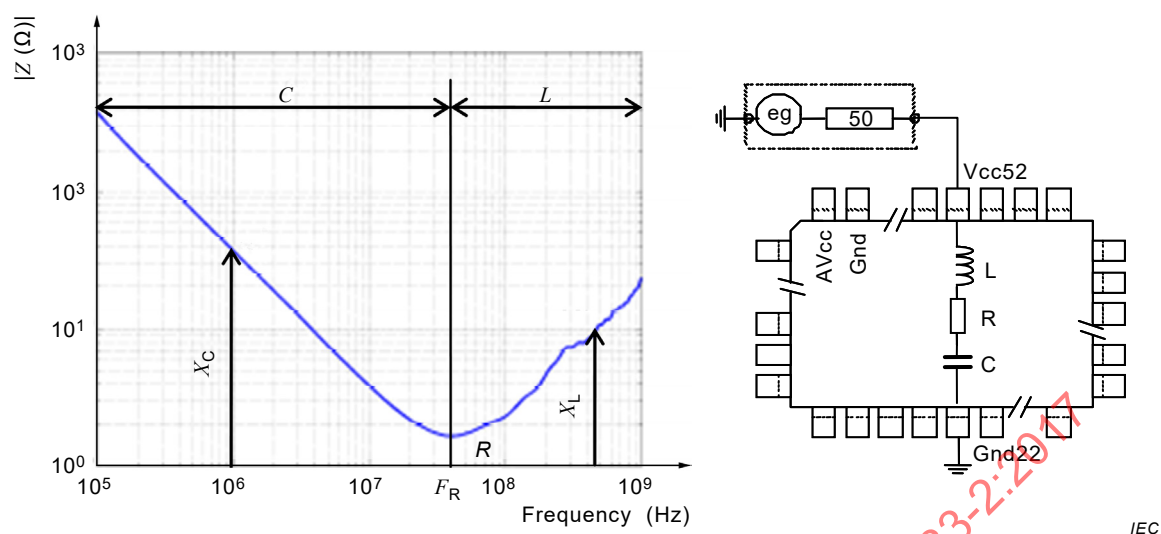


Figure E.24 – RLC configuration

The third configuration, depicted in Figure E.25, is nearly the same as the previous one except that the VCC (Vcc52) and Gnd (Gnd53) pins are closer together and magnetically coupled.

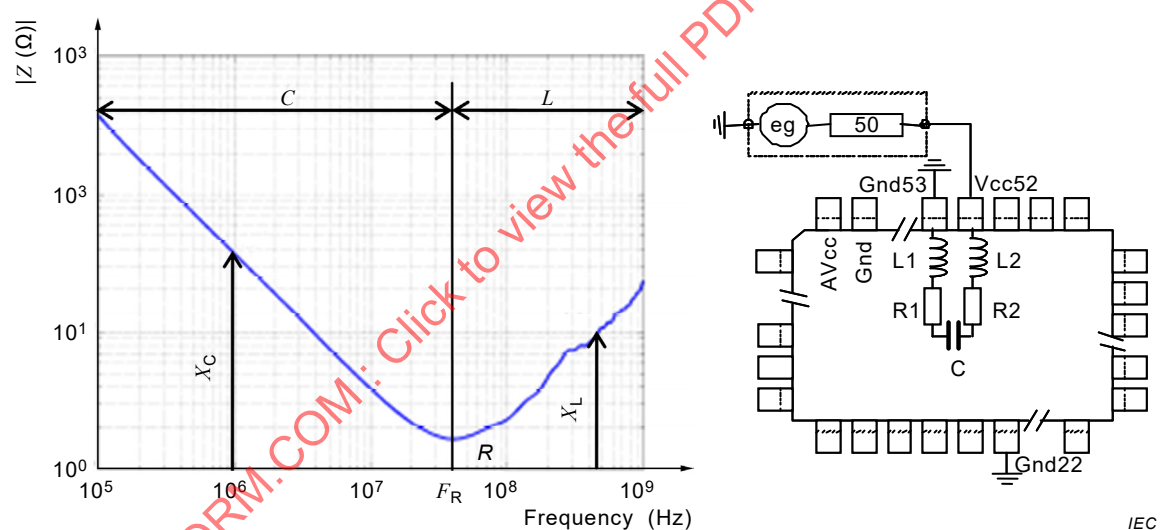


Figure E.25 – RLC with magnetic coupling configuration

The R, L and C parameters are extracted by using the same method described in the second configuration. The mutual inductance and the magnetic coupling can be determined with the following equations, assuming i_1 and i_2 (Figure E.26) have the same magnitude but opposite phase (true for the digital part of the IC).

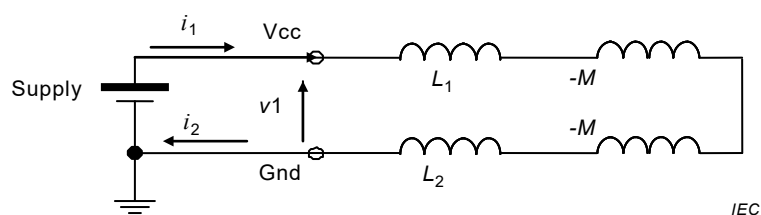


Figure E.26 – Impedance seen from Vcc and Gnd

The input impedance seen across the Vcc and Gnd pins can be determined with the following equation:

$$Z_1 = j\omega \cdot (L_1 + L_2) - j\omega \cdot 2M \quad (\text{E.8})$$

In such condition, the equivalent inductance is equal to

$$L_{eq} = L_1 + L_2 - 2M \quad (\text{E.9})$$

The mutual inductance and the magnetic coupling factor can be determined using the following formulae:

$$M = \frac{L_1 + L_2 - L_{eq}}{2} \quad (\text{E.10})$$

$$k = \frac{M}{\sqrt{L_1 \cdot L_2}} \quad (\text{E.11})$$

E.4.2.7.5 Solving the equation

A mathematical solver is used to solve the equation with five (5) unknowns as is shown in E.12.

$$\begin{bmatrix} R_{AGnd} + R_{AVcc} = R_{AGndAVcc} \\ R_{Vcc} + R_{Gnd} = R_{VccGnd} \\ R_{Sub} + R_{AVcc} + R_{Gnd} = R_{SubAVccGnd} \\ R_{Sub} + R_{Vcc} + R_{AGnd} = R_{SubVccAGnd} \\ R_{Sub} + R_{AGnd} + R_{Gnd} = R_{SubAGndGnd} \end{bmatrix} \xrightarrow{\text{Resolve}} \begin{bmatrix} R_{Vcc} \\ R_{Gnd} \\ R_{Sub} \\ R_{AGnd} \\ R_{AVcc} \end{bmatrix} \Rightarrow \begin{bmatrix} 0,995 \\ 0,365 \\ 0,44 \\ 0,295 \\ 1,335 \end{bmatrix} \quad (\text{E.12})$$

E.4.2.7.6 Building the PDN component

The same process is applied to extract the inductance parameters. The complete PDN component can be built (Figure E.27).

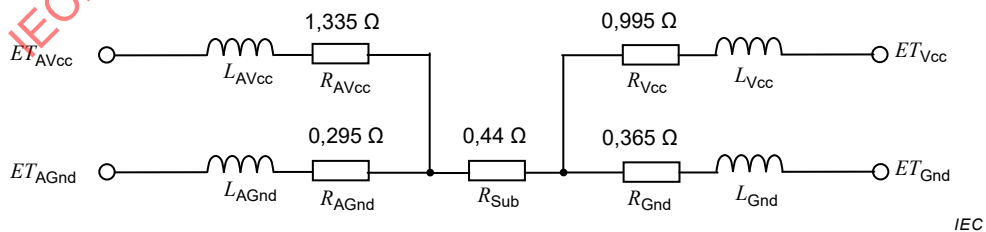


Figure E.27 – Complete PDN component

E.4.2.7.7 Choosing a correlation configuration

A correlation configuration is determined to check the accuracy of the PDN component. This configuration shall be completely different from those used to extract the RLC parameters. Figure E.28 depicts an example where all the Gnd pins are connected together on a small ground plane and all the Vcc pins are connected together on a small land plane. The device is supplied with an external power supply through the VNA. The analogue power supply is

performed by an independent external power supply in order to simplify the correlation process.

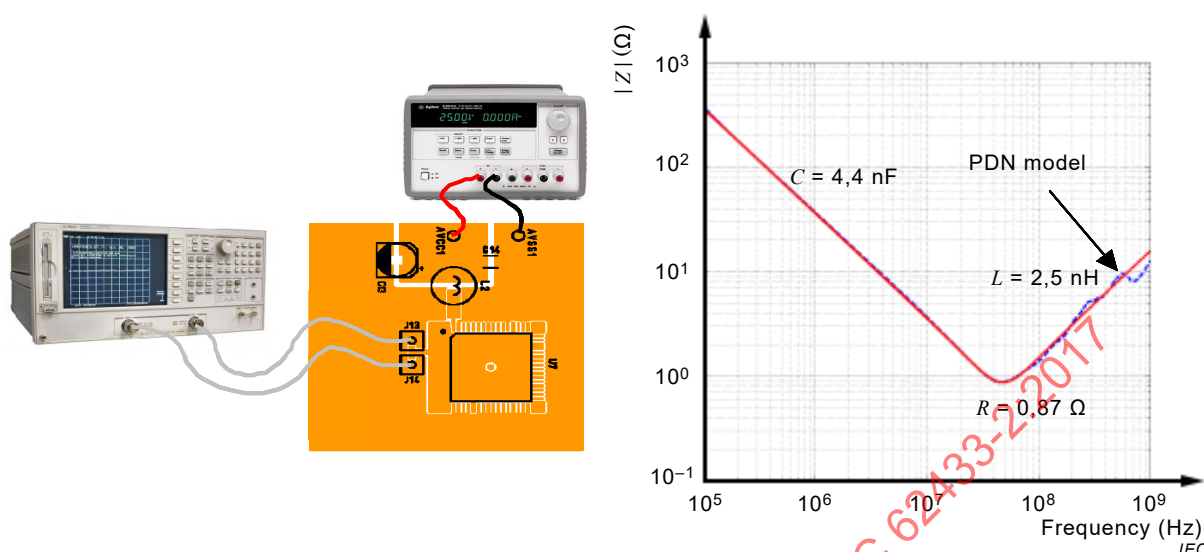


Figure E.28 – Set-up for correlation (left), measurement and prediction model (right)

The measurement is performed between the Vcc and the Ground plans and compared to the prediction of the PDN component. Figure E.28 shows quite a good correlation between the modelled PDN component and the measurement.

E.4.2.7.8 Extraction of the internal decoupling capacitors

Because these parameters vary with the power supply, the device has to be supplied to extract the internal decoupling capacitors. The device shall be supplied with the nominal power supply voltage. Figure E.29 uses the previous set-up to measure the decoupling capacitor of the digital part except that the power is maintained on. The capacitance is now doubled. Figure E.29 plots the impedances of the measurement and of the PDN simulation.

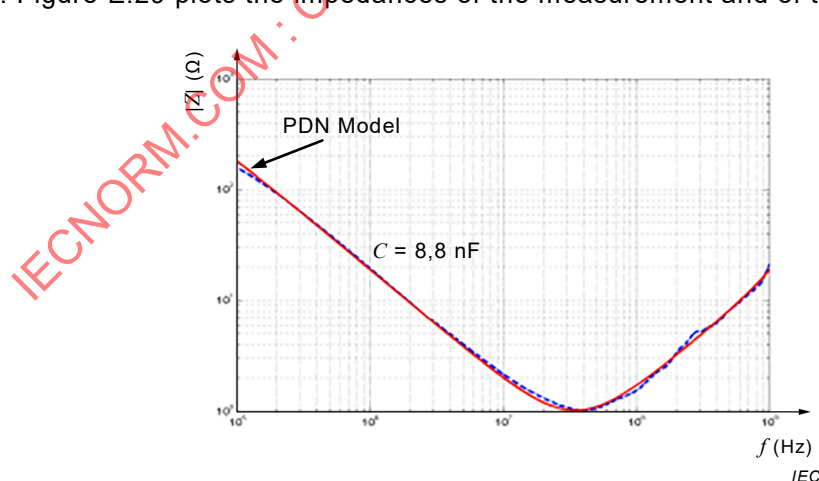


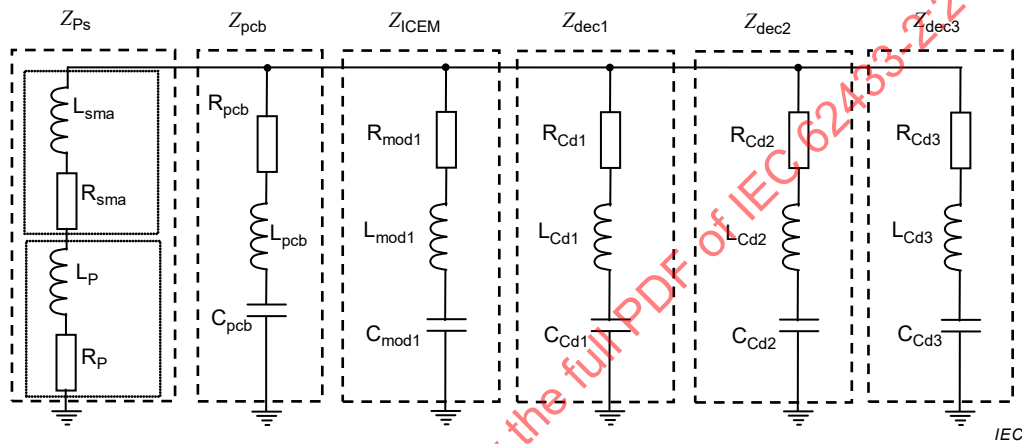
Figure E.29 – Set-up used to measure the internal decoupling capacitor

Annex F (informative)

Decoupling capacitors optimization

Annex F presents a case study in order to show how the ICEM-CE can help to design and to optimize the decoupling network during the design phase of a complete electronic system. The value and the number of decoupling capacitors can be evaluated and determined.

Figure F.1 shows the complete electronic system builds around a power supply (Z_{Ps}), a PCB (Z_{pcb}), the ICEM-CE macro-model of the device to decouple (Z_{ICEM}) and the decoupling capacitor network (Z_{dec1} , Z_{dec2} and Z_{dec3}). Each part of this system is represented with its impedance model.



Key

Z_{Ps}	Impedance of the power supply along with the connector
Z_{pcb}	Impedance of the PCB
Z_{ICEM}	PDN section of the ICEM-CE macro-model of the pin to be decoupled
Z_{dec1} , Z_{dec2} , Z_{dec3}	Impedance of the decoupling networks

Figure F.1 – Equivalent schematic of the complete electronic system

There are several ways to define the number and the values of the decoupling capacitors. The approach presented in this annex used the ICEM-CE macro-model. The process recommended to define the decoupling capacitors is commented hereafter.

The frequency bandwidth of the integrated circuit has to be determined. If there is no specific requirement, a good practice is to limit the bandwidth to the tenth harmonics of the clock frequency (example: $f_{osc} = 16$ MHz, Bandwidth = 160 MHz) or defined by the general formula:

$$\text{Bandwidth} = \frac{0,35}{t_{rise}} \text{ MHz} \quad (\text{F.1})$$

Maximum impedance measured at the Vdd and Vss pins of the IC has to be defined to guarantee a minimum noise level. This last parameter depends on the application and is based on the knowledge of the electronic system manufacturer. If there is no specific requirement, a good practice is to choose an impedance less than 1 Ω in the full frequency range (bandwidth).

The next step consists of calculating decoupling capacitors on the fundamental frequency and on the first harmonics. For example, if the clock frequency is 16 MHz and the parasitic

inductance is 1 nH, the decoupling capacitor (in Farad) is determined with the following formula:

$$C = \frac{1}{(2\pi \times 16 \times 10^6)^2 \times 1 \times 10^{-9}} = 100 \text{ nF} \quad (\text{F.2})$$

The values and the number are tuned based on the simulation of the model of the complete electronic system shown in Figure F.1. Without the ICEM-CE macro-model and the other models, it is very difficult to define the decoupling network because a capacitor is not only a capacitor but an inductance and a resistance as well. Above the resonance frequency the capacitor is inductive and can cause a new resonance if there is another capacitor around. Also putting several capacitors needs to master the combination of all these resonances.

Figure F.2 shows the impedances before and after the decoupling optimization.

- ZPDN is the ICEM-CE macro-model impedance of the device alone,
- ZVccVss is the profile impedance of the complete system before the decoupling optimization,
- ZVccVssDecMod is the total impedance of ZVccVss added to the decoupling network predicted by simulation,
- ZVccVssDecMeas is the total impedance of ZVccVss added to the decoupling network measured with the network analyzer.

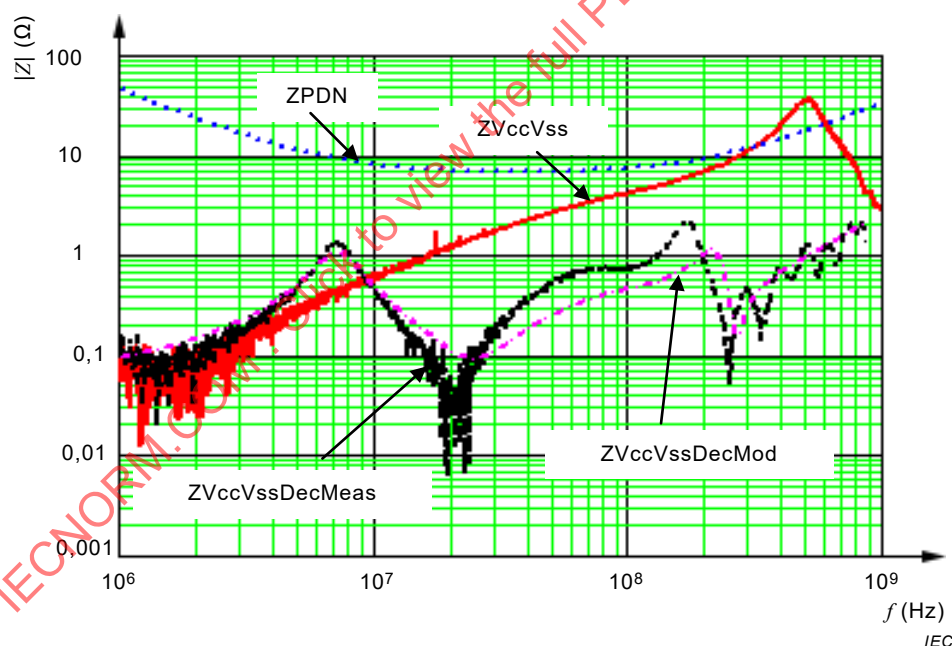


Figure F.2 – Impedance prediction and measurements

Figure F.2 shows a good correlation between ZVccVssDecMod and ZVccVssDecMeas. The remaining impedance differences come from the model of C_{dec3} . The parasitic inductance is more complex to model due to the geometry and technology used. This result is however good enough for this application.

Annex G (informative)

Conducted emission prediction

Annex G uses the ICEM-CE macro-model to predict the emission level according to the IEC 61967-4 proposal [6]. Figure G.1 shows the standard test set-up used for this measurement and how the ICEM-CE of the 16-bit microcontroller is inserted in this set-up.

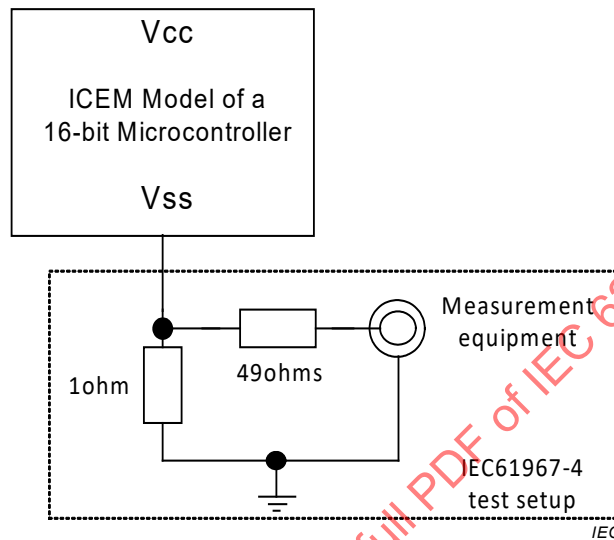


Figure G.1 – IEC 61967-4 test set-up standard

The simulated spectrum is obtained according to Figure G.1 where the ICEM-CE macro-model of the microcontroller and the IEC 61967-4 test set-up are simulated using a Spice simulator. The spectrums in Figure G.2 show the measured and simulated emissions. Thanks to the ICEM-CE macro-model the agreement is very good up to 200 MHz. Above 200 MHz, to increase the accuracy additional information has to be added.

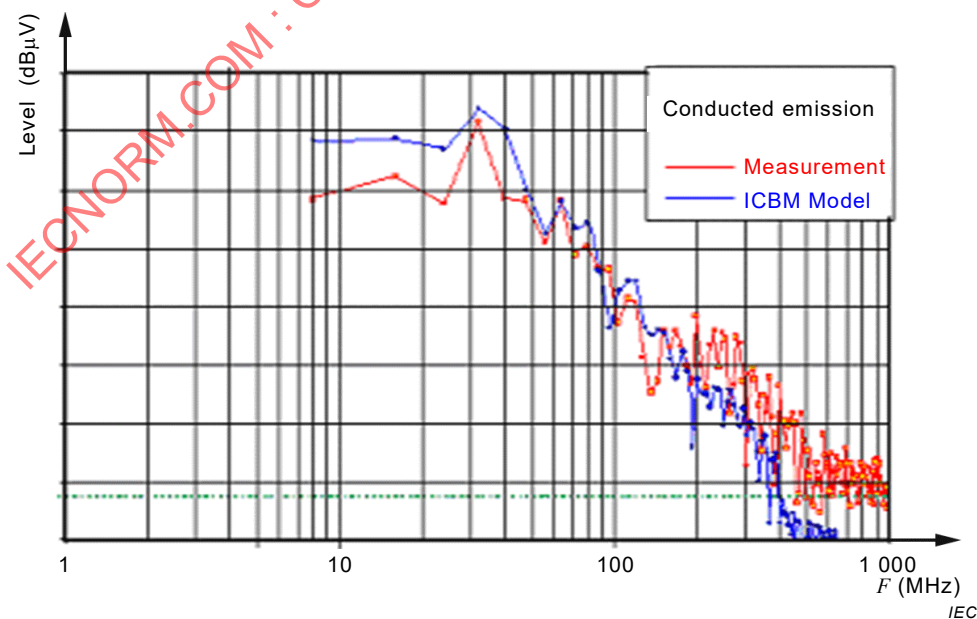


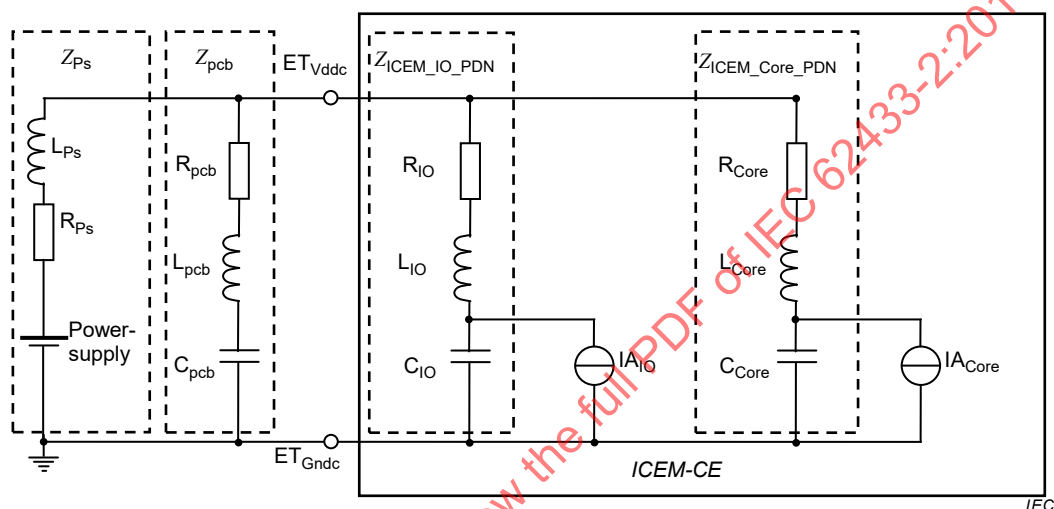
Figure G.2 – Comparison between prediction and measurement

Annex H (informative)

Conducted emission prediction at PCB level

In this application, the ICEM-CE macro-model is used to predict the level of the conducted emission measured on the ETV_{Vddc} at the PCB level.

Figure H.1 shows the complete description of a typical microcontroller application used to evaluate the conducted emissions. The figure illustrates the ICEM model of the microcontroller (Z_{ICEM}), the PCB model (Z_{pcb}) and the power supply model (Z_{Ps}). And the ICEM model consists of the PDN and IA components of the core and the I/Os.



Key

Z_{Ps}	Impedance of the power-supply
Z_{pcb}	Impedance of the PCB
$Z_{\text{ICEM_IO_PDN}}$	PDN block of the ICEM-CE macro-model of the IC core
$Z_{\text{ICEM_Core_PDN}}$	PDN block of the ICEM-CE macro-model of IOs of the IC
IA_{IO}	IA block of the ICEM-CE macro-model of IOs of the IC
IA_{Core}	IA block of the ICEM-CE macro-model of the IC core
$ET_{\text{Vddc}}, ET_{\text{Gndc}}$	External terminals of the supply pins of the IC

Figure H.1 – Prediction of ETV_{Vddc} noise level at PCB level

Figure H.2 plots the measured and the predicted level of the conducted emissions. Thanks to ICEM-CE, there is a good agreement on the noise envelope. The high frequency has not been modelled due to the limitation of the measurement equipment (500 MHz); this is why the model has filtered the high frequency noise.

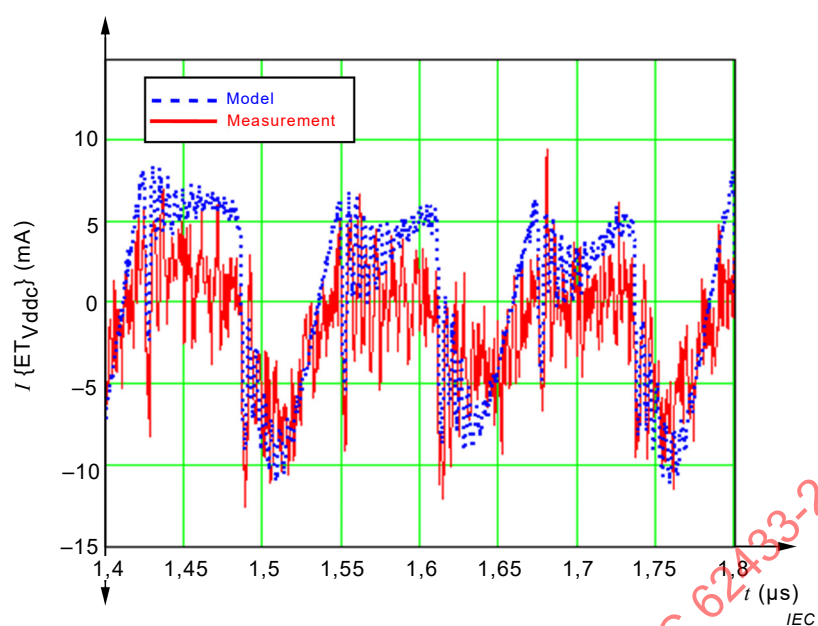


Figure H.2 – Good agreements on the noise envelope

Bibliography

- [1] IEC 61967 (all parts), *Integrated circuits – Measurement of electromagnetic emissions*
- [2] T. Quarles, A.R.Newton, D.O.Pederson, A.Sangiovanni-Vincentelli, *SPICE3 Version 3f3 User's Manual*, Departement Electrical Engineering Computer Sciences University of California, 1994
- [3] ISO 8879, *Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*, October 1986
- [4] ANSI INCITS 4:1986, *Information Systems – Coded Character Sets – 7-Bit American National Standard Code for Information Interchange*
- [5] POZAR, David M, *Microwave engineering*, 4th ed, ISBN 978-0-470-63155-3, Wiley, 2011
- [6] IEC 61967-4, *Integrated circuits – Measurement of electromagnetic emissions, 150 kHz to 1 GHz – Part 4: Measurement of conducted emissions, 1 ohm/150 ohm direct coupling method*

IECNORM.COM : Click to view the full PDF of IEC 62433-2:2017

SOMMAIRE

AVANT-PROPOS	111
1 Domaine d'application	113
2 Références normatives	113
3 Termes, définitions, abréviations et conventions	114
3.1 Termes et définitions	114
3.2 Abréviations	115
3.3 Conventions	115
4 Philosophie	116
4.1 Généralités	116
4.2 Emissions conduites provenant de l'activité du cœur (cause numérique)	117
4.3 Emissions conduites provenant de l'activité d'E/S	117
4.4 Format d'échange de données	117
5 Composants de base du macromodèle ICEM-CE	117
5.1 Généralités	117
5.2 Activité interne (IA)	117
5.2.1 Généralités	117
5.2.2 Exemples d'IA	118
5.3 Réseau de distribution passif (PDN)	119
5.3.1 Généralités	119
5.3.2 Exemples de PDN	120
6 Macromodèles de circuits intégrés	121
6.1 Types de macromodèles de circuits intégrés	121
6.2 Macromodèle général de circuit intégré	121
6.3 Macromodèle de circuit intégré à base de blocs	122
6.3.1 Composant de bloc	122
6.3.2 Composant de couplage entre blocs (IBC)	124
6.3.3 Structure de macromodèle de circuit intégré à base de blocs	125
6.4 Macromodèle de circuit intégré à base de sous-modèles	127
6.4.1 Composant de sous-modèle	127
6.4.2 Structure de macromodèle de circuit intégré à base de sous-modèles	128
7 Format CEML	130
7.1 Généralités	130
7.2 Structure CEML	131
7.3 Mots-clés globaux	131
7.4 Section <i>Header</i> (en-tête)	131
7.5 Section <i>Lead_definitions</i> (définitions des broches)	132
7.6 Macromodèles SPICE	133
7.7 Section <i>Validity</i> (validité)	135
7.7.1 Généralités	135
7.7.2 Définitions d'attributs	136
7.8 PDN	139
7.8.1 Généralités	139
7.8.2 Définitions d'attributs	140
7.8.3 Description	143
7.9 IBC	148
7.9.1 Généralités	148

7.9.2	Définitions d'attributs	149
7.10	IA.....	150
7.10.1	Généralités	150
7.10.2	Définitions d'attributs	151
7.10.3	Description	155
8	Exigences relatives à l'extraction de paramètres	156
8.1	Généralités	156
8.2	Contraintes d'extraction liées à l'environnement.....	156
8.3	Extraction des paramètres d'IA	156
8.4	Extraction des paramètres du PDN	157
8.5	Extraction des paramètres d'IBC.....	157
Annexe A	(normative) Définitions préliminaires pour la représentation XML	158
A.1	Notions fondamentales du langage XML	158
A.1.1	Déclaration XML	158
A.1.2	Eléments de base	158
A.1.3	Elément racine	158
A.1.4	Commentaires	159
A.1.5	Terminaisons de ligne.....	159
A.1.6	Hierarchie des éléments	159
A.1.7	Attributs d'éléments	159
A.2	Exigences relatives aux mots-clés	159
A.2.1	Généralités	159
A.2.2	Caractères des mots-clés	160
A.2.3	Syntaxe des mots-clés.....	160
A.2.4	Structure d'un fichier.....	160
A.2.5	Valeurs	163
Annexe B	(normative) Mots-clés CEML valides et usage.....	165
B.1	Mots-clés de l'élément racine (Root)	165
B.2	Mots-clés d'en-tête (Header) de fichier	166
B.3	Mots-clés de la section validité (Validity).....	167
B.4	Mots-clés globaux.....	168
B.5	Mots-clés de broches (Lead).....	168
B.6	Attributs de la section définition de broche (Lead_definitions).....	169
B.7	Attribut de la section macromodèles (Macromodels)	169
B.8	Mots-clés de la section réseau de distribution passif (Pdn)	170
B.8.1	Mots-clés de l'élément broche (Lead)	170
B.8.2	Mots-clés de la section liste d'interconnexions (Netlist)	172
B.9	Mots-clés de la section couplage entre blocs (Ibc)	172
B.9.1	Mots-clés de l'élément broche (Lead)	172
B.9.2	Mots-clés de la section liste d'interconnexions (Netlist)	174
B.10	Mots-clés de la section activité interne (Ia)	174
B.10.1	Mots-clés de l'élément broche (Lead)	174
B.10.2	Mots-clés de la section tension (Voltage).....	175
B.10.3	Mots-clés de la section courant (Current)	177
Annexe C	(informative) Exemple de macromodèle ICEM-CE, au format CEML.....	179
C.1	Généralités	179
C.2	Sous-modèle de PDN et d'IBC	179
C.3	Sous-modèle d'IA.....	180

C.4	Macromodèle ICEM-CE dans le domaine fréquentiel en CEML	182
C.5	Macromodèle ICEM-CE dans le domaine temporel en CEML	184
Annexe D (informative)	Conversions entre types de paramètres	186
D.1	Généralités	186
D.2	Conversion des paramètres pour un PDN à un seul port	186
D.3	Conversion des paramètres pour un PDN à deux ports	186
Annexe E (informative)	Génération des paramètres de modèle	188
E.1	Généralités	188
E.2	Structure et valeurs par défaut	188
E.2.1	Généralités	188
E.2.2	Paramètres d'IA	188
E.2.3	Paramètres du PDN	189
E.3	Génération des paramètres de modèle à partir des informations de conception	190
E.3.1	Généralités	190
E.3.2	Paramètres d'IA	190
E.3.3	Paramètres du PDN	194
E.4	Génération des paramètres de modèle à partir des mesures	197
E.4.1	Paramètres d'IA	197
E.4.2	Paramètres du PDN	200
Annexe F (informative)	Optimisation des condensateurs de découplage	211
Annexe G (informative)	Prédiction des émissions conduites	213
Annexe H (informative)	Prédiction des émissions conduites au niveau du circuit imprimé	215
Bibliographie		217
Figure 1	Exemple de décomposition d'un circuit intégré numérique pour l'analyse des émissions conduites	116
Figure 2	Composant d'IA dans le cas d'une source de courant	118
Figure 3	Exemple de caractéristiques d'IA dans le domaine temporel	119
Figure 4	Exemple de caractéristiques d'IA dans le domaine fréquentiel	119
Figure 5	Exemple de PDN à quatre bornes utilisant des éléments à constantes localisées	120
Figure 6	Exemple de PDN à sept bornes utilisant des éléments à constantes réparties	120
Figure 7	Exemple de PDN à douze bornes utilisant une représentation matricielle	121
Figure 8	Macromodèle général de circuit intégré	122
Figure 9	Exemple de composant de bloc avec une seule IA	123
Figure 10	Exemple de composants de bloc pour E/S	124
Figure 11	Exemple d'IBC avec quatre bornes internes	125
Figure 12	Relation entre blocs et IBC	125
Figure 13	Macromodèle de circuit intégré à base de blocs	126
Figure 14	Exemple de macromodèle de circuit intégré à base de blocs	127
Figure 15	Exemple de sous-modèle simple	128
Figure 16	Macromodèle de circuit intégré à base de sous-modèles	129
Figure 17	Hierarchie des éléments du modèle CEML	130

Figure 18 – Exemple de fichier de liste d'interconnexions définissant un circuit secondaire	135
Figure 19 – PDN représenté par des paramètres S au format Touchstone	146
Figure 20 – Forme d'onde d'IA simulée avec les paramètres correspondants	154
Figure A.1 – Fichiers XML (CEML) multiples	161
Figure A.2 – Fichiers XML avec des fichiers de données (*.dat)	162
Figure A.3 – Fichiers XML avec des fichiers supplémentaires	162
Figure C.1 – Exemple de brochage d'un microcontrôleur et broches modélisées	179
Figure C.2 – Topologie du sous-modèle de PDN	180
Figure C.3 – Topologie du sous-modèle d'IA	181
Figure C.4 – IA du bloc numérique dans le domaine fréquentiel	181
Figure C.5 – IA du bloc numérique dans le domaine temporel	182
Figure E.1 – Schéma typique de caractérisation de porte de courant	191
Figure E.2 – Courant de crête pendant une transition de commutation	191
Figure E.3 – Exemple de procédure d'extraction d'IA à partir de la conception	192
Figure E.4 – Influence de la technologie	193
Figure E.5 – Forme d'onde de courant finale pour une période de programme	193
Figure E.6 – Comparaison entre la mesure et la simulation	194
Figure E.7 – Exemple de modèle d'élément à constantes localisées d'un boîtier	195
Figure E.8 – Structure des circuits de la liste d'interconnexions	197
Figure E.9 – Principe du calcul d'IA dans le domaine fréquentiel	198
Figure E.10 – Processus impliqué pour modéliser $i_A(t)$	199
Figure E.11 – $i_{Ext}(t)$ mesuré en utilisant l'IEC 61967-4	199
Figure E.12 – Profils de $i_A(t)$ et $i_{Ext}(t)$	200
Figure E.13 – Mesure conventionnelle du paramètre S à un seul port	200
Figure E.14 – Méthode à deux ports pour les mesures basse impédance	201
Figure E.15 – Méthode à deux ports pour les mesures haute impédance	202
Figure E.16 – Exemple de montage matériel utilisé pour extraire les paramètres du PDN	203
Figure E.17 – Connecteurs coaxiaux miniatures 50 Ω	203
Figure E.18 – Sonde d'impédance utilisant deux connecteurs coaxiaux miniatures	204
Figure E.19 – Terminaisons en circuit ouvert et en court-circuit	204
Figure E.20 – Modèle de sonde de mesure	204
Figure E.21 – Principe d'extraction	205
Figure E.22 – Exemple de structure de PDN prédéfinie	206
Figure E.23 – Configuration RL	207
Figure E.24 – Configuration RLC	208
Figure E.25 – Configuration RLC avec couplage magnétique	208
Figure E.26 – Impédance vue de Vcc et Gnd	208
Figure E.27 – Composant de PDN complet	209
Figure E.28 – Montage pour la mesure de la corrélation (gauche) et modèle de prédiction (droite)	210
Figure E.29 – Montage utilisé pour mesurer le condensateur de découplage interne	210
Figure F.1 – Schéma équivalent du système électronique complet	211

Figure F.2 – Prédiction et mesures d'impédance	212
Figure G.1 – Montage d'essai normalisé conforme à l'IEC 61967-4	213
Figure G.2 – Comparaison entre la prédiction et la mesure	214
Figure H.1 – Prédiction du niveau de bruit de ETVddc au niveau du circuit imprimé	215
Figure H.2 – Bonnes correspondances sur l'enveloppe du bruit	216
Tableau 1 – Attributs du mot-clé <i>Lead</i> dans la section <i>Lead_definitions</i>	132
Tableau 2 – Compatibilité entre les champs <i>Mode</i> et <i>Type</i> pour une annotation CEML correcte	133
Tableau 3 – Définition de <i>Subckt</i>	133
Tableau 4 – Définition de la section <i>Validity</i>	135
Tableau 5 – Définition du mot-clé <i>Lead</i> pour la section <i>Pdn</i>	139
Tableau 6 – Formats de données valides et unités par défaut dans la section <i>Pdn</i>	142
Tableau 7 – Extensions de fichiers valides dans la section <i>Pdn</i>	143
Tableau 8 – Champs valides du mot-clé <i>Lead</i> dans la section <i>Pdn</i>	144
Tableau 9 – Définition de <i>Netlist</i>	147
Tableau 10 – Différences entre les champs des sections <i>Pdn</i> et <i>Ibc</i>	149
Tableau 11 – Champs valides du mot-clé <i>Lead</i> pour la définition d'un IBC	149
Tableau 12 – Définition du mot-clé <i>Lead</i> pour la section <i>Ia</i>	150
Tableau 13 – Définition de <i>Voltage</i> et <i>Current</i>	152
Tableau 14 – Extensions de fichiers valides dans la section <i>Ia</i>	152
Tableau 15 – Définition du mot-clé <i>Pulse</i> dans la section <i>Voltage</i> ou <i>Current</i>	153
Tableau 16 – Unités de base des champs de la section <i>Pulse</i>	153
Tableau 17 – Formats de données valides et unités par défaut pour les éléments <i>Voltage</i> et <i>Current</i>	155
Tableau A.1 – Unités logarithmiques valides	164
Tableau D.1 – Conversion des paramètres pour un PDN à un seul port	186
Tableau D.2 – Conversion de paramètres pour un PDN à un deux ports	187
Tableau E.1 – Paramètres typiques pour les technologies en logique CMOS	189
Tableau E.2 – Nombre typique de portes logiques en fonction de la technologie du CPU	189
Tableau E.3 – Paramètres de R, L et C pour divers types de boîtiers	190
Tableau E.4 – Configurations de mesure et paramètres RLC extraits	206

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

MODÈLES DE CIRCUITS INTÉGRÉS POUR LA CEM –

**Partie 2: Modèles de circuits intégrés pour la simulation
du comportement lors de perturbations électromagnétiques –
Modélisation des émissions conduites (ICEM-CE)**

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62433-2 a été établie par le sous-comité 47A: Circuits intégrés, du Comité d'étude 47 de l'IEC: Dispositifs à semiconducteurs.

Cette deuxième édition annule et remplace la première édition parue en 2008 dont elle constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

Intégration d'un format d'échange basé sur le langage XML pour la représentation des modèles.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
47A/999/FDIS	47A/1007/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62433, publiées sous le titre général *Modèles de circuits intégrés pour la CEM*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

MODÈLES DE CIRCUITS INTÉGRÉS POUR LA CEM –

Partie 2: Modèles de circuits intégrés pour la simulation du comportement lors de perturbations électromagnétiques – Modélisation des émissions conduites (ICEM-CE)

1 Domaine d'application

La présente partie de l'IEC 62433 définit des macromodèles pour circuits intégrés destinés à simuler les émissions électromagnétiques conduites sur un circuit imprimé. Ce modèle est habituellement appelé "modèle des émissions de circuits intégrés – Emissions conduites" (ICEM-CE: *Integrated Circuit Emission Model – Conducted Emission*).

Le macromodèle ICEM-CE peut également être utilisé pour modéliser une puce de circuit intégré, un bloc fonctionnel et un bloc à propriété intellectuelle (IP).

Le macromodèle ICEM-CE peut être utilisé pour modéliser à la fois des circuits intégrés numériques et des circuits intégrés analogiques.

Les émissions conduites ont fondamentalement deux origines:

- les émissions conduites aux bornes d'alimentation et dans les structures de référence de masse;
- les émissions conduites aux bornes entrée/sortie (E/S).

Le macromodèle ICEM-CE traite ces deux types d'origines en une approche unique.

La présente norme définit les structures et les composants du macromodèle pour la simulation des perturbations électromagnétiques induites par les activités internes du circuit intégré.

La présente partie de l'IEC 62433 comporte deux parties principales:

- la première partie décrit les éléments électriques du macromodèle ICEM-CE avec les exigences spécifiques données à titre d'information.
- la deuxième partie présente un format d'échange de données universel appelé CEML basé sur le format XML. Ce format permet de coder le macromodèle ICEM-CE pour obtenir une forme générique plus simple à utiliser pour simuler les émissions conduites.

2 Références normatives

Les documents suivants cités dans le texte constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC TS 62433-1:2011, *Modèles de circuits intégrés pour la CEM – Partie 1: Cadre de modèle général*

CISPR 17, *Méthodes de mesure des caractéristiques d'antiparasitage des éléments de réduction des perturbations radioélectriques et des filtres passifs*

3 Termes, définitions, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

3.1.1

borne externe

borne d'un macromodèle de circuit intégré, interfaçant le modèle avec l'environnement externe du circuit intégré

EXEMPLE Broches d'alimentation et broches d'entrée/sortie.

Note 1 à l'article: Dans la présente partie de l'IEC 62433, le nom de chaque borne externe commence par "ET" (External Terminal).

3.1.2

borne interne

borne d'un composant du macromodèle de circuit intégré, interfaçant le composant avec les autres composants du macromodèle de circuit intégré

Note 1 à l'article: Dans la présente partie de l'IEC 62433, le nom de chaque borne interne commence par "IT" (Internal Terminal).

3.1.3

section

élément XML situé un niveau en dessous de l'élément racine ou à l'intérieur d'une autre section. Une section contient un ou plusieurs éléments XML, mais aucune valeur

3.1.4

parent

mot-clé situé un niveau au-dessus d'un autre mot-clé

3.1.5

enfant

mot-clé situé un niveau en dessous d'un autre mot-clé

3.1.6

analyseur syntaxique

outil réalisant l'analyse syntaxique de données codées dans un format spécifié

3.1.7

paramètre S

paramètre de diffusion

S_{ij}

élément de la matrice S qui exprime les coefficients de transmission et de réflexion d'un dispositif

Note 1 à l'article: Chaque paramètre S , qui est le plus couramment utilisé, associe le champ électrique complexe d'une onde réfléchie ou transmise à celui d'une onde incidente. Les indices d'un paramètre S typique, S_{ij} , se rapportent aux ports de sortie et d'entrée liés par le paramètre S , qui peut varier avec la fréquence.

[SOURCE: CISPR 17:2011, 3.1.13]

3.1.8

CEML

langage de balisage pour émissions conduites:

langage d'échange de données pour le macromodèle ICEM-CE

Note 1 à l'article: L'abréviation «CEML» est dérivée du terme anglais développé correspondant «Conducted Emissions Markup Language».

3.1.9**CEMLBase**

base du langage de balisage pour émissions conduites

type abstrait à partir duquel tous les composants du modèle CEML sont directement ou indirectement dérivés dans la définition du macromodèle ICEM-CE

3.1.10**PDN****réseau de distribution passif**

composant d'un modèle de circuit intégré qui représente les caractéristiques du chemin de propagation des bruits électromagnétiques, tel que le réseau de distribution des alimentations

Note 1 à l'article: L'abréviation «PDN» est dérivée du terme anglais développé correspondant «Passive Distribution Network».

[SOURCE: IEC TS 62433-1:2011, 3.4]

3.1.11**IA**

activité interne

composant d'un modèle de circuit intégré représenté par une source de courant ou de tension provenant de l'activité des dispositifs actifs dans un circuit intégré ou dans une partie du circuit intégré

Note 1 à l'article: L'abréviation «IA» est dérivée du terme anglais développé correspondant «Internal Activity».

[SOURCE: IEC TS 62433-1:2011, 3.3]

3.1.12**IBC**

couplage entre blocs

réseau d'éléments passifs présentant un effet de couplage entre différents blocs de circuit à l'intérieur d'un circuit intégré

Note 1 à l'article: L'abréviation «IBC» est dérivée du terme anglais développé correspondant «Inter Block Coupling».

[SOURCE: IEC TS 62433-1:2011, 3.5]

3.1.13**VNA**

analyseur de réseau vectoriel

analyseur de réseau capable de mesurer les valeurs complexes des paramètres S

Note 1 à l'article: L'abréviation «VNA» est dérivée du terme anglais développé correspondant «Vector Network Analyzer».

[SOURCE: CISPR 16-1-4:2010, 3.1.21, modifiée – pour un VNA avec deux ports ou plus]

3.2 Abréviations

CEM	modèle des émissions conduites
XML	langage de balisage extensible
SPICE	programme de simulation de circuits intégrés

3.3 Conventions

Dans un souci de clarté, à quelques exceptions près, les conventions d'écriture du langage XML ont été utilisées dans le texte et les tableaux.

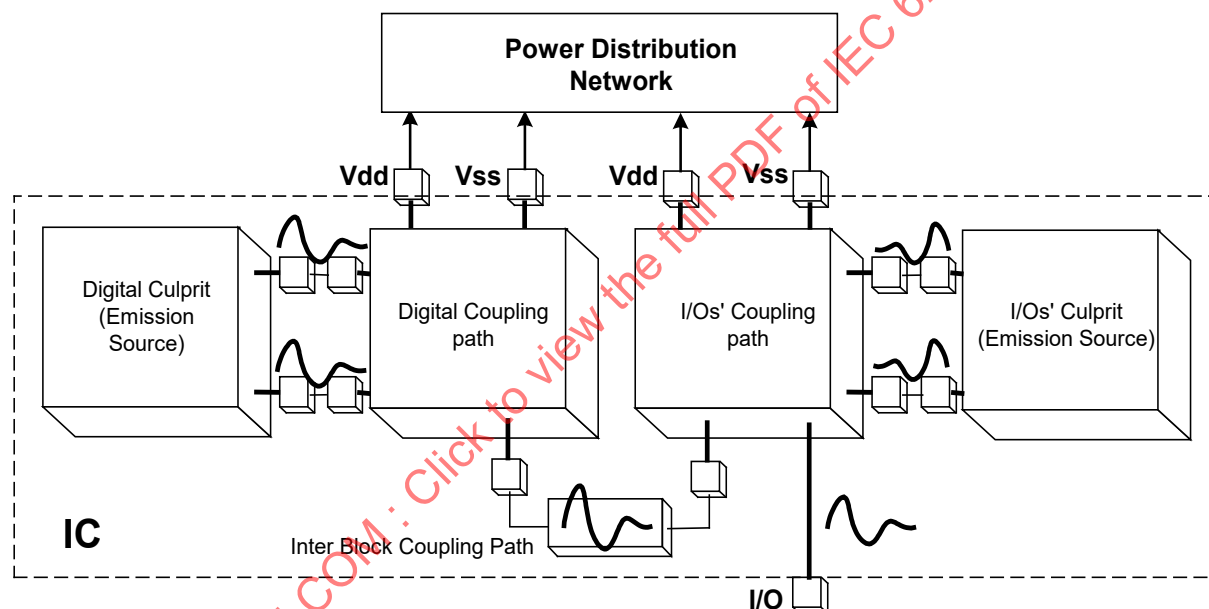
Dans le texte, le symbole " μ " est utilisé pour représenter "micro" = 1e-6. Dans les parties en langage XML, le symbole " u " est utilisé pour représenter "micro" = 1e-6.

4 Philosophie

4.1 Généralités

Les circuits intégrés comporteront de plus en plus de portes sur le silicium et la technique se développera plus rapidement. Pour prédire le comportement électromagnétique d'un équipement, il est nécessaire de modéliser efficacement les commutations des entrées/sorties et des activités internes d'un circuit intégré.

La Figure 1 est un exemple de décomposition d'un circuit intégré permettant d'effectuer l'analyse des émissions conduites. L'activité numérique interne (cause) est une source de bruit électromagnétique provenant de la commutation des dispositifs actifs. Le chemin de couplage propage les émissions vers les bornes externes du circuit intégré: broches/plots. Le chemin de couplage est le réseau de distribution des alimentations ou les lignes d'E/S à l'intérieur du circuit intégré.



IEC

Anglais	Français
Power Distribution Network	Réseau de distribution des alimentations
Digital Culprit (Emission Source)	Cause numérique (Source d'émission)
Digital Coupling path	Chemin de couplage numérique
I/Os' Coupling path	Chemin de couplage des E/S
I/O's Culprit (Emission Source)	Cause des E/S (Source d'émission)
Inter Block Coupling Path	Chemin de couplage entre blocs
IC	Circuit intégré
I/O	E/S

Figure 1 – Exemple de décomposition d'un circuit intégré numérique pour l'analyse des émissions conduites

4.2 Emissions conduites provenant de l'activité du cœur (cause numérique)

Les transitoires de courant sont générés dans la zone de cœur de la puce de circuit intégré. En raison des caractéristiques des chemins de couplage numérique, du réseau de distribution passif sur le circuit imprimé (PCB) et du découplage de la puce, une partie de ces courants transitoires apparaît sur les broches d'alimentation du circuit intégré.

NOTE Ces courants d'alimentation en dehors de la puce peuvent être mesurés en se conformant à la série IEC 61967 [1]¹.

4.3 Emissions conduites provenant de l'activité d'E/S

Les activités des E/S peuvent créer des variations de tension des niveaux de l'alimentation et de la masse, et des émissions conduites apparaissent sur les broches d'alimentation et de masse par l'intermédiaire du chemin de couplage des E/S. Par ailleurs, les signaux de sortie sur les broches de sortie elles-mêmes sont des sources d'émissions conduites sur les circuits imprimés.

NOTE Le montage de mesure est réalisé conformément à la série IEC 61967 [1].

4.4 Format d'échange de données

Les données du macromodèle ICEM-CE sont arrangées de manière imbriquée et déchiffrable. Elles sont au format XML. Ce format d'échange, appelé langage de balisage pour émissions conduites (CEML), a pour objectif de créer un accès universel simple et pratique pour le macromodèle ICEM-CE. Les définitions préliminaires pour la représentation au format XML sont données à l'Annexe A. Les mots-clés du langage CEML et leurs règles d'utilisation sont présentés en détail à l'Annexe B. Un exemple de macromodèle ICEM-CE, au format CEML, est présenté à l'Annexe C.

5 Composants de base du macromodèle ICEM-CE

5.1 Généralités

Les composants de base sont des éléments constitutifs du macromodèle de circuit intégré, ou un composant de bloc ou un composant de sous-modèle. Les composants de base du macromodèle ICEM-CE sont définis en 5.2 et 5.3.

NOTE Le composant de bloc et le composant de sous-modèle sont respectivement définis en 6.3.1 et 6.4.1.

5.2 Activité interne (IA)

5.2.1 Généralités

Le composant d'activité interne (IA) constitue la source de bruit électromagnétique provenant de la commutation des dispositifs actifs dans le circuit intégré ou dans une partie du circuit intégré. Ce composant s'applique à la fois aux circuits analogiques et numériques.

L'IA est décrite en utilisant une source de courant indépendante ou une source de tension indépendante avec deux bornes internes. La Figure 2 représente un exemple de source de courant indépendante.

¹ Les chiffres entre crochets font référence à la Bibliographie.

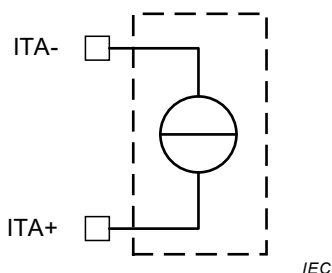


Figure 2 – Composant d'IA dans le cas d'une source de courant

Les caractéristiques du composant d'IA sont généralement décrites dans le domaine temporel. Elles peuvent également être décrites dans le domaine fréquentiel.

La description d'un composant d'IA doit contenir les informations suivantes.

- Le nom du composant d'IA
- Les noms de ses bornes internes
- Le mode de fonctionnement ou le vecteur d'essai
- Le domaine (temporel ou fréquentiel)
- La définition de l'origine des temps et le temps de cycle pour le mode de fonctionnement (pour le domaine temporel)
- La définition de l'origine des phases (pour le domaine fréquentiel)
- Les conditions de fonctionnement et les plages applicables
 - a) Plages de tensions d'alimentation
 - b) Plage de températures
 - c) Plage de fréquences
- Les caractéristiques de l'IA
 - a) Forme d'onde de courant ou de tension sur la totalité du temps de cycle (pour le domaine temporel)
 - b) Amplitude et phase de courant ou de tension, en fonction de la fréquence, sur la totalité de la plage de fréquences (pour le domaine fréquentiel)

5.2.2 Exemples d'IA

5.2.2.1 Exemple d'IA dans le domaine temporel

La Figure 3 est un exemple de caractéristiques d'IA dans le domaine temporel. La forme d'onde dépend du mode opérationnel spécifique du fonctionnement. Une forme d'onde simple telle qu'une forme d'onde triangulaire peut être utilisée pour la description d'un composant.

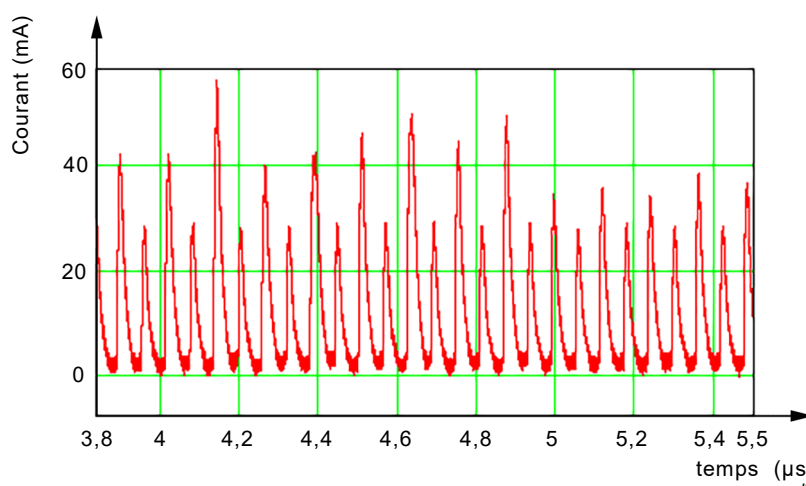


Figure 3 – Exemple de caractéristiques d'IA dans le domaine temporel

5.2.2.2 Exemple d'IA dans le domaine fréquentiel

La Figure 4 est un exemple de caractéristiques d'IA dans le domaine fréquentiel.

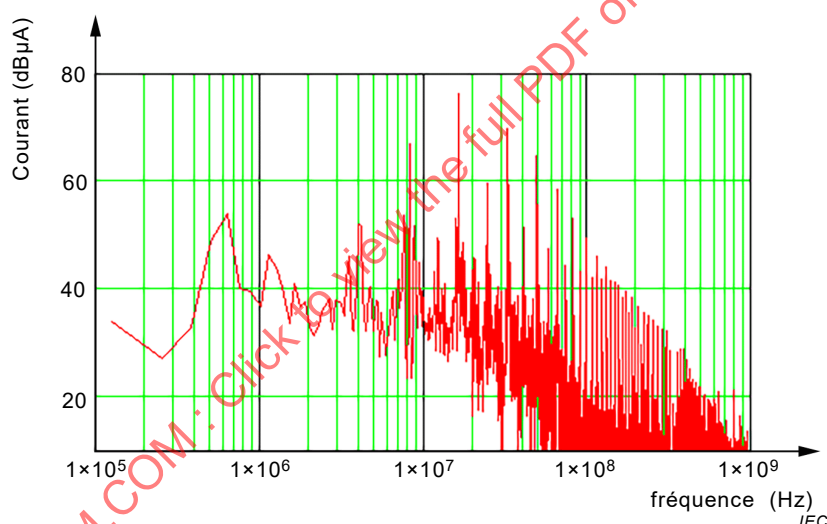


Figure 4 – Exemple de caractéristiques d'IA dans le domaine fréquentiel

5.3 Réseau de distribution passif (PDN)

5.3.1 Généralités

Le composant de réseau de distribution passif (PDN) présente les caractéristiques du chemin de propagation des bruits électromagnétiques, tel que le réseau de distribution des alimentations (partie du PDN). Le PDN peut être linéaire ou non linéaire.

Le PDN est constitué d'éléments passifs et il est muni de bornes internes et externes.

Lorsque, par hypothèse, le PDN peut être linéaire, il peut être spécifié par différents paramètres de réseau, tels que:

- $Z(f)$: impédance. Il s'agit du rapport de la tension sur le courant au niveau de l'entrée de la perturbation du PDN. Elle représente le schéma électrique de l'impédance passive d'entrée, souvent constituée d'éléments parasites (résistance, inductance et capacité)
- $Y(f)$: admittance. Il s'agit de l'inverse de $Z(f)$

- $S(f)$: paramètre S . Il s'agit du rapport de la tension de l'onde réfléchie sur de la tension de l'onde incidente au niveau de l'entrée de la perturbation du PDN. Typiquement, ce paramètre est utilisé pour mesurer les caractéristiques des ports des signaux aux fréquences radioélectriques (RF)

La conversion entre les trois types de paramètres est décrite à l'Annexe D. La plage de validité des fréquences du PDN est définie par les conditions de mesure.

Pour un PDN linéaire ou non linéaire, elle peut être décrite à l'aide d'une liste d'interconnexions de type SPICE [2].

La description d'un composant de PDN doit contenir les informations suivantes.

- a) Le nom du composant de PDN
- b) Les noms de ses bornes internes et de ses bornes externes
- c) Les plages applicables
 - 1) Plage de tensions d'alimentation
 - 2) Plage de températures
 - 3) Conditions de charge applicables si le PDN s'applique à une sortie
 - 4) Plage de fréquences applicable
- d) Les caractéristiques du PDN

5.3.2 Exemples de PDN

La Figure 5 est un exemple de PDN à quatre bornes utilisant des éléments à constantes localisées. ETVdd et ETVss sont deux bornes externes du PDN. IT0 et IT1 sont deux bornes internes.

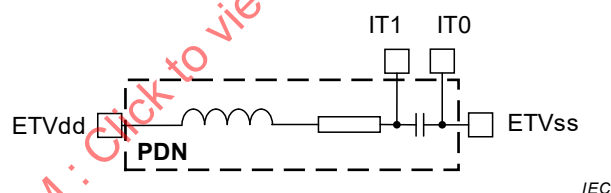


Figure 5 – Exemple de PDN à quatre bornes utilisant des éléments à constantes localisées

La Figure 6 représente la structure d'un PDN à sept bornes utilisant des éléments à constantes réparties tels que des lignes de transmission. Les ETVxx sont les quatre bornes externes, les ITVxx sont deux bornes internes et ETGnd est la masse commune des quatre lignes de transmission, raccordée à la masse du circuit imprimé.

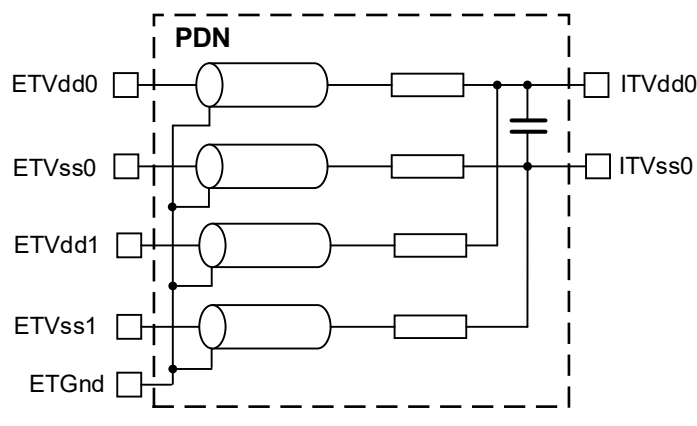
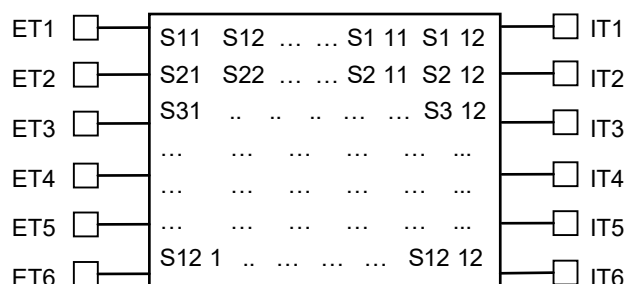


Figure 6 – Exemple de PDN à sept bornes utilisant des éléments à constantes réparties

La Figure 7 est un exemple de PDN à douze bornes utilisant des paramètres de dispersion dans un format matriciel (boîte noire). ET1 à ET6 sont des bornes externes. IT1 à IT6 sont des bornes internes. Un plan de masse au-dessous du circuit intégré modélisé est pris comme masse de référence idéale pour ces bornes.



IEC

Figure 7 – Exemple de PDN à douze bornes utilisant une représentation matricielle

6 Macromodèles de circuits intégrés

6.1 Types de macromodèles de circuits intégrés

Un circuit intégré est modélisé sous forme d'un macromodèle de circuit intégré. Trois types de macromodèles de circuits intégrés sont possibles: un modèle général, un modèle à base de blocs et un modèle à base de sous-modèles, comme défini en 6.2, 6.3 et 6.4 respectivement.

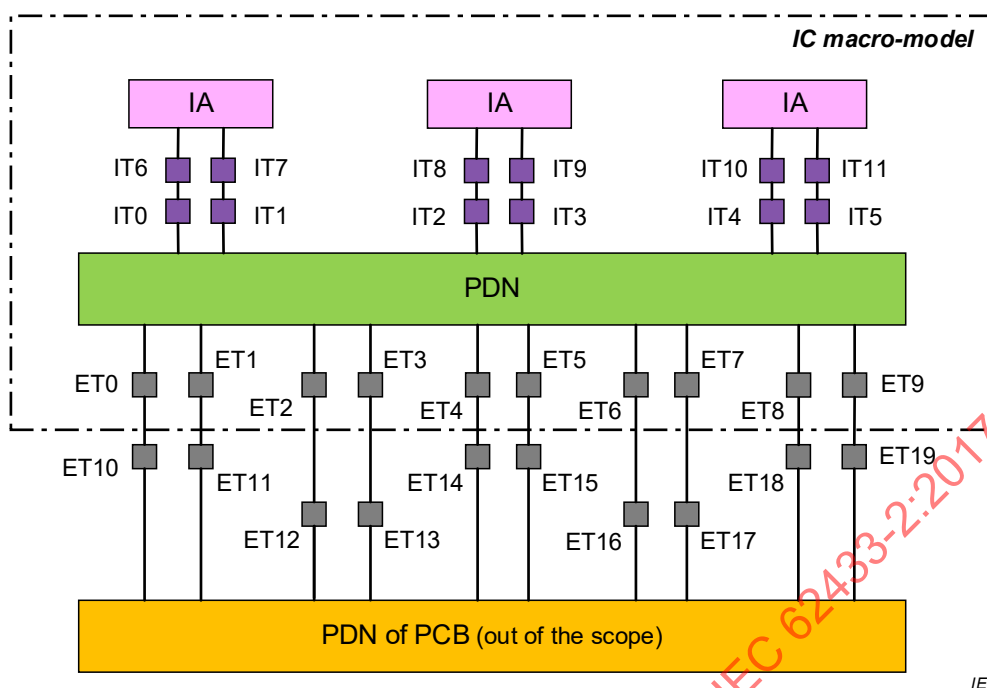
La description d'un macromodèle de circuit intégré doit contenir les informations suivantes concernant la distribution du modèle.

- Le nom du macromodèle de circuit intégré
- Le type de modèle: modèle général ou modèle à base de blocs ou modèle à base de sous-modèles
- Les noms des composants qui sont inclus dans le macromodèle de circuit intégré
- Les noms de ses bornes externes
- Les connexions des bornes internes de ses composants

6.2 Macromodèle général de circuit intégré

Le modèle général est constitué d'un seul PDN et d'une ou de plusieurs IA, comme cela est représenté à la Figure 8. Le PDN doit inclure à la fois la totalité du PDN de la puce du circuit intégré et la totalité du PDN du boîtier. Le cas échéant, un condensateur de découplage sur puce doit également être inclus dans le PDN.

NOTE Cette structure est adaptée à la distribution du modèle aux utilisateurs de circuits intégrés en raison de la divulgation restreinte des informations confidentielles du fournisseur de circuits intégrés.



Anglais	Français
IC macro-model	Macromodèle de circuit intégré
PDN of PCB (out of the scope)	PDN de circuit imprimé (hors domaine d'application)

Figure 8 – Macromodèle général de circuit intégré

6.3 Macromodèle de circuit intégré à base de blocs

6.3.1 Composant de bloc

6.3.1.1 Généralités

Le composant de bloc représente les propriétés de compatibilité électromagnétique d'un bloc fonctionnel spécifique de circuit intégré tel qu'une mémoire incorporée.

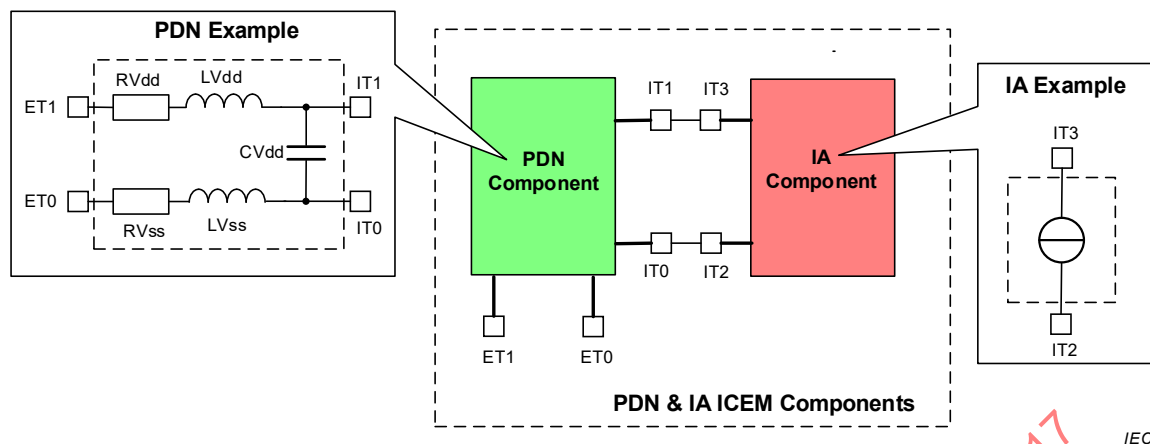
Le composant de bloc est constitué d'un seul PDN et d'une ou de plusieurs IA. Le PDN comporte un PDN du bloc fonctionnel spécifique, une partie du réseau global d'alimentation/de masse et une partie du PDN de boîtier, qui sont directement impliqués dans la fonctionnalité du bloc. Le condensateur de découplage sur puce fait partie du PDN. Le composant est équipé de bornes externes et de bornes internes.

La description d'un composant de bloc doit inclure les informations suivantes.

- Le nom du 7 de bloc
- Les noms des composants de base constituant le composant de bloc
- Les connexions des bornes internes de ses composants de base

6.3.1.2 Exemple de composant de bloc avec un seul PDN et une seule IA

La Figure 9 est un exemple de composant de bloc. Le bloc est constitué d'une IA et d'un PDN. Les bornes internes de l'IA sont raccordées aux bornes internes du PDN.



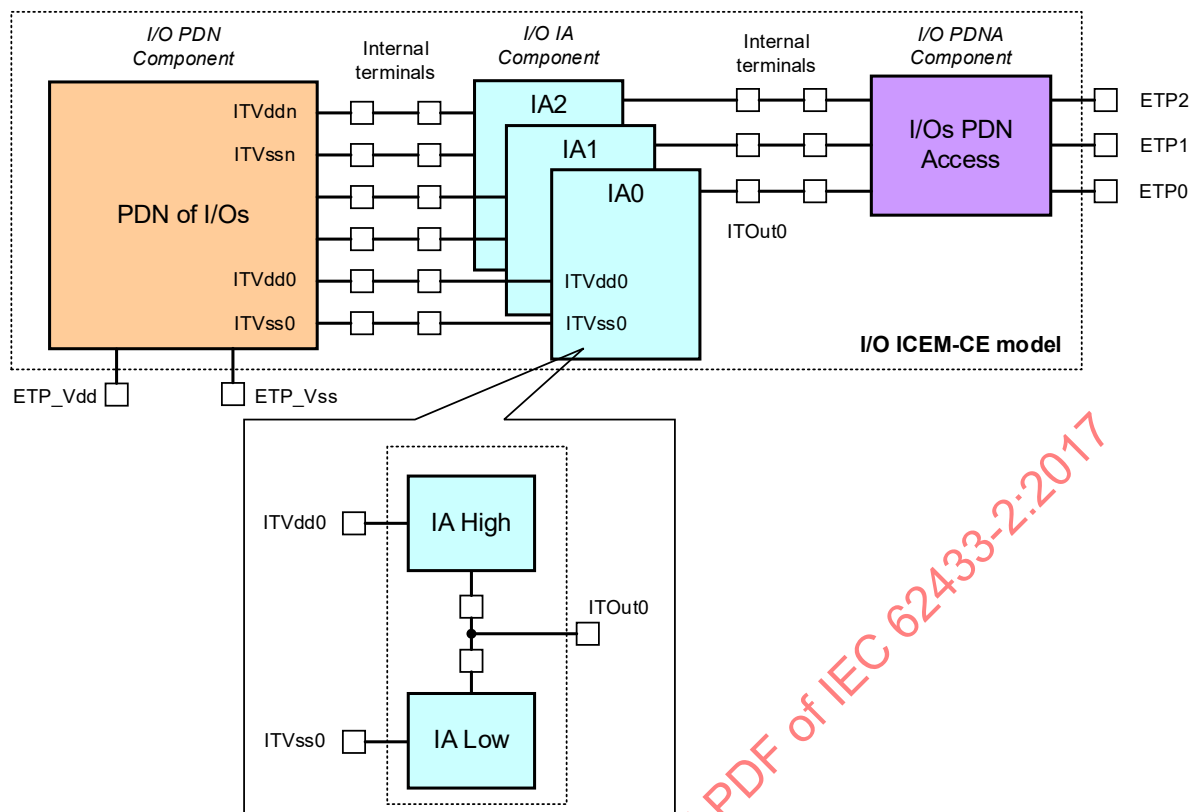
IEC

Anglais	Français
PDN Example	Exemple de PDN
PDN Component	Composant de PDN
IA Component	Composant generateur d'IA
IA Example	Exemple d'IA
PDN & IA ICEM Components	Composants ICEM de PDN et générateur d'IA

Figure 9 – Exemple de composant de bloc avec une seule IA

6.3.1.3 Exemple de composant de bloc avec plusieurs IA

La Figure 10 représente un modèle à trois E/S. Le composant de PDN d'une E/S décrit la façon dont les E/S sont alimentées et le PDNA d'E/S décrit les caractéristiques de transfert de bruit entre les bornes. Les composants générateurs d'IA d'une E/S décrivent l'activité en cours. Ils sont réalisés en utilisant deux composants générateurs d'IA; le premier pour spécifier le comportement à l'état haut et l'autre pour spécifier le comportement à l'état bas. La Figure 10 représente les deux types de composants de PDN d'une E/S du modèle d'E/S complet.



IEC

Anglais	Français
I/OPDN Component	Composant de PDN d'une E/S
Internal terminals	Bornes internes
I/OIA Component	Composant générateur d'IA d'une E/S
I/OPDNA Component	Composant de PDNA d'une E/S
PDN of I/Os	PDN d'une E/S
I/Os PDN Access	Port PDN d'une E/S
I/O ICEM-CE Model	Modèle ICEM-CE d'une E/S
IA High	IA à état haut
IA Low	IA à état bas

Figure 10 – Exemple de composants de bloc pour E/S

6.3.2 Composant de couplage entre blocs (IBC)

6.3.2.1 Généralités

Le couplage entre blocs (IBC) est un réseau d'éléments passifs présentant un effet de couplage entre blocs. L'IBC est muni de deux bornes internes au moins. De tels blocs peuvent être utilisés pour modéliser le phénomène de couplage entre différentes bornes de masse d'un circuit intégré, les pertes du substrat, les inductances mutuelles au niveau de la puce et l'isolation entre les bornes internes de mise à la terre et d'alimentation.

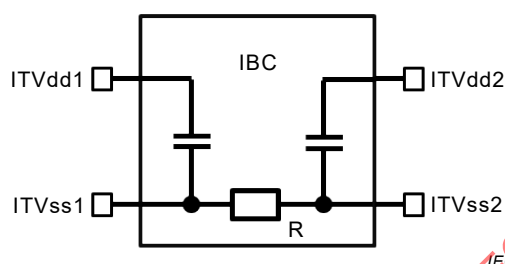
La description d'un composant d'IBC doit contenir les informations suivantes.

- Le nom de l'IBC
- Les noms de ses bornes internes
- Les plages applicables

- 1) Plages de tensions d'alimentation
 - 2) Plage de températures
 - 3) Plage de fréquences applicable
- d) Les caractéristiques de l'IBC

6.3.2.2 Exemple d'un IBC et sa relation avec le PDN

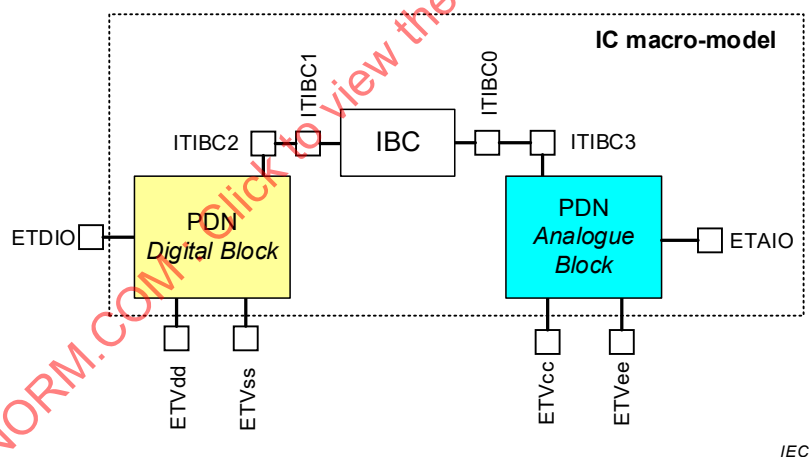
Un exemple de réseau d'IBC est représenté à la Figure 11. Dans cet exemple, les deux condensateurs modélisent les propriétés diélectriques et la résistance modélise les propriétés résistives du substrat. D'autres propriétés peuvent être modélisées en utilisant des réseaux d'IBC plus complexes.



NOTE ITVdd1, ITVss1, ITVdd2 et ITVss2 sont des bornes internes.

Figure 11 – Exemple d'IBC avec quatre bornes internes

La relation entre les différents blocs de PDN et l'IBC est représentée à la Figure 12.



IEC

Anglais	Français
IC macro-model	Macromodèle de circuit intégré
PDN Digital Block	Bloc numérique de PDN
PDN Analogue Block	Bloc analogique de PDN

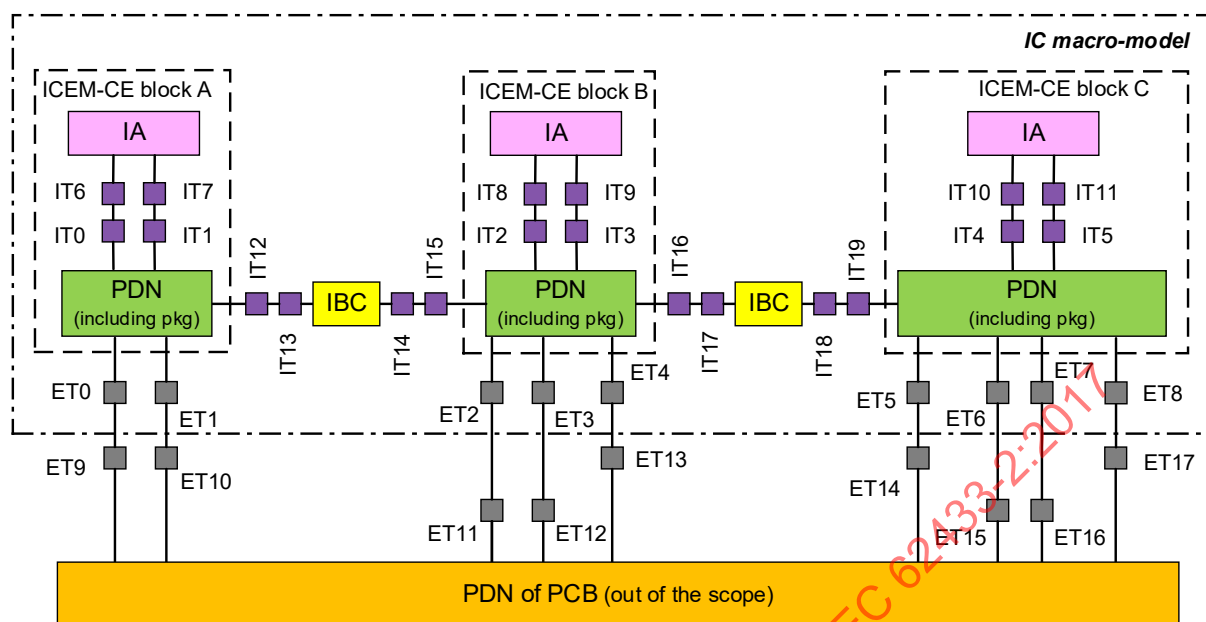
Figure 12 – Relation entre blocs et IBC

6.3.3 Structure de macromodèle de circuit intégré à base de blocs

Un exemple de structure à base de blocs est représenté à la Figure 13. Le modèle est constitué de composants de blocs et de composants d'IBC. Le PDN de câblage global sur une puce et le PDN sur un boîtier sont incorporés dans les PDN des blocs.

NOTE 1 Cette structure est adaptée à une modélisation à partir de mesures.

NOTE 2 En combinant des PDN de blocs et des IBC, cette structure peut être convertie en une structure générale.

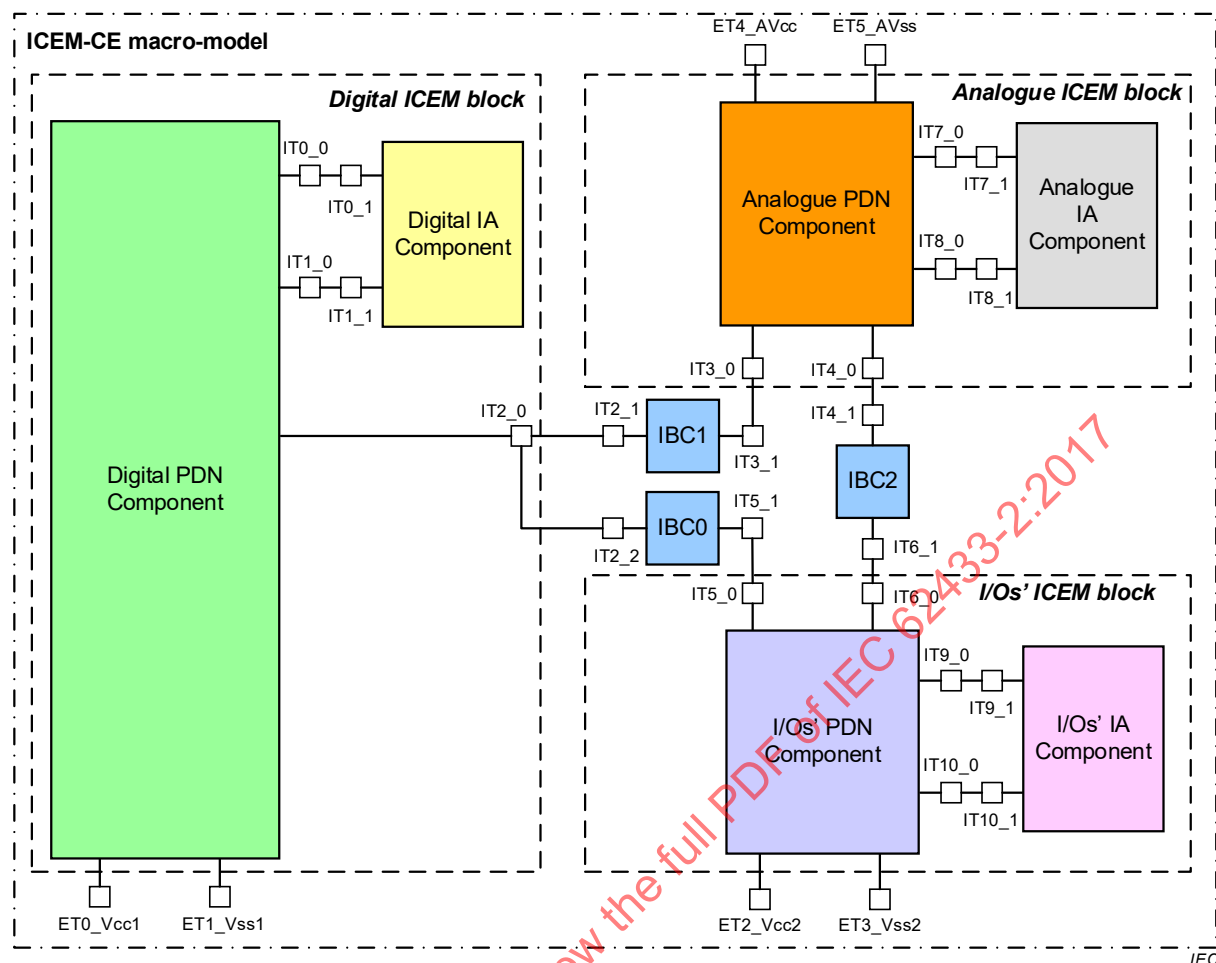


IEC

Anglais	Français
IC macro-model	Macromodèle de circuit intégré
ICEM-CE block A	Bloc A de l'ICEM-CE
ICEM-CE block B	Bloc B de l'ICEM-CE
ICEM-CE block C	Bloc C de l'ICEM-CE
PDN (including pkg)	PDN (boîtier inclus)
PDN of PCB (out of the scope)	PDN de circuit imprimé (hors domaine d'application)

Figure 13 – Macromodèle de circuit intégré à base de blocs

La Figure 14 est un exemple de macromodèle de circuit intégré à base de blocs. Chaque bloc possède deux bornes externes et trois bornes internes. Trois blocs IBC interconnectent les bornes internes IT2, IT3, IT4, IT5 et IT6. Dans cet exemple, des IBC sont utilisés pour modéliser le couplage du substrat provoqué par la résistance de surface entre les trois bornes de masse internes.



Anglais	Français
ICEM-CE macro-model	Macromodèle d'ICEM-CE
Digital ICEM Block	Bloc ICEM numérique
Analogue ICEM Block	Bloc ICEM analogique
Digital IA Component	Composant générateur d'IA numérique
Analogue PDN Component	Composant de PDN analogique
Analogue IA Component	Composant générateur d'IA analogique
Digital PDN Component	Composant de PDN numérique
I/Os' ICEM Block	Bloc ICEM d'E/S
I/Os' PDN Component	Composant de PDN d'une E/S
I/Os' IA Component	Composant générateur d'IA d'une E/S

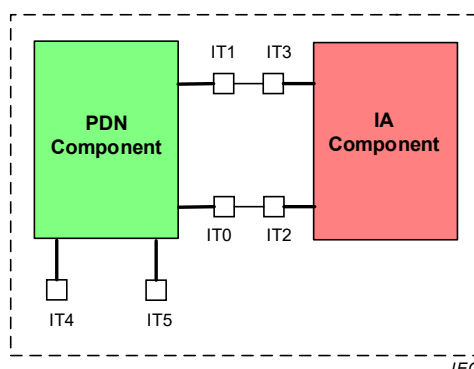
Figure 14 – Exemple de macromodèle de circuit intégré à base de blocs

6.4 Macromodèle de circuit intégré à base de sous-modèles

6.4.1 Composant de sous-modèle

Le sous-modèle de la Figure 16 représente le comportement électromagnétique de circuits fonctionnels spécifiques d'un circuit intégré. Des exemples incluent un bloc IP, des circuits fonctionnels tels qu'une mémoire intégrée, un cœur d'unité centrale de traitement (CPU), etc. Ces blocs peuvent être modélisés en utilisant le sous-modèle. Ce dernier peut être utilisé de manière répétée dans le circuit intégré et/ou dans d'autres circuits intégrés.

Le sous-modèle est constitué d'un seul PDN et d'une ou de plusieurs IA. Ce PDN est un PDN des circuits fonctionnels spécifiques. Le sous-modèle est muni de bornes internes, mais il ne comporte aucune borne externe. Un exemple de sous-modèle simple est représenté à la Figure 15.



Anglais	Français
PDN Component	Composant de PDN
IA Component	Composant générateur d'IA

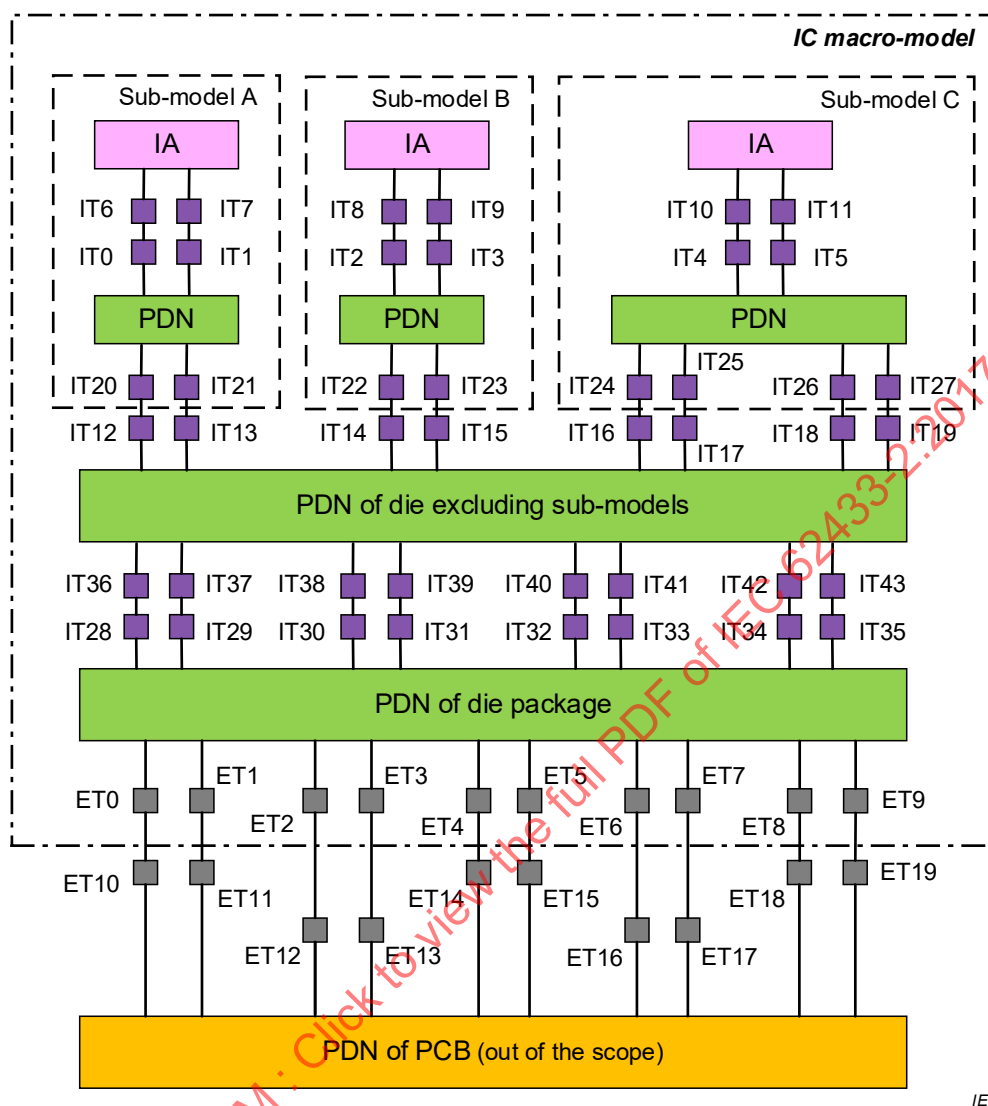
Figure 15 – Exemple de sous-modèle simple

La description d'un sous-modèle doit contenir les informations suivantes.

- Le nom du sous-modèle
- Les noms de ses bornes internes
- Les noms des composants de base qui sont inclus dans le sous-modèle
- Les connexions des bornes internes des composants de base

6.4.2 Structure de macromodèle de circuit intégré à base de sous-modèles

La structure basée sur des sous-modèles est représentée à la Figure 16. Elle est constituée d'un ou de plusieurs sous-modèles, un PDN de la puce et un PDN du boîtier. Le PDN de la puce inclut le réseau global d'alimentation/de masse de la puce de circuit intégré et le condensateur de découplage sur la puce, le cas échéant, mais ne comporte pas les PDN appartenant à des sous-modèles. Certaines IA telles que les IA pour des circuits à cellules normalisés peuvent être directement raccordées au PDN de la puce (non représenté dans la figure).



IEC

Anglais	Français
IC macro-model	Macromodèle de circuit intégré
Sub-model	Sous modèle
PBN of die excluding sub-models	PBN de la puce à l'exception des sous-modèles
PBN of die package	PBN du boîtier de puce
PBN of PCB (out of the scope)	PBN de circuit imprimé (hors domaine d'application)

Figure 16 – Macromodèle de circuit intégré à base de sous-modèles

NOTE 1 Cette structure est adaptée à une modélisation utilisant les informations de conception d'un circuit intégré.

NOTE 2 En combinant des PBN de la totalité du macromodèle de circuit intégré, un macromodèle de circuit intégré à base de sous-modèles peut être converti en un macromodèle de circuit intégré général.

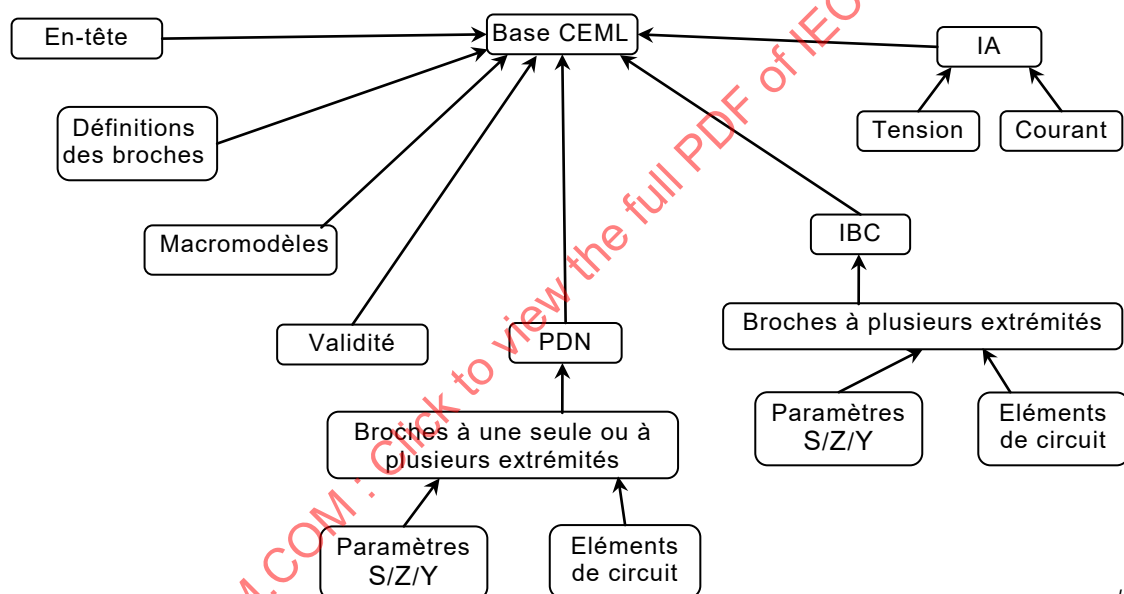
7 Format CEML

7.1 Généralités

Les données du macromodèle ICEM-CE sont présentées dans un langage appelé "langage de balisage pour émissions conduites" (CEML) au format XML. Les données du modèle sont divisées en sept parties:

- Un en-tête contenant des informations générales
- La description des broches ou des nœuds du circuit intégré
- La description de macromodèles utilisés dans les données de PDN et d'IBC
- La description des conditions de validité du modèle
- La description du PDN
- La description de l'IBC
- La description de l'IA

La hiérarchie des éléments du modèle CEML est représentée à la Figure 17.



IEC

Figure 17 – Hiérarchie des éléments du modèle CEML

Le niveau supérieur de la définition d'un modèle CEML est simplement constitué des composants suivants:

```

beginning of model definition
  model header definition
  DUT lead definitions
  SPICE macro-models
  model validity conditions
  model PDN
  model IBC
  model IA
end of model definition
  
```

Ce format d'échange utilise le langage XML 1.0 (quatrième édition) pour structurer les informations. Le langage XML est dérivé du langage normalisé de balisage généralisé [3].

Les règles présentées ci-dessous garantissent que le fichier peut être analysé correctement par un analyseur syntaxique de modèle d'émissions conduites (CEM). Un exemple de fichier CEML est présenté à l'Annexe C.

7.2 Structure CEML

En tenant compte de toutes les règles définies à l'Annexe A, et en respectant les exigences relatives aux descriptions de modèles de l'IEC TS 62433-1, un fichier de macromodèle ICEM-CE typique est représenté ci-dessous:

```
<?xml version="1.0" encoding="UTF-8"?>
<CEmodel>
<!-- HEADER -->
  <Header>
    ...
  </Header>
<!-- DUT LEAD DEFINITIONS-->
  <Lead_definitions>
    ...
  </Lead_definitions>
<!-- SPICE MACROMODEL DEFINITIONS -->
  <Macromodels>
    ...
  </Macromodels>
<!-- MODEL VALIDITY CONDITIONS -->
  <Validity>
    ...
  </Validity>
<!-- MODEL PDN DATA -->
  <Pdn>
    ...
  </Pdn>
<!-- MODEL IBC DATA -->
  <Ibc>
    ...
  </Ibc>
<!-- MODEL IA DATA -->
  <Ia>
    ...
  </Ia>
</CEmodel>
```

Les sections *Header*, *Lead_definitions*, *Validity*, *Pdn* et *Ia* sont les informations obligatoires minimales du macromodèle ICEM-CE. Les sections *Macromodels* et *Ibc* sont facultatives. Chaque section du fichier est présentée ci-dessous.

7.3 Mots-clés globaux

Les sections *Documentation*, *Notes* et *Unit* sont des mots-clés globaux qui peuvent être placés n'importe où dans le fichier, sauf à l'intérieur d'un élément contenant une valeur. Voir B.4 pour plus d'informations sur l'utilisation de ces mots-clés.

7.4 Section *Header* (en-tête)

Les lecteurs peuvent s'interroger sur les raisons d'une section *Header* indépendante. Une approche plus simple pour créer les informations de l'en-tête consisterait à les placer directement au niveau le plus élevé, sous <CEmodel> ... </CEmodel>. Au lieu de cela, les informations sont groupées dans l'élément XML <Header>, parce qu'une telle approche aide à organiser les composants dans un meilleur format et facilite la lecture des définitions des modèles. Il est donc proposé de définir l'information d'en-tête dans la balise *Header*. Les informations minimales sont le numéro de la version du modèle, le nom de fichier et le

numéro de la version du fichier. D'autres informations peuvent être fournies dans le contenu de l'en-tête, de dimension fixée librement, conformément à l'IEC TS 62433-1, par exemple:

- La référence du DUT
- Le nom de l'auteur
- La date
- La méthode de mesure
- Les droits d'auteur
- Les informations de non-responsabilité
- La documentation

EXEMPLE Une section *Header* peut être définie comme suit:

```
<Header>
  <Cem_ver>1.0</Cem_ver>
  <Filename>ExampleICEMCE_file.CEML</Filename>
  <File_ver>1.0</File_ver>
  <Author>Valeo 1</Author>
  <Dut>MCYY/Dut>
  <Date>April 3, 2013</Date>
  <Meas_method> PDN using Netlist, IA in time-domain </Meas_method>
</Header>
```

Une liste détaillée de mots-clés valides dans la section *Header* est présentée à l'Annexe B.

7.5 Section *Lead_definitions* (définitions des broches)

Cette section décrit les différentes broches ou les différentes bornes du circuit intégré en essai. Chaque broche de la section *Lead_definitions* utilise la balise *Lead* définie au Tableau 1. Différentes structures de broches (*Lead*) sont présentées les unes après les autres dans une liste pour former la structure *Lead_definitions*.

Tableau 1 – Attributs du mot-clé *Lead* dans la section *Lead_definitions*

<i>Lead</i>
Id: identité de la broche présentée sous la forme d'une chaîne valide (exigée) Name: nom de la broche conformément à la désignation de la fiche technique (facultatif). Valeur par défaut ="None" Mode: mode dans lequel la broche est utilisée pour le modèle ICEM-CE ("GND") (facultatif). Valeur par défaut ="None" Type: type de la broche pour le modèle ICEM-CE ("internal", "external") (facultatif). Valeur par défaut ="external"

Pour chaque structure de broche dans la section *Lead_definitions*, le champ *Id*, représentant l'identité de la broche, est obligatoire. En outre, la broche peut être définie avec les champs *Name*, *Mode* et *Type*. Si le champ *Name* n'est pas défini ou n'est pas renseigné, sa valeur par défaut est "None". La valeur du champ *Mode* est "None" dans les cas suivants:

- Si *Mode* n'est pas renseigné pour une broche particulière
- Si sa valeur est explicitement définie comme "None" parce que la broche respective est différente de GND (chemin de retour de référence).

Le champ *Type* est facultatif et peut prendre la valeur "external" ou "internal". Ces termes sont définis en 3.1.1 et 3.1.2.

Les broches dont *Type*="external" sont utilisées assurer la liaison entre le modèle et l'environnement externe du circuit intégré, par exemple les broches d'alimentation et les broches d'E/S. Les bornes "internal" sont utilisées pour assurer la liaison entre différents

blocs du macromodèle ICEM-CE à base de blocs ou à base de sous-modèles (voir 6.3 et 6.4). Les bornes "internal" doivent être définies avec *Mode*="None". Ces broches sont dédiées à l'interfaçage de différents blocs des composants ICEM-CE à base de blocs ou à base de sous-modèles.

Les broches définies avec *Mode*="GND" sont considérées comme des bornes "external".

NOTE Des bornes de masse internes peuvent être spécifiées comme n'importe quel autre nœud interne, c'est-à-dire *Type*="internal" avec *Mode*="None".

Le Tableau 2 présente la compatibilité entre les champs *Mode* et *Type* de la structure *Lead* pour une annotation CEML correcte par l'analyseur syntaxique de modèle d'émissions conduites.

Tableau 2 – Compatibilité entre les champs *Mode* et *Type* pour une annotation CEML correcte

Mode	Type	
	external	internal
GND	Oui	Non
None	Oui	Oui

Exemple de section *Lead_definitions* représentant la Figure 5:

```
<Lead_definitions>
  <Lead Id="1" Name="ETVdd" Type="external"/>
  <Lead Id="2" Name="IT1" Type="internal" Mode="None"/>
  <Lead Id="3" Name="IT0" Type="internal"/>
  <Lead Id="4" Name="ETVss" Mode="GND"/>
</Lead_definitions>
```

7.6 Macromodèles SPICE

Cette section décrit les différents macromodèles SPICE dans un format de liste d'interconnexions [2].

Ces circuits secondaires sont référencés dans la balise *Pdn* de la section *Netlist* (liste d'interconnexion) comme cela est expliqué en 7.8.3.4 et 7.8.3.5. Chaque macromodèle SPICE de la section *Macromodels* est défini à l'aide de la balise *Subckt*, définie au Tableau 3. Cette section est facultative pour le macromodèle ICEM-CE.

Tableau 3 – Définition de *Subckt*

<i>Subckt</i>
Name: nom du macromodèle (exigé)
Nodes: nœuds externes de connexion du circuit principal (exigés)
Kind: format de liste d'interconnexions SPICE (facultatif), valeur par défaut: "SPICE3"
Data_files: macromodèle défini dans un fichier externe (facultatif)

Le mot-clé *Subckt* a deux champs obligatoires: *Name* et *Nodes*. Le champ *Name* est constitué de lettres et de chiffres définis en A.2.5.4. Puisque ces définitions partagent le même espace de noms dans le format CEML, chaque circuit secondaire doit porter un nom unique.

Le champ *Nodes* définit les nœuds externes connectés à l'extérieur du circuit secondaire, c'est-à-dire les nœuds à travers lesquels un élément du circuit secondaire est connecté au circuit principal. Les différents nœuds externes sont définis sous la forme d'une séquence et sont séparés par une virgule (","). Ces nœuds appartiennent exclusivement à la définition de *Subckt*. Ces nœuds peuvent être identifiés par des chiffres et/ou des caractères [2].

L'attribut facultatif *Kind* indique à l'analyseur syntaxique de modèle d'émissions conduites que le circuit secondaire défini respecte une syntaxe spécifique. La version 1 du langage CEML prend en charge les syntaxes de liste d'interconnexions de type SPICE conformes à une norme industrielle [2]. Si le champ *Kind* n'est pas renseigné, sa valeur par défaut est la valeur générique "SPICE3".

Un exemple d'élément *Subckt* est représenté dans l'exemple 1.

EXEMPLE 1 Exemple d'élément *Subckt* de type SPICE.

```
<Subckt Name="PDN_pin1" Nodes="Node1,Node2" Kind="SPICE3">
  L1 Node1 int1 2.2e-9
  R1 int1 int2 10e-3
  C1 int2 int3 800e-12
  L1 int3 Node2 2.2e-9
</Subckt>
```

Pour assurer la compatibilité entre plusieurs types SPICE, "0" n'est pas autorisé comme nœud externe. Les différents éléments d'un circuit secondaire sont présentés les uns après les autres dans une liste pour former la section *Macromodels* (voir l'exemple 2).

EXEMPLE 2 Exemple de section *Macromodels* avec plusieurs éléments *Subckt*.

```
<Macromodels>
  </Subckt Name="PDN_example1" Nodes="Node1,Node2">
  ...
</Subckt>
  </Subckt Name="PDN_example2" Nodes="Node1,Node2,Node3">
  ...
</Subckt>
  ...
</Macromodels>
```

Une fois définis, les différents circuits secondaires peuvent être utilisés n'importe où dans les balises *Pdn* et *Ibc* en utilisant un identifiant commençant par le caractère "X". L'appel doit être fait dans la section *Netlist* et il est référencé en utilisant le champ *Name* (voir 7.8.3.4 et 7.8.3.5). Le nombre de nœuds sur la ligne d'appel doit correspondre au nombre indiqué dans la ligne des attributs *Nodes* de l'élément *Subckt* spécifique. Par exemple, "PDN_pin1" défini ci-dessus (dans l'exemple 1) peut être appelé de la manière suivante:

```
<Pdn>
  ...
  <Lead Id="1" Type="Ckt">
    ...
    <Netlist>
      Xpin1 1 4 PDN_pin1
    </Netlist>
  </Lead>
  ...
</Pdn>
```

Si un ou plusieurs modèles de circuits secondaires sont définis dans des fichiers de bibliothèques externes, alors les fichiers sont référencés en utilisant la balise *Data_files* comme cela est représenté dans l'exemple 3.

EXEMPLE 3 Exemple de section *Macromodels* avec plusieurs éléments *Subckt* définis dans des fichiers de données externes.

```

<Macromodels>
  <Subckt Kind="SPICE3">
    <Data_files>
      subckt_example1.lib
      subckt_example2.lib
      ...
    </Data_files>
  </Subckt>
</Macromodels>

```

La Figure 18 présente une définition de circuit secondaire typique.

```

*Netlist file created on Fri 27 Nov 2012
*Time: 12:01:27
*PDN macro-model model of PIN1 with respect to GND1
.Subckt Pin1_PDN N1 N2
C1 N1 int1 20e-9
L1 int1 int2 9e-9
R1 int2 N2 230e-3
.ends

```

IEC

Figure 18 – Exemple de fichier de liste d'interconnexions définissant un circuit secondaire

Pour éviter toute ambiguïté, lorsque des fichiers de données externes contenant les bibliothèques de modèles SPICE sont utilisés, l'analyseur syntaxique de modèle d'émissions conduites n'analyse que les déclarations de données définies dans les mots-clés SPICE suivants: ".SUBCKT" et ".ENDS". Ces mots-clés ne sont pas sensibles à la casse [2]. S'il y a plusieurs sections ".SUBCKT", elles sont analysées comme des éléments de circuits secondaires indépendants.

7.7 Section *Validity* (validité)

7.7.1 Généralités

La section *Validity* doit être utilisée pour représenter les conditions dans lesquelles le macromodèle ICEM-CE est défini conformément à l'IEC TS 62433-1. Cette section est donnée à titre d'information uniquement. Elle est indépendante de toutes les autres sections du fichier CEML.

Le Tableau 4 donne une liste des différents mots-clés reconnus de la section *Validity*.

Tableau 4 – Définition de la section *Validity*

<i>Validity</i>
Power_supply: plage d'alimentation présentée sous la forme d'une chaîne (exigée)
Temperature_range: plage de températures dans laquelle le modèle est extrait, avec l'unité (exigée)
Frequency_range: plage de fréquences présentée sous la forme d'une chaîne avec l'unité (exigée pour les données de modèles dans le domaine fréquentiel)
Phase_origin: origine des phases pour une IA dans le domaine fréquentiel (exigée pour les données de modèles dans le domaine fréquentiel)
Time_range: plage de temps présentée sous la forme d'une chaîne avec l'unité (exigée pour les données de modèles dans le domaine temporel)
Time_origin: origine des temps pour une IA dans le domaine temporel (exigée pour les données de modèles dans le domaine temporel)

Cette définition n'est pas exhaustive et est laissée ouverte pour que d'éventuelles améliorations et évolutions y soient apportées.

Les documents relatifs au circuit intégré, tels que les fiches techniques, les rapports d'essais ou le rapport d'extraction du modèle ICEM-CE, doivent être spécifiés avec le mot-clé global *Documentation* sous la forme d'une chaîne valide qui spécifie le chemin vers les documents correspondants. Plusieurs chemins de fichiers peuvent être présentés les uns après les autres dans une liste. Se reporter à B.4 pour plus d'informations sur l'utilisation du mot-clé *Documentation*.

Toutes les autres informations spécifiques relatives à la définition du macromodèle ICEM-CE complet qui peuvent être nécessaires pour comprendre le modèle doivent être définies dans la section *Notes* sous la forme d'une chaîne valide. Se reporter à B.4 pour plus d'informations sur l'utilisation du mot-clé *Notes*.

Cette section est obligatoire et définie directement dans l'élément racine *CEmodel* comme cela est représenté dans l'exemple 1 et l'exemple 2.

EXEMPLE 1 Exemple de section *Validity* d'un macromodèle ICEM-CE dans le domaine fréquentiel:

```
<CEmodel>
...
  <Validity>
    <Power_supply>5V</Power_supply>
    <Temperature_range>[20Celsius-40Celsius]</Temperature_range>
    <Frequency_range>[1MHz - 1GHz]</Frequency_range>
    <Phase_origin>10degrees</Phase_origin>
    <Notes>Only PLL is activated</Notes>
    <Documentation>
      test_report.pdf
      ICdatasheet.pdf
    </Documentation>
  </Validity>
...
</CEmodel>
```

EXEMPLE 2 Exemple de section *Validity* d'un macromodèle ICEM-CE dans le domaine temporel:

```
<CEmodel>
...
  <Validity>
    <Power_supply>5V</Power_supply>
    <Temperature_range>[20Celsius-40Celsius]</Temperature_range>
    <Time_range>from 1us to 101us</Time_range>
    <Time_origin>1us</Time_origin>
    <Notes>Only PLL is activated</Notes>
    <Documentation>
      test_report.pdf
      ICdatasheet.pdf
    </Documentation>
  </Validity>
...
</CEmodel>
```

7.7.2 Définitions d'attributs

7.7.2.1 Power_supply

L'attribut *Power_supply* est utilisé pour définir les conditions d'alimentation pour lesquelles le macromodèle ICEM-CE est valide. Cet attribut informe l'utilisateur que le macromodèle ICEM-CE est extrait dans la plage d'alimentation spécifiée. Ce champ est obligatoire.

Il n'y a pas de format particulier pour définir la valeur de cet attribut. Il doit être facilement compréhensible pour permettre une utilisation correcte du modèle. La valeur doit être une chaîne de caractères constituée de lettres et/ou de chiffres avec les unités. Les unités valides sont présentées en A.2.5.5.

Trois exemples sont présentés ci-après.

EXEMPLE 1 La syntaxe suivante spécifie que le modèle est défini pour une tension d'alimentation de 5 V.

```
<Power_supply>5V</Power_supply>
```

EXEMPLE 2 La syntaxe suivante spécifie que les blocs numériques des données du modèle sont définis avec une tension d'alimentation de 5 V et les blocs analogiques sont définis pour une tension d'alimentation de 12 V.

```
<Power_supply>5V for digital blocks, 12V for analogue blocks</Power_supply>
```

EXEMPLE 3 Les syntaxes suivantes spécifient que le modèle est défini pour des tensions d'alimentation comprises entre 2,5 V et 18 V.

```
<Power_supply>[2.5V-18V]</Power_supply>
```

ou

```
<Power_supply>between 2.5V and 18V</Power_supply>
```

7.7.2.2 Temperature_range

L'attribut *Temperature_range* est utilisé pour définir la plage de températures dans laquelle le macromodèle ICEM-CE est extrait. Ce champ est obligatoire.

Il n'y a pas de format particulier pour définir la valeur de cet attribut. Il doit être facilement compréhensible pour permettre une utilisation correcte du modèle. La valeur doit être une chaîne de caractères constituée de lettres et/ou de chiffres avec les unités. Les unités valides sont présentées en A.2.5.5.

Deux exemples sont présentés ci-après.

EXEMPLE 1 Les syntaxes suivantes spécifient que le modèle est valide pour des températures comprises entre 20 °C et 40 °C.

```
<Temperature_range>[20Celsius-40Celsius]</Temperature_range>
```

ou

```
<Temperature_range>from 20Celsius to 40Celsius</Temperature_range>
```

EXEMPLE 2 La syntaxe suivante spécifie que le modèle est défini uniquement pour une température de 298,15 K.

```
<Temperature_range>298.15K</Temperature_range>
```

La plage de températures de validité du macromodèle ICEM-CE doit être la plage de températures commune des composants du PDN et générateurs de l'IA.

7.7.2.3 Frequency_range

L'attribut *Frequency_range* est utilisé pour définir la plage de fréquences dans laquelle le macromodèle ICEM-CE est valide. Cet attribut informe l'utilisateur que le macromodèle ICEM-CE est extrait dans le domaine fréquentiel, qu'il est spécifié dans la plage de fréquences donnée et qu'il est utilisable dans cette même plage. Ce champ est obligatoire pour un macromodèle ICEM-CE dans le domaine fréquentiel.

Il n'y a pas de format particulier pour définir la valeur de cet attribut. Il doit être facilement compréhensible pour permettre une utilisation correcte du modèle. La valeur doit être une chaîne de caractères constituée de lettres et/ou de chiffres avec les unités (voir l'exemple 1).

EXEMPLE 1 Les syntaxes suivantes spécifient que le modèle est valide pour des fréquences comprises entre 1 MHz et 1 GHz.

```
<Frequency_range>[1MHz-1GHz]</Frequency_range>
```

ou

```
<Frequency_range>from 1MHz to 1GHz</Frequency_range>
```

La plage de fréquences de validité du macromodèle ICEM-CE doit être la plage de fréquences commune des composants du PDN et générateurs de l'IA.

7.7.2.4 Phase_origin

L'attribut *Phase_origin* est utilisé pour définir l'origine des phases pour laquelle le macromodèle ICEM-CE est valide. Cette information est spécifique au sous-modèle d'IA. Ce champ est obligatoire pour un macromodèle ICEM-CE dans le domaine fréquentiel.

Il n'y a pas de format particulier pour définir la valeur de cet attribut. Il doit être facilement compréhensible pour permettre une utilisation correcte du modèle. La valeur de cet attribut est une chaîne de caractères constituée de lettres et/ou de chiffres avec les unités (voir les exemples 1 et 2).

EXEMPLE 1 La syntaxe suivante spécifie que l'origine des phases des données du modèle (IA) est 10 degrés.

```
<Phase_origin>10degrees</Phase_origin>
```

EXEMPLE 2 La syntaxe suivante spécifie que l'origine des phases des données d'IA du bloc numérique est 5 degrés et celle du bloc analogique est 22 degrés.

```
<Phase_origin>5degrees for digital block, 22 degrees for analogue block</Phase_origin>
```

7.7.2.5 Time_range

L'attribut *Time_range* est utilisé pour définir la plage de temps dans laquelle le macromodèle ICEM-CE est valide. Cet attribut informe l'utilisateur que le macromodèle ICEM-CE est extrait dans le domaine temporel, est spécifié dans la plage de temps donnée et est utilisable dans cette même plage. Ce champ est obligatoire pour un macromodèle ICEM-CE dans le domaine temporel.

Il n'y a pas de format particulier pour définir la valeur de cet attribut. Il doit être facilement compréhensible pour permettre une utilisation correcte du modèle. La valeur de cet attribut est une chaîne de caractères constituée de lettres et/ou de chiffres avec les unités (voir l'exemple 1).

EXEMPLE 1 La syntaxe suivante spécifie que le modèle est valide pour des temps compris entre 100 ps et 1 ms.

```
<Time_range>[100ps-1ms]</Time_range>
```

ou

```
<Time_range>from 100ps to 1ms</Time_range>
```

La plage de temps de validité du macromodèle ICEM-CE doit être la plage de temps commune des composants du PDN et générateurs de l'IA.

7.7.2.6 Time_origin

L'attribut *Time_origin* est utilisé pour définir l'origine des temps et le temps de cycle pour le mode opérationnel dans lequel le macromodèle ICEM-CE est valide. Cette information est spécifique au sous-modèle d'IA. Ce champ est obligatoire pour un macromodèle ICEM-CE dans le domaine temporel.

Il n'y a pas de format particulier pour définir la valeur de cet attribut. Il doit être facilement compréhensible pour permettre une utilisation correcte du modèle. La valeur de cet attribut est une chaîne de caractères constituée de lettres et/ou de chiffres avec les unités (voir l'exemple 1).

EXEMPLE 1 La syntaxe suivante spécifie que l'origine des temps des données d'IA est 100 ns.

```
<Time_origin>100ns</Time_origin>
```

EXEMPLE 2 La syntaxe suivante spécifie que l'origine des temps des données d'IA du bloc numérique est 100 ps et celle du bloc analogique est 1 µs.

<Time_origin>100ps for digital block, 1us for analogue block</Time_origin>

7.8 PDN

7.8.1 Généralités

La section *Pdn* du macromodèle ICEM-CE contient les données du PDN qui décrivent le modèle. Les données doivent être définies dans le mot-clé *Pdn* de la manière suivante:

```
<Pdn>
    PDN data
</Pdn>
```

Les données du PDN sont définies pour un circuit intégré particulier et la définition doit être donnée dans une balise *Lead*. La hiérarchie structurale est représentée à la Figure 17. Le Tableau 5 donne une liste des différents champs reconnus du mot-clé *Lead*.

Tableau 5 – Définition du mot-clé *Lead* pour la section *Pdn*

<i>Lead</i>
Id: identité de la broche présentée sous la forme d'une chaîne valide (exigée)
Ground_id: identité de la broche présentée sous la forme d'une chaîne valide (exigée si Type="S", "Z", "Y", sinon facultative)
Blockname: nom du bloc PDN présenté sous la forme d'une chaîne valide (facultatif)
Type: paramètre de la source du PDN ("S", "Z", "Y", "Ckt")
Param_order: ordre dans lequel les paramètres du PDN sont définis
Format: format de données ("RI", "MA", "DB")
Meas_type: Méthode mise en œuvre pour réaliser les mesures du PDN
Reference_impedance: impédance de référence utilisée pour les mesures du PDN
Use: paramètre utilisé spécifiquement
Netlist: définition du PDN utilisant un format normalisé de la liste d'interconnexions (exigée si Type="Ckt")
Unit_freq: définition de l'unité des fréquences
Unit_param: unité des paramètres du PDN
Power_level: niveau de puissance de mesure pendant l'extraction du PDN
Data_files: paramètre de la source du PDN défini dans un fichier externe (exigé en l'absence de List)
List: liste des paramètres du PDN (exigée en l'absence de Data_files)

Les données du PDN peuvent être définies comme un seul ou plusieurs blocs PDN (interconnectés ou non connectés). Comme cela est expliqué en 5.3.1, le PDN ou les blocs PDN peuvent être représentés en utilisant des paramètres *S*, *Y* ou *Z* à un seul port ou à plusieurs ports (paramètres du réseau) ou en utilisant des listes d'interconnexions de type SPICE [2]. Dans les deux cas, les données peuvent être définies en liste ou en utilisant des fichiers de données externes.

La plage de fréquences ou la plage de temps des informations du PDN doit être spécifiée dans la balise *Frequency_range* ou dans la balise *Time_range*, respectivement de la section *Validity*.

NOTE La plage de fréquences (ou la plage de temps) de validité du macromodèle ICEM-CE est la plage de fréquences (ou la plage de temps) commune des composants du PDN et générateurs de l'IA.

Conformément à l'IEC TS 62433-1, toutes les autres informations pertinentes nécessaires à la compréhension (et à l'utilisation) du PDN doivent être facultativement définies dans les balises *Notes* et *Documentation*. D'autres informations, telles que le mode de fonctionnement du circuit intégré, les condensateurs de découplage sur les lignes d'alimentation, les plages

d'alimentation, les fonctions activées, les conditions de charge, les informations sur la mise à la masse, les fiches techniques et les rapports d'essais, peuvent être définies. Se reporter à B.4 pour plus d'informations sur l'utilisation des mots-clés *Documentation* et *Notes*.

7.8.2 Définitions d'attributs

7.8.2.1 Id

Il convient que les identités des broches (*Id*) utilisées dans cette définition du PDN aient été définies au préalable dans la balise *Lead_definitions* comme cela est décrit en 7.5. Des broches "internes" et des broches "externes" peuvent être définies.

Ce champ est obligatoire.

7.8.2.2 Ground_id

Il convient que l'identité de la broche de retour à la masse (*Ground_id*) utilisée dans cette définition du PDN ait été définie au préalable dans la balise *Lead_definitions* comme cela est décrit en 7.5. Cette broche définit le chemin du signal de retour pour définir le PDN.

Cet attribut est défini uniquement si les paramètres du réseau sont utilisés pour définir le PDN (*Type*="S", "Z" ou "Y"). L'attribut *Ground_id* est facultatif si *Type*="Ckt", c'est-à-dire si une liste d'interconnexions est utilisée pour représenter le PDN. Voir 7.8.2.4 pour plus d'informations sur l'attribut *Type*.

Lorsqu'il est utilisé, un seul *Ground_id* unique est admis par définition de *Lead*. La borne de masse doit être la masse de référence idéale utilisée pour obtenir les paramètres du réseau comme indiqué en 5.3.2.

Ce champ est facultatif, mais il est exigé si le PDN est représenté par des paramètres du réseau.

7.8.2.3 Blockname

Le champ *Blockname* est utilisé pour définir le nom du PDN dans le cas d'une modélisation ICEM-CE à base de blocs ou à base de sous-modèles. Si le PDN et l'IA ont le même nom, ils sont regroupés par l'analyseur syntaxique de modèle d'émissions conduites en un sous-modèle.

Ce champ est facultatif.

7.8.2.4 Type

L'attribut *Type* est utilisé pour représenter le type des données du PDN. Les types valides sont:

- "S": données des paramètres *S*.
- "Z": données des paramètres *Z*. Ces paramètres ne sont pas normalisés pour l'impédance de référence.
- "Y": données des paramètres *Y*. Ces paramètres ne sont pas normalisés pour l'impédance de référence.
- "Ckt": description du circuit à l'aide de listes d'interconnexions.

Ce champ est facultatif. Si ce champ n'est pas renseigné, sa valeur par défaut est "S".

7.8.2.5 Param_order

L'attribut *Param_order* indique à l'analyseur syntaxique de modèle d'émissions conduites la façon dont les données du PDN sont représentées. Les chaînes suivantes sont dédiées à la spécification de l'ordre des paramètres:

- "Freq" et "Frequency": grandeur de fréquence utilisée pour la définition des paramètres
- "Time": grandeur de temps utilisée pour la définition des paramètres
- "Sij": paramètres *S*. *i* et *j* sont des nombres entiers représentant les ports/broches de mesure (exemple: "S11", "S21", "S31", etc.)
- "Zij": paramètres *Z*. *i* et *j* sont des nombres entiers représentant les ports/broches de mesure (exemple: "Z11", "Z21", "Z31", etc.)
- "Yij": paramètres *Y*. *i* et *j* sont des nombres entiers représentant les ports/broches de mesure (exemple: "Y11", "Y21", "Y31", etc.)
- All: tous les termes définis par rapport au nombre d'identifiants de broches (*Id*) sont pris en compte.

Les différents termes de *Param_order* doivent être séparés par une virgule (",").

Ce champ est facultatif. Il n'est pas défini si le PDN est représenté comme un modèle de circuit selon 7.8.3.4 et 7.8.3.5. Si ce champ n'est pas renseigné, sa valeur par défaut est "Freq,All".

7.8.2.6 Format

L'attribut *Format* détermine le format des données. Il n'est pas défini si le PDN est représenté comme un modèle de circuit selon 7.8.3.4 et 7.8.3.5. Des formats de données valides sont:

- "RI": format réel/imaginaire (PDN dans le domaine fréquentiel)
- "DB": amplitude en décibels avec l'angle de phase en degrés (PDN dans le domaine fréquentiel)
- "MA": amplitude à l'échelle linéaire avec l'angle de phase en degrés (PDN dans le domaine fréquentiel)
- "DBMAG": amplitude en décibels sans données de phase (PDN dans le domaine temporel)
- "MAG": données d'amplitude à l'échelle linéaire sans données de phase (PDN dans le domaine temporel)

Ce champ est facultatif. Si ce champ n'est pas renseigné, sa valeur par défaut est "RI".

7.8.2.7 Meas_type

L'attribut *Meas_type* est utilisé par l'analyseur syntaxique de modèle d'émissions conduites pour calculer l'impédance de l'identifiant de la broche spécifique (*Id*) lorsque les paramètres *S* (Type= "S") sont utilisés. La définition de l'attribut *Meas_type* dépend du type du PDN (à une seule extrémité, différentiel, à plusieurs extrémités). La version 1 du CEML prend en charge les valeurs suivantes de *Meas_type* pour un PDN à une seule extrémité avec les paramètres *S*:

- "0" pour une configuration de mesure normalisée ou conventionnelle (valeur par défaut)
- "1" lorsque le DUT et les ports de mesure sont connectés en parallèle (connexion parallèle)
- "2" lorsque le DUT et les ports de mesure sont connectés en série (connexion série)

Pour un PDN à plusieurs extrémités, la valeur de *Meas_type* est toujours "0".

Les différentes configurations de mesure et les différents montages pour une extraction du PDN à une seule extrémité doivent être conformes aux spécifications du CISPR 17. Certains aspects fondamentaux sont étudiés à l'Annexe E.

Ce champ est facultatif et est défini uniquement lorsque Type="S". Si ce champ n'est pas renseigné, sa valeur par défaut est "0".

7.8.2.8 Reference_impedance

L'attribut *Reference_impedance* est utilisé par l'analyseur syntaxique de modèle d'émissions conduites pour calculer l'impédance de l'identifiant de la broche spécifique (*Id*) lorsque les paramètres du réseau sont utilisés pour définir le PDN.

Ce champ est facultatif et est défini uniquement lorsque Type="S". S'il n'est pas défini avec une valeur numérique, la valeur par défaut est "50ohm" conformément au CISPR 17.

7.8.2.9 Use

L'attribut *Use* indique à l'analyseur syntaxique de modèle d'émissions conduites d'utiliser une des valeurs de l'attribut *Param_order*. Lorsqu'il est défini explicitement, le paramètre correspondant est utilisé comme PDN. S'il n'est pas défini, le terme "Sij", "Zij" ou "Yij" est utilisé par défaut.

Ce champ est facultatif.

7.8.2.10 Netlist

Le mot-clé *Netlist* est utilisé pour définir le PDN comme une liste d'interconnexions de type SPICE qui représente la connectivité électrique des éléments du PDN. Le schéma de connectivité électrique doit être conforme à la spécification SPICE [2]. Une description détaillée du modèle de circuit est présentée en 7.8.3.4 et en 7.8.3.5.

Ce champ est obligatoire si Type="Ckt".

7.8.2.11 Unit_freq et Unit_param

Les unités des paramètres sont définies dans les balises *Unit_freq* et *Unit_param* pour la fréquence et les données respectivement. Lorsqu'elle n'est pas renseignée, la valeur par défaut de *Unit_freq* est "Hz". Le Tableau 6 ci-dessous donne la valeur par défaut de l'attribut *Unit_param* en fonction du type et du format de données lorsqu'elle n'est pas renseignée.

Tableau 6 – Formats de données valides et unités par défaut dans la section *Pdn*

Type de données	Format de données	Valeur par défaut de Unit_param
S	RI	1
	MA ou MAG	1
	DB ou DBMAG	dB
Z	RI	ohm
	MA ou MAG	ohm
	DB ou DBMAG	dBohm
Y	RI	S
	MA ou MAG	S
	DB ou DBMAG	dBS

Les unités valides sont présentées en A.2.5.5.

Ces champs sont facultatifs.

7.8.2.12 Power_level

Le niveau de puissance de mesure utilisé pour l'extraction du PDN est spécifié dans le champ *Power_level*. Pour des raisons de simplicité, il convient de définir ensemble le niveau de puissance et les unités comme indiqué en A.2.5.3.

Ce champ est facultatif. Si ce champ n'est pas renseigné, sa valeur par défaut est "0dBm".

7.8.2.13 Data_files et List

La balise *Data_files* est utilisée pour différencier les données en liste des données externes. Si les données du PDN sont définies dans un fichier externe, le lien vers le fichier doit être spécifié dans la balise *Data_files*. Les données en liste sont définies directement dans la balise de niveau supérieur *Lead* dans une balise *List*, si nécessaire. Les fichiers possédant les extensions indiquées dans le Tableau 7 sont valides.

Tableau 7 – Extensions de fichiers valides dans la section *Pdn*

Extension de fichier	Nom commun
dat ou DAT	Fichier de données
csv ou CSV	Valeurs séparées par des virgules
txt ou TXT	Fichier de texte
snp ou SnP	Touchstone, n est un nombre entier (1,2,...)
cir ou CIR	Fichier de circuit (liste d'interconnexions)
lib ou LIB	Fichier de bibliothèque (liste d'interconnexions)
net ou NET	Fichier de liste d'interconnexions

La balise *List* n'est pas définie si un modèle de circuit est utilisé pour le modèle de PDN. La balise *Data_files* est définie pour les paramètres du réseau (*S/Z/Y*) et les définitions de *Netlist*.

Ce champ est obligatoire. Un élément unique *List* ou *Data_files* doit être utilisé pour définir les données du PDN.

7.8.3 Description

7.8.3.1 Généralités

Conformément à 6.1, le PDN du macromodèle ICEM-CE peut être un seul bloc (macromodèle de circuit intégré générique) ou être constitué de plusieurs blocs (et sous-modèles) pour une modélisation ICEM-CE à base de blocs. En fonction de l'outil de simulation utilisé, le PDN peut être représenté avec différents paramètres, typiquement les paramètres *S*, *Z* et *Y*, ou sous la forme d'un circuit/liste d'interconnexions.

Le Tableau 8 donne une liste des champs valides d'une structure *Lead* pour la définition d'un PDN et leur utilisation.

Tableau 8 – Champs valides du mot-clé *Lead* dans la section *Pdn*

Champ	Description	Utilisation	Règles
Id	identifiant ou numéro de broche	Exigée	Un ou plusieurs identifiants, nœuds internes ou externes, définis au préalable dans la section <i>Lead_definitions</i>
Ground_id	Identifiant de la broche de masse	Exigée si Type="S", "Z" ou "Y"	Un identifiant unique défini au préalable dans la section <i>Lead_definitions</i>
Blockname	Nom du bloc PDN	Facultative	Chaîne valide (voir 7.8.1)
Type	Paramètre de la source du PDN (S, Z, Y, circuit)	Facultative	Chaîne valide: "S" (par défaut), "Z", "Y" ou "Ckt"
Param_order	Ordre dans lequel les paramètres sont définis	Facultative	Chaîne valide (voir 7.8.1)
Format	Format de données	Facultative	"RI" (par défaut), "DB" ou "MA"
Meas_type	Méthode mise en œuvre pour réaliser les mesures du PDN	Facultative	"0", "1" ou "2"
Reference_impedance	Impédance de référence utilisée pour les mesures du PDN	Facultative	Valeur numérique valide avec unité: "50ohm"
Use	Paramètre utilisé spécifiquement	Facultative	Une des valeurs dans <i>Param_order</i> autre que le terme <i>Frequency</i> . Valeur par défaut: terme S, Z ou Y
Netlist	Liste d'interconnexions de type SPICE	Facultative	Voir 7.8.3.4 et 7.8.3.5
Unit_freq	Unité de fréquence	Facultative	Voir A.2.5.5
Unit_param	Unité de paramètre	Facultative	Voir A.2.5.5
Power_level	Niveau de puissance de mesure	Facultative	Voir A.2.5.2 et A.2.5.3
Data_files	Paramètre de la source du PDN défini dans un fichier externe	Exigée si différent de <i>List</i>	Valide pour un fichier de liste d'interconnexions externe ou un fichier de paramètres de réseau externe
List	Données du PDN sous la forme d'une liste	Exigée si différent de <i>Data_files</i>	Valeurs numériques valides avec ou sans unité dans l'ordre spécifié

La composition des champs *Id*, *Type*, *Param_order*, *Meas_type*, *Reference_impedance* et *Format* a été présentée en 7.8.1. Pour un PDN à une seule extrémité, *Id* est une chaîne unique et pour un PDN à plusieurs extrémités, *Id* contient les différents identifiants (*Id*) des broches séparés par une virgule (","). L'exemple 1 présente un PDN à une seule extrémité et l'exemple 2 présente un PDN à plusieurs extrémités.

EXEMPLE 1 PDN à une seule extrémité avec une broche "1" par rapport à la broche de masse "5", défini en utilisant les paramètres *S*:

```
<Lead Id="1" Ground_id="5" Type="S".../>
```

EXEMPLE 2 PDN à une plusieurs extrémités avec les broches "1", "2", "3", "4", etc. par rapport à la broche de masse "5", défini en utilisant les paramètres *S*:

```
<Lead Id="1,2,3,4,..." Ground_id="5" Type="S" ... />
```

Pour les macromodèles ICEM-CE à base de blocs ou à base de sous-modèles, le nom du sous-bloc PDN est spécifié en utilisant le champ *Blockname*. Ce champ est donné uniquement à titre d'information. L'analyseur syntaxique de modèle d'émissions conduites n'interprète pas ce champ. Les sous-modèles de PDN, d'IBC et d'IA partagent le même espace de noms. Donc, si un PDN (et un IBC) et une IA ont la même valeur (chaîne de nom) dans leur élément *Blockname* respectif, ils sont regroupés en un seul composant de sous-modèle ICEM-CE. Un exemple de PDN dans un macromodèle ICEM-CE à base de blocs est représenté dans l'exemple 3.

EXEMPLE 3 PDN à une seule extrémité représentant "Digital_block_A":

```
<Lead Id="1" Blockname="Digital_block_A".../>
```

Les données peuvent être définies directement dans le fichier CEML ou elles peuvent être définies en utilisant des fichiers de données externes pour la définition des paramètres du réseau et pour la définition du circuit (liste d'interconnexions).

7.8.3.2 Données en liste – paramètres du réseau

Une pratique courante consiste à extraire les paramètres du réseau, tels que les paramètres S , Y ou Z , et à les définir dans le domaine fréquentiel. S'ils sont définis pour une seule fréquence, ils peuvent être définis directement dans le mot-clé *Lead*. Si le PDN est défini pour plusieurs fréquences, alors les données doivent être écrites dans le mot-clé *List*. Il est également possible de représenter les paramètres S dans le domaine temporel.

Les différentes broches (internes et externes), dont le PDN doit être défini, doivent être définies en utilisant l'attribut *Id* et la broche de masse (broche dont *Mode*="GND" dans la section *Lead_definitions*) utilisée comme chemin du signal de retour (masse de référence idéale) doit être spécifiée en utilisant l'attribut *Ground_id*.

Les unités de fréquence doivent être définies dans le mot-clé *Unit_freq* et les unités de données doivent être spécifiées en utilisant le mot-clé *Unit_param*. Le niveau de puissance utilisé pour la mesure doit être défini en utilisant la balise *Power_level* et l'impédance de référence en utilisant la balise *Reference_impedance*. Voir 7.8.2 pour plus d'informations ces attributs et leurs valeurs par défaut.

NOTE Il n'est pas nécessaire que les valeurs des identifiants (*Id*) des broches soient cohérentes avec les numéros des ports utilisés pour les mesures. Les valeurs sont mises en correspondance en séquence avec les numéros des ports, de gauche à droite, dans le format Touchstone.

L'exemple 1 représente un PDN à plusieurs extrémités des broches "1", "2", "3" et "4" par rapport à la broche de masse "5". Les données des paramètres S (dans le domaine fréquentiel), extraites avec une puissance de "10dBm", sont définies au format réel/imaginaire et le champ *Param_order* est défini comme "Freq,All", ce qui signifie que tous les paramètres S sont définis en séquence (format Touchstone): Fréquence, $\text{Re}\{S_{11}\}$, $\text{Im}\{S_{11}\}$, $\text{Re}\{S_{12}\}$, $\text{Im}\{S_{12}\}$, $\text{Re}\{S_{13}\}$, $\text{Im}\{S_{13}\}$, $\text{Re}\{S_{14}\}$, $\text{Im}\{S_{14}\}$, etc.

EXEMPLE 1 Exemple de PDN à plusieurs extrémités dans le domaine fréquentiel.

```
<Pdn>
  <!-- S Parameters inline data, multi-frequencies -->
  <Lead Id="1,2,3,4";Ground_id="5" Type="S"; Param_order="Freq,All"; Format="RI">
    <Unit_freq>kHz</Unit_freq>
    <Unit_param>1</Unit_param>
    <Power_level>10dBm</Power_level>
    <List>
100 0.004 0.002 0.0018 0.0023 -0.0013 -0.0021 0.99 -0.0037 0.0019 0.002 0.0042 0.0024
0.998 -0.0038 -0.0011 -0.002 -0.0013 -0.0021 0.998 -0.0037 0.0041 0.0027 0.0013
0.0026 0.998 -0.0036 -0.001 -0.0019 0.0013 0.0027 0.0022 0.0027
...
    </List>
  </Lead>
</Pdn>
```

7.8.3.3 Données de fichiers externes – paramètres du réseau

Pour indiquer à l'analyseur syntaxique de modèle d'émissions conduites d'utiliser le fichier de données externe nécessaire dans le format ASCII, la définition doit être faite en utilisant le mot-clé *Data_files*. Si le fichier externe a une extension Touchstone (*.snp ou *.SnP, où n est un nombre entier 1, 2, ...), les définitions de *Unit_param*, *Unit_freq*, *Param_order*, *Reference_impedance* et *Type* ne sont pas nécessaires (ignorées si cela est explicitement défini). Pour toutes les autres extensions de fichier (*.dat, *.csv, *.txt), les valeurs définies sont utilisées. Si aucune valeur n'est renseignée, les valeurs par défaut sont automatiquement utilisées. Les propriétés par défaut sont indiquées dans le Tableau 8. Comme pour la définition en liste présentée en 7.8.3.2, le champ *Ground_id* est obligatoire et le champ *Meas_type* est facultatif.

Il convient que le niveau de puissance de mesure fasse toujours partie du fichier externe sous la forme d'un commentaire. Les lignes de commentaires d'un fichier PDN doivent commencer par un point d'exclamation ("!"), comme dans l'exemple 1 ci-dessous.

EXEMPLE 1 !This is a comment line

La Figure 19 est un exemple de fichier de paramètres S à quatre ports au format Touchstone, dans le domaine fréquentiel, avec data au format dB. Ce fichier est utilisé pour définir le PDN des broches "1", "2", "3" et "4" par rapport à la broche de masse "5" (voir l'exemple 2). Le modèle est basé sur des blocs avec un nom défini.

NOTE Le niveau de puissance de mesure est inclus sous la forme d'un commentaire dans le fichier PDN.

```
!Agilent Technologies,E5071C,MY46100659,A.07.02
!Date: Wed Jan 23 09:49:50 2013
!Data & Calibration Information:
!Freq
!Power_level: 10dBm
# Hz S RI R SO
100000 0.004274546 0.00241179 0.001886209 0.002311387 -0.001362228 -0.002127369 0.9983465 -0.003734921
0.00190445 0.00233215 0.004235411 0.002405719 0.998264 -0.003837676 -0.001106818 -0.002012462
-0.001369703 -0.002143684 0.9981545 -0.003736369 0.004101523 0.002755493 0.001347644 0.002668388
0.9983041 -0.003624817 -0.001178987 -0.001959733 0.001348257 0.002710212 0.002291472 0.002748889
100679 0.00422729 0.002403554 0.001871881 0.002334189 -0.001349262 -0.002154236 0.998417 -0.003759051
0.001902756 0.002356657 0.004170153 0.002522967 0.9982947 -0.003870883 -0.001127652 -0.002029074
-0.001356177 -0.002142675 0.9980684 -0.003715199 0.00410353 0.002777282 0.001371339 0.002686834
0.9982006 -0.003591272 -0.001147017 -0.001990634 0.001357759 0.002703257 0.002186219 0.002865791
101359 0.004252984 0.002479831 0.001877719 0.002357767 -0.001379005 -0.002139817 0.9983823 -0.00378314
0.001916786 0.002348817 0.004238612 0.002526162 0.9983067 -0.003854939 -0.001140214 -0.002014498
-0.00136173 -0.002137753 0.9980044 -0.003766839 0.004121085 0.002838985 0.001346626 0.00268385
0.9981698 -0.003693599 -0.001171574 -0.00198756 0.001361229 0.00271512 0.002269425 0.002912863
```

IEC

Figure 19 – PDN représenté par des paramètres S au format Touchstone

EXEMPLE 2 Description d'un PDN en utilisant un fichier de données externe

```
<Pdn>
  <Lead Id="1,2,3,4;Ground_id="5">
    <Power_level>0dBm</Power_level>
    <Blockname>memory_block</Blockname>
    <Data_files>
      Pin1234_PDN.s4p
    </Data_files>
  </Lead>
</Pdn>
```

7.8.3.4 Définition de circuit en liste

Une autre manière de définir le PDN consiste à utiliser un schéma (de circuit). L'analyseur syntaxique de modèle d'émissions conduites comprend que le PDN est défini par un circuit lorsque l'attribut *Type* d'une broche ou d'un port particulier est "Ckt". La définition de circuit doit être directement déclarée dans la balise *Netlist* de la section *Lead*. La section *Netlist* doit contenir les informations suivantes:

- Définition de *Power_level*
- Déclaration de données de type SPICE définissant les connexions électriques des composants [2]

Tous les éléments reconnus par SPICE peuvent être définis dans cette section et la représentation doit suivre la syntaxe de SPICE. Les déclarations de données sont présentées dans un format libre et sont constituées de champs séparés par un espace. Pour continuer une déclaration à la ligne suivante, utiliser le signe "+" (signe de continuation) au début de la ligne suivante. Les nombres peuvent être des nombres entiers ou des nombres à virgule flottante [2].

EXEMPLE 1 Une résistance R1 de 1 k Ω raccordant des nœuds "In" et "Out" peut être représentée de la manière suivante:

R1 In Out 1e3

Tous les nœuds utilisés dans les déclarations de données partagent le même espace de noms avec les identifiants de broches définis dans la section *Lead_definition*. Le champ *Ground_id* est facultatif (s'il est défini, il est ignoré par l'analyseur syntaxique de modèle d'émissions conduites) lors de la définition du PDN en utilisant une liste d'interconnexions. L'attribut *Id* du mot-clé *Lead* doit porter l'information sur l'identifiant des broches internes, externes et de masse (comme cela est défini dans la section *Lead_definitions*).

Dans plusieurs déclarations de données utilisées pour définir la liste d'interconnexions, au moins un nœud doit correspondre à la broche définie comme interne ou externe et au moins un nœud doit être GND dans la section *Lead_definition*. Tous les autres nœuds intérieurs utilisés pour la définition du circuit ne doivent pas porter l'identifiant de la broche dans la section *Lead_definition*. Pour assurer la compatibilité entre plusieurs types SPICE, "0" n'est pas autorisé comme numéro de nœud.

La définition du mot-clé *Netlist* est représentée dans le Tableau 9.

Tableau 9 – Définition de Netlist

<i>Netlist</i>
Kind: format de liste d'interconnexions SPICE (facultatif), valeur par défaut: "SPICE3"
Data_files: liste d'interconnexions SPICE dans un fichier externe (facultative)

L'attribut facultatif *Kind* indique à l'analyseur syntaxique de modèle d'émissions conduites que le circuit secondaire défini respecte une syntaxe spécifique. La version 1 du langage CEML prend en charge les syntaxes de liste d'interconnexions de type SPICE conformes à une norme industrielle.

Si le champ n'est pas renseigné, la valeur par défaut du champ *Kind* est la valeur générique "SPICE3".

L'exemple 1 est un exemple de PDN utilisant une liste d'interconnexions de type SPICE. Le même PDN peut également être exprimé comme un macromodèle SPICE, voir l'exemple 2. Dans l'exemple 2, par hypothèse "PDN_pin12345" est défini dans la section *Macromodels* (voir 7.6).

EXEMPLE 1 Exemple de PDN utilisant une définition de liste d'interconnexions en liste. Dans ce cas, une broche dont *Id*="5" est utilisée comme broche de masse.

```
<Pdn>
  <!-- General circuit model -->
  <Lead Id="1,2,3,4,5" Type="Ckt">
    <Power_level>10dBm</Power_level>
    <Netlist Kind="SPICE3">
```

```

L1 1 n00 1.820e-09
R1 n01 n02 2.408e-01
C1 n00 5 1e-12
L2 2 n10 1.828e-09
R2 n11 n12 2.434e-01
C2 n10 5 1e-12
...
</Netlist>
</Lead>
</Pdn>

```

EXEMPLE 2 Exemple de PDN utilisant des macromodèles SPICE. Dans ce cas, une broche dont Id="5" est utilisée comme broche de masse. XPDN1 est l'identifiant du circuit secondaire dans le circuit principal.

```

<Pdn>
<!-- General circuit model using macro-model-->
<Lead Id="1,2,3,4,5" Type="Ckt">
  <Power_level>10dBm</Power_level>
  <Netlist>
    XPDN1 1 2 3 4 5 PDN_pin12345
  </Netlist>
</Lead>
</Pdn>

```

7.8.3.5 Fichier externe contenant la liste d'interconnexions

A la place de la définition de la liste d'interconnexions en liste (voir 7.8.3.4), il est possible de définir le PDN dans un fichier de liste d'interconnexions externe. La définition est faite avec la balise *Data_files* dans la section *Netlist*. Toutes les règles concernant les numéros des nœuds utilisés dans la définition de la liste d'interconnexions sont les mêmes que celles définies en 7.8.3.4.

Dans certains cas, un fichier de liste d'interconnexions externe peut contenir la définition du circuit principal et les définitions des circuits secondaires. Pour éviter toute ambiguïté, l'analyseur syntaxique de modèle d'émissions conduites analyse uniquement les déclarations de données définies à l'extérieur des mots-clés SPICE: ".SUBCKT" et ".ENDS". Ces mots-clés ne sont pas sensibles à la casse. Pour analyser la syntaxe de l'élément de circuit secondaire, il est obligatoire de définir le même fichier dans la section *Macromodels* (voir 7.6).

L'exemple 1 représente l'utilisation d'un fichier de liste d'interconnexions externe et le PDN est défini comme un bloc.

EXEMPLE 1 Exemple de PDN utilisant un fichier externe contenant la définition de liste d'interconnexions. Dans ce cas, une broche dont Id="5" est utilisée comme broche de masse.

```

<Pdn>
<!-- General circuit model using netlist -->
<Lead Id="1,2,3,4,5" Type="Ckt">
  <Power_level>0dBm</Power_level>
  <Blockname>block2</Blockname>
  <Netlist>
    <Data_files>Pin12345_PDN_Ckt.lib</Data_files>
  </Netlist>
</Lead>
</Pdn>

```

7.9 IBC

7.9.1 Généralités

Comme pour le PDN, différents niveaux de complexité d'IBC peuvent être considérés. Le niveau de configuration le plus simple est la modélisation de l'IBC avec deux bornes internes reliant deux bornes internes de différents blocs PDN. La configuration d'un IBC réalisant l'interface entre de nombreux blocs PDN et de nombreuses bornes internes peut devenir très complexe.

La section *Ibc* du macromodèle ICEM-CE contient les données de l'IBC qui décrivent le PDN d'un ICEM-CE. Les données doivent être définies dans le mot-clé *Ibc* de la manière suivante:

```
<Ibc>
  IBC data
</Ibc>
```

Les données d'un IBC sont définies pour réaliser l'interface entre différents blocs PDN et la définition doit être donnée dans une balise *Lead* (semblable au PDN). La hiérarchie structurelle est représentée à la Figure 17. Plusieurs éléments *Lead* peuvent être présentés les uns après les autres dans une liste dans la section *Ibc*.

7.9.2 Définitions d'attributs

La structure générale du mot-clé *Lead* est cohérente avec celle donnée dans le Tableau 5. Toutefois, certaines spécifications d'attributs sont différentes pour le réseau IBC. Ces différences sont présentées dans le Tableau 10. Tous les autres champs du Tableau 5 restent inchangés.

Tableau 10 – Différences entre les champs des sections *Pdn* et *Ibc*

Champ	Description	Utilisation		Règles	
		PDN	IBC	PDN	IBC
Id	identifiant ou numéro de broche	Exigée	Exigée	Un ou plusieurs <u>identifiants</u> définis au préalable dans la section <i>Lead_definitions</i> comme des <u>bornes externes ou internes</u>	Deux <u>identifiants</u> ou plus définis au préalable dans la section <i>Lead_definitions</i> comme des <u>bornes internes</u>
Ground_id	Identifiant ou numéro de broche de la broche du signal de retour	Exigée pour Type="S", "Z" ou "Y"	NA	Un identifiant unique défini au préalable comme GND dans la section <i>Lead_definitions</i>	NA

Le Tableau 11 donne une liste des champs valides d'une structure *Lead* pour la définition d'un IBC et leur utilisation.

Tableau 11 – Champs valides du mot-clé *Lead* pour la définition d'un IBC

Champ	Description	Utilisation	Règles
Id	identifiant ou numéro de broche	Exigée	Deux identifiants ou plus définis au préalable dans la section <i>Lead_definitions</i> comme des bornes internes
Blockname	Nom du composant de l'IBC	Facultative	Chaîne valide (voir 7.8.1)
Type	Paramètre de la source de l'IBC (S, Z, Y, circuit)	Facultative	Chaîne valide: "S" (par défaut), "Z", "Y" ou "Ckt"
Param_order	Ordre dans lequel les paramètres de l'IBC sont définis	Facultative	Chaîne valide (voir 7.8.1)
Format	Format de données	Facultative	"RI" (par défaut), "DB" ou "MA"

Champ	Description	Utilisation	Règles
Meas_type	Méthode mise en œuvre pour réaliser les mesures de l'IBC	Facultative "0"	Valeur numérique valide: "0", "1" ou "2" Valeur par défaut: "0"
Reference_impedance	Impédance de référence utilisée pour les mesures de l'IBC	Facultative	Valeur numérique valide avec unité: "50ohm"
Use	Paramètre utilisé spécifiquement	Facultative	Une des valeurs dans <i>Param_order</i> autre que le terme <i>Frequency</i> . Valeur par défaut: terme S, Z ou Y
Netlist	Liste d'interconnexions de type SPICE	Facultative	Voir 7.8.3.4 et 7.8.3.5
Unit_freq	Unité de fréquence	Facultative	Voir A.2.5.5
Unit_param	Unité de paramètre	Facultative	Voir A.2.5.5
Power_level	Niveau de puissance de mesure	Facultative	Voir A.2.5.2 et A.2.5.3
Data_files	Paramètre de la source de l'IBC défini dans un fichier externe	Exigée si différent de <i>List</i>	Valide pour un fichier de liste d'interconnexions externe ou un fichier de paramètres de réseau externe
List	Données de l'IBC sous la forme d'une liste définie en ligne	Exigée si différent de <i>Data_files</i>	Valeurs numériques valides avec ou sans unité dans l'ordre spécifié

Toutes les autres informations pertinentes nécessaires à la compréhension (et à l'utilisation) de l'IBC doivent être facultativement définies dans les balises *Notes* et *Documentation*.

7.10 IA

7.10.1 Généralités

La section *Ia* du macromodèle ICEM-CE contient les données de l'IA qui décrivent le modèle. Les données doivent être définies dans le mot-clé *Ia* de la manière suivante:

```
<Ia>
  IA data
</Ia>
```

Comme pour la définition d'un PDN et d'un IBC, les données d'une IA sont définies pour une broche ou une borne de circuit intégré et la définition doit être donnée dans une balise *Lead* (la hiérarchie de la structure est présentée à la Figure 17). Le Tableau 12 donne une liste des différents champs reconnus du mot-clé *Lead* dans la section *Ia*.

Tableau 12 – Définition du mot-clé *Lead* pour la section *Ia*

<i>Lead</i>
Id: identité de la broche présentée sous la forme d'une chaîne valide (exigée)
Type: paramètre d'une source d'IA. "time" pour les données du domaine temporel et "freq" pour les données du domaine fréquentiel (exigé)
Blockname: nom de bloc IA présenté sous la forme d'une chaîne valide (facultatif)
Voltage: données d'IA définies comme une source de tension (exigées s'il ne s'agit pas d'un courant)
Current: données d'IA définies comme une source de courant (exigées s'il ne s'agit pas d'une tension)

Les conditions de validité (plage de fréquences ou de temps, conditions d'alimentation, origine des temps ou des phases, etc.) du sous-modèle d'IA doivent être spécifiées dans la section *Validity* (voir 7.7).

NOTE La plage de fréquences (ou la plage de temps) de validité du macromodèle ICEM-CE est la plage de fréquences (ou la plage de temps) commune du PDN et de l'IA.

Conformément à l'IEC TS 62433-1, toutes les autres informations pertinentes nécessaires à la compréhension (et à l'utilisation) de l'IA doivent être facultativement définies dans les balises *Notes* et *Documentation*. D'autres informations, telles que le mode de fonctionnement du circuit intégré, les condensateurs de découplage sur les lignes d'alimentation, les fonctions activées, les informations sur la mise à la masse, les fiches techniques et les rapports d'essais, peuvent être définies.

7.10.2 Définitions d'attributs

7.10.2.1 Id

Il convient que l'identité (*Id*) des broches utilisées dans la définition de l'IA ait été définie au préalable comme des broches internes dans la balise *Lead_definitions* comme cela est décrit en 7.5. Seuls deux identifiants sont admis par définition de broche puisque la source d'IA est toujours connectée entre deux bornes. Par hypothèse, la borne ayant le premier identifiant est le nœud positif et la borne ayant le deuxième identifiant est le nœud négatif. Le courant circule du nœud positif vers le nœud négatif.

Ce champ est obligatoire.

7.10.2.2 Type

L'attribut *Type* est utilisé pour représenter le type des données de l'IA. La version 1 du langage CEML prend en charge deux types: "time" pour les données d'IA dans le domaine temporel et "freq" pour les données d'IA dans le domaine fréquentiel.

Ce champ est obligatoire.

7.10.2.3 Blockname

Le champ *Blockname* est utilisé pour définir le nom du bloc IA. Ce champ est obligatoire pour l'IA. Si le PDN et l'IA ont le même nom de bloc, ils sont regroupés par l'analyseur syntaxique de modèle d'émissions conduites en un sous-bloc.

Ce champ est facultatif.

7.10.2.4 Voltage et Current

La principale source d'IA est définie dans les sections *Voltage* et/ou *Current*.

Au moins une définition (champ *Voltage* ou champ *Current*) est exigée dans la section *Ia*. Les données de la tension sont définies dans le mot-clé *Voltage* et les données du courant sont définies dans le mot-clé *Current*.

La définition commune des sections *Voltage* et *Current* est donnée dans le Tableau 13.

Tableau 13 – Définition de Voltage et Current

Voltage / Current
List: liste des données d'IA (exigée en l'absence de Data_files ou Pulse)
Data_files: paramètre de la source de l'IA défini dans un fichier externe (exigé en l'absence de List ou Pulse)
Pulse: données d'IA représentées par une forme d'onde simulée (exigées en l'absence de List ou Data_files, valide uniquement si Type="time")
Param_order: ordre dans lequel les paramètres de l'IA sont définis (facultatif)
Format: format de données d'IA (facultatif)
Unit_voltage: unité des termes de tension (facultative)
Unit_current: unité des termes de courant (facultative)
Unit_freq: unité des termes de fréquence (facultative)
Unit_time: unité des termes de temps (facultative)

Ces sections exigent un des éléments enfants suivants: *List*, *Data_files* ou *Pulse*. Un de ces éléments doit être utilisé pour définir les données de l'IA. Tous les autres attributs et éléments du Tableau 13 sont facultatifs. Si elles ne sont pas renseignées, les valeurs et unités par défaut sont utilisées.

7.10.2.5 Data_files et List

Les définitions des mots-clés *Data_files* et *List* sont semblables à celles d'un PDN (voir 0). La balise *Data_files* est utilisée pour différencier les données en liste des données externes. Si les données du PDN sont définies dans un fichier externe, le lien vers le ou les fichiers doit être spécifié dans la balise *Data_files*. Les données en liste sont définies directement dans la balise de niveau supérieur *Voltage* ou *Current* dans une balise *List*, si nécessaire. Les fichiers possédant les extensions indiquées dans le Tableau 14 sont autorisés pour définir l'IA.

Tableau 14 – Extensions de fichiers valides dans la section Ia

Extension de fichier	Nom commun
dat ou DAT	Fichier de données
csv ou CSV	Valeurs séparées par des virgules
txt ou TXT	Fichier de texte

7.10.2.6 Pulse

Ce mot-clé est dédié à la représentation du sous-modèle d'IA en utilisant une forme d'onde simulée, telle qu'une forme d'onde triangulaire ou impulsionnelle. Ceci ne s'applique qu'à la définition d'une IA dans le domaine temporel, c'est-à-dire Type="time" (voir 7.10.2.2). La section *Pulse* est définie directement dans la balise de niveau supérieur *Voltage* ou *Current*.

Cette section est obligatoire si aucun élément *List* ou *Data_files* n'a été utilisé pour définir les données de tension ou de courant de l'IA. Le Tableau 15 donne une liste des différents champs reconnus du mot-clé *Pulse* dans la section *Voltage* ou *Current*.

Tableau 15 – Définition du mot-clé *Pulse* dans la section *Voltage* ou *Current*

<i>Pulse</i>
Initial_value: amplitude initiale ou minimale (exigée)
Final_value: amplitude finale ou maximale (exigée)
Period: période de temps du signal (exigée en l'absence de Frequency)
Frequency: fréquence du signal (exigée en l'absence de Period)
Time_delay: retard avant le front avant du signal (facultatif), valeur par défaut: "0"
Rise_time: durée nécessaire pour que le signal passe de la valeur initiale à la valeur finale (facultative), valeur par défaut: "0"
Fall_time: durée nécessaire pour que le signal passe de la valeur finale à la valeur initiale (facultative), valeur par défaut: "0"
On_time: durée pendant laquelle le signal maintient sa valeur finale (facultative), valeur par défaut: une demi-période
Ncycles: nombre de cycles de répétition (facultatif), valeur par défaut: l'infini

Les trois champs suivants sont obligatoires dans une section *Pulse*: *Initial_value*, *Final_value* et *Period* ou *Frequency* (inverse de la période). Tous les autres attributs du Tableau 15 sont facultatifs. La valeur de tous les champs est une valeur numérique avec ou sans unité. Les unités valides sont présentées en A.2.5.5.

Les unités de base de ces champs sont décrites au Tableau 16. Les facteurs de mise à l'échelle valides sont présentés en A.2.5.5. Des unités peuvent également être définies pour les paramètres en utilisant les mots-clés définis en 7.10.2.9.

Tableau 16 – Unités de base des champs de la section *Pulse*

Attribut	Unité de base	
	Section <i>Voltage</i>	Section <i>Current</i>
Initial_value	V	A
Final_value	V	A
Period	s	
Frequency	Hz	
Time_delay	s	
Rise_time	s	
Fall_time	s	
On_time	s	
Ncycles	1	

Une forme d'onde typique définie par le mot-clé *Pulse*, avec les champs correspondants est représentée à la Figure 20.

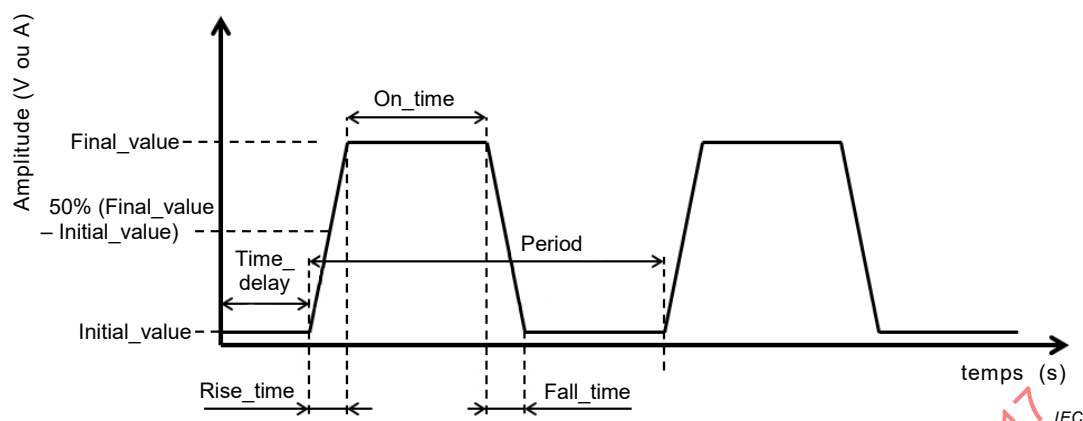


Figure 20 – Forme d'onde d'IA simulée avec les paramètres correspondants

Un exemple de données d'IA utilisant une balise *Pulse* est donné ci-dessous.

EXEMPLE Une section *Pulse* peut être définie comme suit:

```
<Ia>
  <Lead Id="1,3" Type="time">
    <Current>
      <Pulse>
        <Initial_value>0</Initial_value>
        <Final_value>50e-3</Final_value>
        <Period>125n</Period>
        <Rise_time>2n</Rise_time>
        <Fall_time>4n</Fall_time>
        <On_time>0.1n</On_time>
      </Pulse>
    </Current>
  </Lead>
</Ia>
```

7.10.2.7 Param_order

L'attribut *Param_order* indique à l'analyseur syntaxique de modèle d'émissions conduites la façon dont les données d'IA sont représentées. Les chaînes suivantes sont dédiées à la spécification de l'ordre des paramètres:

- "Freq" et "Frequency": grandeur de fréquence utilisée pour la définition des paramètres
- "Time": grandeur de temps utilisée pour la définition des paramètres
- "Volt" et "Voltage": tension utilisée pour la définition des paramètres
- "Curr" et "Current": courant utilisé pour la définition des paramètres

Si aucune valeur n'est renseignée:

- si Type="freq" alors la valeur par défaut est "Freq,Volt" dans la section *Voltage* et "Freq,Curr" dans la section *Current*,
- si Type="time", alors la valeur par défaut est "Time,Volt" dans la section *Voltage* et "Time,Curr" dans la section *Current*.

7.10.2.8 Format

L'attribut *Format* détermine le format des données. Des formats de données valides pour une IA sont:

- "RI": format réel/imaginaire. Ce format est valide pour les IA dans le domaine fréquentiel.
- "DB": amplitude en décibels avec l'angle de phase en degrés. Ce format est valide pour les IA dans le domaine fréquentiel.

- "MA": amplitude à l'échelle linéaire avec l'angle de phase en degrés. Ce format est valide pour les IA dans le domaine fréquentiel.
- "DBMAG": amplitude en décibels sans information sur la phase. Ce format est valide pour les IA dans le domaine fréquentiel et dans le domaine temporel.
- "MAG": amplitude à l'échelle linéaire sans information sur la phase. Ce format est valide pour les IA dans le domaine fréquentiel et dans le domaine temporel.

Ce champ est facultatif. Si ce champ n'est pas renseigné, sa valeur par défaut est "MAG".

7.10.2.9 Unit_freq, Unit_time, Unit_voltage et Unit_current

Les unités des paramètres sont définies dans les balises *Unit_freq*, *Unit_time*, *Unit_voltage* et *Unit_current* pour les grandeurs de fréquence, de temps, de tension et de courant respectivement. En fonction de l'élément parent, seules les unités correspondantes sont considérées, c'est-à-dire *Unit_voltage* est autorisé dans la section *Voltage* et ainsi de suite. Seuls les attributs *Unit_freq* et *Unit_time* sont définis pour plusieurs sections.

Lorsqu'elle n'est pas renseignée, la valeur par défaut de *Unit_freq* est "Hz" et la valeur par défaut de *Unit_time* est "s". Les valeurs par défaut des balises *Unit_voltage* et *Unit_current* en fonction du format de données sont données dans le Tableau 17.

Tableau 17 – Formats de données valides et unités par défaut pour les éléments *Voltage* et *Current*

Paramètre	Format de données	Valeur par défaut de Unit_voltage	Valeur par défaut de Unit_current
Voltage	RI	V	-
	MA ou MAG	V	-
	DB ou DBMAG	dBV	-
Current	RI	-	A
	MA ou MAG	-	A
	DB ou DBMAG	-	dBA

Si des valeurs complexes sont utilisées pour représenter l'IA (Format="MA" ou "DB"), l'unité par défaut des phases est "degrees". Les unités valides sont présentées en A.2.5.5.

7.10.3 Description

7.10.3.1 Données en liste

Si les données d'IA (tension et courant) sont définies pour une seule fréquence, elles peuvent être définies directement dans le mot-clé de la section correspondante (*Voltage* et *Current*). Pour les IA à plusieurs fréquences, les données doivent être écrites dans le mot-clé *List* de la section *Current* ou *Voltage* correspondante. Les unités de fréquence doivent être définies dans le mot-clé *Unit_freq*, les unités de temps dans le mot-clé *Unit_time*, les unités de tension doivent être spécifiées dans le mot-clé *Unit_voltage* et les unités de courant dans le mot-clé *Unit_current*. Si aucune valeur n'est renseignée, les valeurs par défaut sont utilisées (voir 7.10.2.9).

L'exemple 1 représente les données d'IA d'un macromodèle ICEM-CE dans le domaine fréquentiel en utilisant la définition des données en liste. Une grandeur de courant est utilisée pour définir l'IA au format décibels ("DB"). Le courant est spécifié en "dBuA" et les grandeurs de fréquences sont définies en "kHz".

EXEMPLE 1 Exemple de section *Ia* en langage CEML.

```

<Ia>
  <Lead Id="10,11" Type="freq">
    <Current Param_order ="Freq,Current" Format="DB">
      <Unit_freq>kHz</Unit_freq>
      <Unit_current>dBuA</Unit_current>
      <List>
        7.700018e+03      -2.207281e+01
        7.800018e+03      -1.900327e+01
        7.900019e+03      -1.330357e+01
        8.000019e+03      6.099079e+01
        8.100019e+03      -1.338519e+01
        8.200019e+03      -1.912021e+01
        8.300020e+03      -2.216923e+01
        ...
      </List>
    </Current>
  </Lead>
</Ia>

```

7.10.3.2 Données de fichier externe

Pour indiquer à l'analyseur syntaxique de modèle d'émissions conduites d'utiliser le fichier de données d'IA externe nécessaire dans le format ASCII, la définition doit être faite en utilisant un mot-clé *Data_files* dans la section *Voltage* ou *Current*. Toutes les autres règles sont identiques à celles de la définition des données en liste de 7.10.3. Noter que toutes les lignes de commentaires, le cas échéant, commencent par un point d'exclamation "!". Elles ne sont pas analysées par l'analyseur syntaxique de modèle d'émissions conduites.

8 Exigences relatives à l'extraction de paramètres

8.1 Généralités

Les paramètres des modèles d'ICEM-CE peuvent être extraits à partir d'informations de conception ou de mesures. La méthodologie détaillée des extractions de paramètres de modèles n'est pas couverte par la présente norme. Cet article fournit les exigences de base pour les extractions de paramètres de modèles d'après des mesures.

NOTE L'Annexe A donne des exemples d'extraction de paramètres à partir des informations de conception et de mesures.

8.2 Contraintes d'extraction liées à l'environnement

Les extractions de paramètres de macromodèles d'ICEM doivent être réalisées dans des conditions normales de température ambiante: $23\text{ °C} \pm 5\text{ °C}$. Aucune exigence supplémentaire ne s'applique à la pression atmosphérique ni à l'humidité.

Lorsque du silicium ouvert est utilisé, l'éclairage doit être atténué ou éteint afin de ne pas générer d'effet photonique.

NOTE Certains matériaux de substrat sont hygroscopiques, ce qui peut avoir une influence sur la permittivité du matériau et sur la tangente de son angle de perte.

8.3 Extraction des paramètres d'IA

L'IA peut être déterminée par des données de conception ou par des calculs à l'aide de mesures dans les conditions suivantes:

- Des tensions d'alimentation nominales et des charges typiques doivent être appliquées au dispositif soumis à l'essai.
- Des signaux d'entrée, tels que le vecteur d'essai spécifique, correspondant au mode de fonctionnement spécifique doivent être appliqués.

8.4 Extraction des paramètres du PDN

Les paramètres du PDN doivent être déterminés en analysant les impédances entre les bornes avec les alimentations nominales.

Les paramètres déterminés ne conviennent et ne sont réutilisables que pour des simulations d'émissions conduites lorsque les conditions de processus, de tension et de température sont les mêmes.

NOTE 1 Le PDN est statique, mais en raison des capacités du puits N et de l'oxyde de grille qui sont impliquées, la tension d'alimentation influe sur celui-ci.

NOTE 2 La plupart des paramètres d'impédance peuvent être déterminés d'après des mesures en utilisant un analyseur d'impédance ou un analyseur de réseau vectoriel.

8.5 Extraction des paramètres d'IBC

L'impédance d'IBC peut être déterminée d'après l'impédance entre Vdd et Vss en soustrayant la partie d'impédance appartenant au PDN. Il convient d'élaborer une méthode détaillée dans le futur.

IECNORM.COM : Click to view the full PDF of IEC 62433-2:2017

Annexe A (normative)

Définitions préliminaires pour la représentation XML

A.1 Notions fondamentales du langage XML

A.1.1 Déclaration XML

Bien que la déclaration XML soit facultative dans un fichier XML, il convient que le fichier de macromodèle ICEM-CE comporte une déclaration XML destinée aux analyseurs syntaxiques XML de base.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

La déclaration XML doit être la première ligne du fichier.

A.1.2 Éléments de base

Toutes les informations sont sauvegardées sous la forme d'éléments XML. Chaque élément commence par une balise de début et se termine par une balise de fin. La balise de début est constituée d'un mot-clé encadré par des crochets triangulaires, <Keyword>. La balise de fin est constituée du même mot-clé préfixé par le caractère "/" et encadré par des crochets, </Keyword>. Le contenu sous forme de texte est encadré par une balise de début et une balise de fin.

Un exemple d'élément est donné ci-dessous:

```
<Keyword>          <!-- start-tag -->
    text            <!-- content -->
</Keyword>         <!-- end-tag -->
```

Il est également autorisé d'écrire un élément sur la même ligne, par exemple pour inclure un contenu court.

```
<Keyword>text</Keyword>
```

Le contenu d'un élément peut être constitué d'un ou plusieurs autres éléments ou d'une valeur (numérique ou alphanumérique). Pour plus de clarté, une indentation avec des caractères de tabulation peut être utilisée. Aucun crochet triangulaire "<" et ">" ne doit faire partie du contenu sauf lorsqu'ils sont utilisés pour encadrer des mots-clés.

Le texte doit utiliser des caractères du jeu de caractères UTF-8 (espace, " ", "<", ">", "&" exclus).

Un élément vide peut être inclus pour indiquer qu'un mot-clé particulier existe, mais il n'a aucun contenu:

```
<empty_element/>
```

A.1.3 Élément racine

Le fichier XML doit contenir un et un seul élément racine. Il encadre tous les autres éléments et en conséquence, il constitue le seul élément parent de tous les autres éléments. La balise de début de l'élément racine est placée au début du fichier ou après la déclaration XML lorsqu'elle existe. La balise de fin de l'élément racine se trouve à la dernière entrée du fichier.

Le mot-clé de l'élément racine ne doit pas être utilisé à d'autres fins dans le fichier XML.

A.1.4 Commentaires

Des commentaires peuvent être insérés dans le fichier, entre "<!--" et "-->". Un exemple est donné ci-dessous:

```
<!-- this line is a comment -->
```

Des commentaires peuvent être insérés n'importe où dans le fichier, sauf à l'intérieur des balises de début et de fin et ils peuvent être écrits sur une seule ligne ou sur plusieurs lignes. L'ensemble du texte encadré par des crochets de commentaire est considéré comme un commentaire et il peut être ignoré.

A.1.5 Terminaisons de ligne

Pour faciliter la lisibilité, le fichier est habituellement organisé en lignes. La séquence de terminaison de ligne doit être, soit un caractère de saut de ligne, soit un caractère de retour chariot suivi par un caractère de saut de ligne.

A.1.6 Hiérarchie des éléments

L'ordre des éléments n'a pas d'importance, mais leur hiérarchie doit être respectée. Un exemple de disposition est donné ci-dessous:

```
<Keyword1> ... </Keyword1>
<Keyword2>
  <Keyword21> ... </Keyword21>
  <Keyword22> ... </Keyword22>
</Keyword2>
<Keyword3> ... </Keyword3>
```

A.1.7 Attributs d'éléments

Les éléments de mots-clés XML peuvent avoir des attributs. Les attributs donnent des informations supplémentaires sur les éléments qui ne font pas partie des données. Les valeurs des attributs doivent être entre guillemets. Des guillemets simples ou doubles peuvent être utilisés. Plusieurs attributs sont séparés par un espace ou par un point-virgule (;). Contrairement aux éléments, les attributs ne contiennent généralement pas plusieurs valeurs ni de structure arborescente, sauf pour certaines exceptions du langage XML. Un exemple de définition d'attribut est donné ci-dessous:

```
<Parent Type="example"; Format="string"/>
```

Dans la syntaxe ci-dessus, *Type* et *Format* sont des attributs du mot-clé *Parent*. Les mêmes termes peuvent être définis comme des éléments. Voir l'exemple suivant:

```
<Parent>
  <Type>example</Type>
  <Format>string</Format>
</Parent>
```

A.2 Exigences relatives aux mots-clés

A.2.1 Généralités

Des mots-clés, placés dans les balises de début et de fin sont utilisés pour introduire des descriptions, des valeurs et des sections qui sont spécifiques au fichier. Certains mots-clés,

tels que *Unit*, *List*, etc., peuvent être présents dans plusieurs sections. Un mot-clé parent est exigé lorsqu'un mot-clé enfant est présent. Les règles ci-dessous garantissent que le fichier peut être correctement analysé par un analyseur syntaxique XML.

A.2.2 Caractères des mots-clés

Seuls des caractères ASCII, tels que définis dans la norme ANSI X4-1986, peuvent être utilisés dans les fichiers [4]. L'utilisation de caractères avec des codes supérieurs à 07E en hexadécimal n'est pas autorisée. D'autre part, les caractères de contrôle ASCII (caractères numériquement inférieurs à 20 en hexadécimal) ne sont pas autorisés, sauf les tabulations ou dans une séquence de terminaison de ligne.

Seuls des caractères alphabétiques ou numériques peuvent être utilisés pour écrire des mots-clés. Les espaces ne sont pas admis. Si nécessaire, le caractère souligné "_" peut séparer les parties d'un mot-clé constitué de plusieurs mots.

A.2.3 Syntaxe des mots-clés

Le contenu du fichier est sensible à la casse. Tous les mots-clés doivent être écrits en minuscules en commençant par une lettre majuscule.

A.2.4 Structure d'un fichier

A.2.4.1 Généralités

Les informations à échanger peuvent être enregistrées dans un fichier XML unique ou dans plusieurs fichiers XML et fichiers de données. Les règles et les lignes directrices suivantes garantissent que les fichiers peuvent être correctement localisés par un analyseur syntaxique XML.

A.2.4.2 Noms de fichier

Pour faciliter la portabilité entre systèmes d'exploitation, il convient que les noms de fichiers soient constitués d'un nom de base ne dépassant pas quarante caractères, suivi d'un point ".", suivi d'une extension de nom de fichier ne dépassant pas quatre caractères. Le nom de fichier et l'extension doivent utiliser les caractères du jeu de caractères présenté en A.1.2 (espace, " ", 0x20 exclus). Le nom de fichier doit être défini dans le mot-clé *Filename* de la manière suivante:

```
<Filename>ExampleXML_file.xml</Filename>
```

A.2.4.3 Chemins de fichier

Pour assurer la portabilité et la compressibilité, seuls des chemins relatifs peuvent être utilisés pour définir un nom de chemin. Un chemin absolu n'est pas exportable et n'est pas admis. Le chemin relatif doit commencer par "./" pour indiquer que le nom de chemin du fichier d'image sera ajouté au chemin du fichier XML courant. Il n'est pas admis de remonter à un plus haut niveau que le chemin XML courant (en utilisant par exemple "../"). Par hypothèse, un nom de fichier sans "./" est situé dans le même répertoire que le fichier XML courant.

A.2.4.4 Fichier XML unique

Lorsque les informations sont contenues dans un fichier XML unique, elles doivent être conformes aux règles et aux lignes directrices applicables aux fichiers XML, comme décrit précédemment.

Les données sont incluses dans la section *Pdn* ou *la* du fichier dans l'élément XML en utilisant le mot-clé *List* chaque fois que cela est nécessaire.

A.2.4.5 Fichiers XML multiples

Le document XML est divisé en plusieurs sections ayant l'élément racine comme parent. Les informations sur le modèle sont alors définies dans la section racine par des mots-clés tels que *Pdn*, *la*, etc. Chaque fichier XML peut contenir une ou plusieurs sections et il doit être conforme aux règles et aux lignes directrices applicables aux fichiers XML, comme décrit précédemment. Une section doit être présente dans un seul des fichiers XML.

Afin de garantir la portabilité et la compressibilité, tous les fichiers XML doivent être placés dans le même répertoire, comme le montre la Figure A.1. L'analyseur syntaxique XML doit analyser tous les fichiers qui se trouvent dans le répertoire principal.

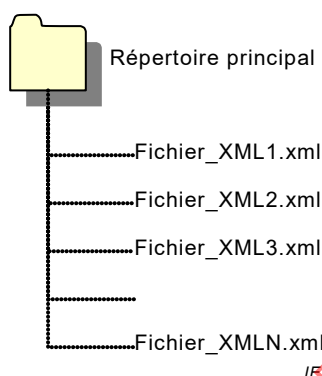


Figure A.1 – Fichiers XML (CEML) multiples

A.2.4.6 Fichiers de données séparés

Les informations peuvent être contenues dans un fichier XML unique ou dans plusieurs fichiers XML et les données peuvent être contenues dans un ou plusieurs fichiers de données supplémentaires. Les fichiers XML doivent être conformes aux règles et aux lignes directrices applicables aux fichiers XML, comme décrit ci-dessus. Les fichiers de données peuvent contenir des lignes de données et les informations d'en-tête doivent être spécifiées avec un point d'exclamation (!) comme premier caractère de chaque ligne d'en-tête (ligne ne contenant pas de données). Il existe toutefois l'exception suivante: pour les fichiers de données avec une extension Touchstone (*.snp, n = 1,2,...) utilisés pour la définition d'un PDN, les lignes commençant par le caractère dièse("#") sont traitées comme une "ligne d'option" qui spécifie le format de données et les unités relatives aux données Touchstone. Une seule "ligne d'option" est autorisée par fichier de données Touchstone. Les noms et les chemins des fichiers de données sont définis par le mot-clé *Data_files* et doivent être conformes à A.2.4.2 et A.2.4.3.

Pour garantir la portabilité et la compressibilité, les fichiers de données doivent être placés, soit dans le même répertoire que les fichiers XML, soit dans un sous-répertoire situé au même niveau ou à un niveau inférieur à celui des fichiers XML, comme indiqué à la Figure A.2. Il n'est pas admis de placer les fichiers supplémentaires à un niveau supérieur à celui des fichiers XML.

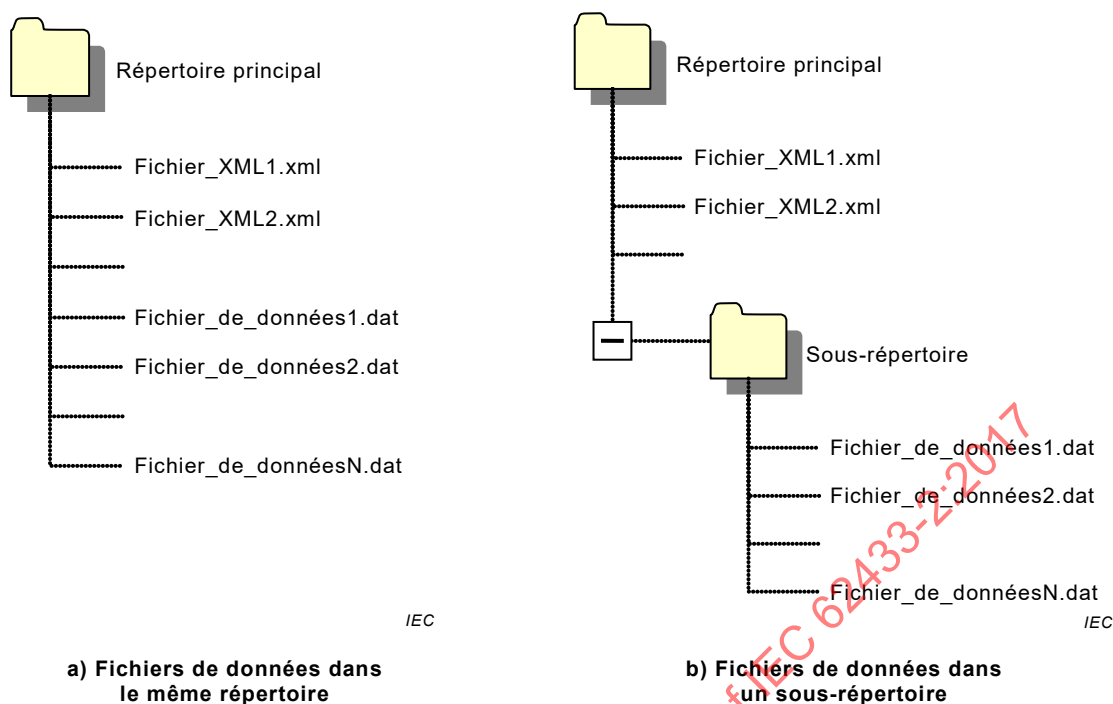


Figure A.2 – Fichiers XML avec des fichiers de données (*.dat)

A.2.4.7 Fichiers supplémentaires

Un fichier XML peut contenir des références à d'autres fichiers, tels que des fichiers de documentation (mot-clé *Documentation*). Pour garantir la portabilité et la compressibilité, ces fichiers supplémentaires doivent être placés soit dans le même répertoire que le fichier XML unique, soit dans un sous-répertoire situé au même niveau ou à un niveau inférieur à celui des fichiers XML, comme indiqué à la Figure A.3. Il n'est pas admis de placer les fichiers supplémentaires à un niveau supérieur à celui des fichiers XML.

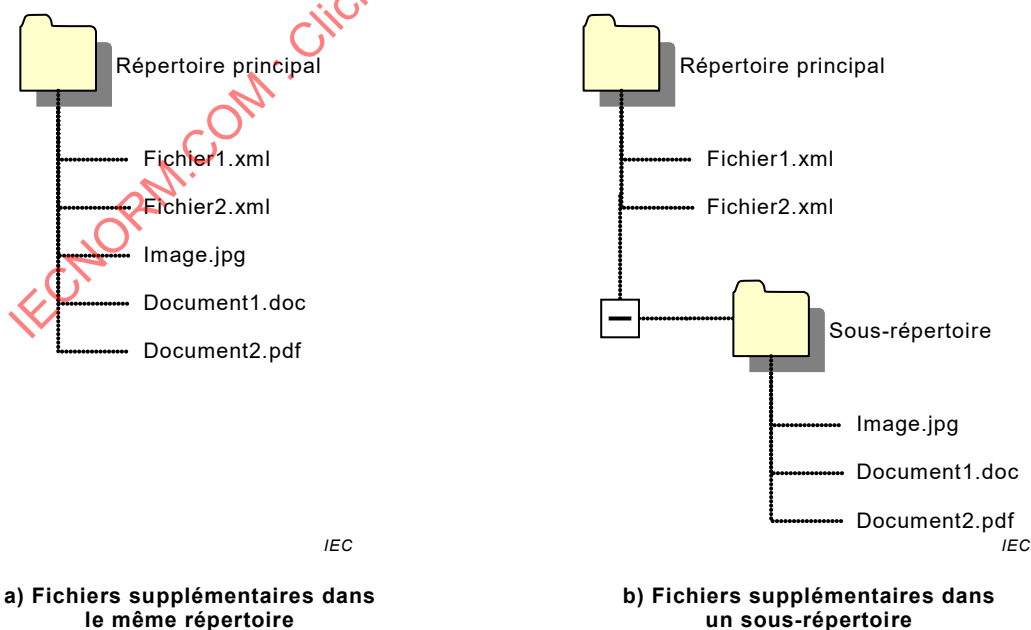


Figure A.3 – Fichiers XML avec des fichiers supplémentaires

A.2.4.8 Compression de fichier

En cas de compression du système de fichiers, des précautions doivent être prises pour inclure dans le fichier compressé les chemins des différents fichiers XML et de données. Ceci

garantit la conservation de la structure de fichiers après décompression. Les chemins ne sont pas nécessaires lorsque tous les fichiers sont enregistrés dans le même répertoire.

A.2.5 Valeurs

A.2.5.1 Généralités

Lorsqu'un élément contient une valeur, celle-ci peut être une valeur numérique (par exemple, 123.45), une valeur numérique avec unités (par exemple, 123.45MHz).

A.2.5.2 Syntaxe des valeurs numériques

Les valeurs numériques peuvent être exprimées sous forme décimale, le point (".") constituant le séparateur décimal (par exemple, 123.45) ou en notation scientifique (par exemple, 1.2345e2). Les espaces " " et les virgules "," qui sont souvent utilisés comme séparateurs des milliers, et les autres caractères, ne sont pas autorisés.

Si plusieurs valeurs numériques sont exigées, elles doivent être séparées par des espaces " " ou des caractères de tabulation.

A.2.5.3 Syntaxe des valeurs numériques avec unités

La valeur numérique (voir A.2.5.2) est suivie d'unités valides, comme décrit cela est en A.2.5.5 (par exemple, 123. 45 MHz). Les espaces ne sont pas autorisés entre la valeur numérique et les unités.

A.2.5.4 Chaîne de caractères

Une chaîne de caractères peut représenter un mot reconnu par l'analyseur syntaxique XML ou peut être un nom de fichier, une description, etc. Une chaîne de caractères peut contenir l'un quelconque des caractères alphanumériques indiqués en A.1.2.

A.2.5.5 Unités valides

Les unités doivent être exprimées sous la forme d'unités SI ou d'unités SI dérivées. Les unités valides sont:

V pour Volt	ohm pour Ohm	S pour Siemens =1/Ohm
A pour Ampère	Hz pour Hertz	m pour mètre
W pour Watt	s pour seconde	H pour Henry
F pour Farads	ohm.m pour résistivité	Celsius pour degrés Celsius
K pour Kelvin	1 pour les grandeurs sans dimensions	

Les unités sont sensibles à la casse.

Les angles sont exprimés en degrés. Le symbole "°" n'est pas exigé.

Le pourcentage est exprimé par "%".

Les facteurs d'échelle valides sont:

T = téra: 1e12	c = centi: 1e-2	p = pico: 1e-12
G = giga: 1e9	m = milli: 1e-3	f = femto: 1e-15
M = méga: 1e6	u = micro: 1e-6	a = atto: 1e-18
k = kilo: 1e3	n = nano: 1e-9	

Lorsqu'aucun facteur d'échelle n'est spécifié, les unités sont par hypothèse les unités de base. Celles-ci sont les volts, ampères, watts, ohms, siemens, hertz, mètres, secondes, Henry, Farads, ohm.m, Celsius et Kelvin. Les abréviations des unités (par exemple, pV, nA, ms, MHz) doivent être utilisées à l'exception de l'Ohm qui doit être écrit en entier. Le Tableau A.1 donne une liste d'unités logarithmiques valides.

Tableau A.1 – Unités logarithmiques valides

Utilisation	Symbole	Unité	Référence
Rapport	dB	Décibel	1
Puissance	dBW	Décibel Watt	1W
Puissance	dBmW	Décibel milli-Watt	1mW
Puissance	dBuW	Décibel micro-Watt	1μW
Tension	dBV	Décibel Volt	1V
Tension	dBmV	Décibel milli-Volt	1mV
Tension	dBuV	Décibel micro-Volt	1μV
Ampère	dBA	Décibel Ampère	1A
Ampère	dBmA	Décibel milli-Ampère	1mA
Ampère	dBuA	Décibel micro-Ampère	1μA
Impédance	dBohm	Décibel Ohm	1ohm
Admittance	dBS	Décibel Siemens	1S

Les unités peuvent être définies dans n'importe quelle section et avec le mot-clé *Unit*. Une description typique d'unité est donnée ci-dessous:

```
<Unit>
    kHz
</Unit>
```

Annexe B (normative)

Mots-clés CEML valides et usage

B.1 Mots-clés de l'élément racine (Root)

Les mots-clés suivants sont placés au début du fichier après la définition XML.

Mot-clé	Parent	Description	Présence	Exemple
Header	Elément Root	Spécifie une section d'en-tête	Exigée	<Header> </Header>
Lead_definitions	Elément Root	Spécifie une section définitions de broche. Les informations de cette section sont utilisées dans toutes les autres sections	Exigée	<Lead_definitions> </Lead_definitions >
Validity	Elément Root	Spécifie une section de validité	Exigée	<Validity> ... </Validity>
Macromodels	Elément Root	Spécifie une section macromodèles SPICE	Facultative	<Macromodels> </Macromodels>
Pdn	Elément Root	Spécifie les données du PDN	Exigée	<Pdn> </Pdn>
Ia	Elément Root	Spécifie les données de l'IA	Exigée	<Ia> </Ia>
Ibc	Elément Root	Spécifie les données de l'IBC	Facultative	<Ibc> </Ibc>

B.2 Mots-clés d'en-tête (Header) de fichier

Les mots-clés suivants sont placés au début du fichier après la balise de début de l'élément racine.

Mot-clé	Parent	Description	Présence	Exemple
Cem_ver	Elément Header	Spécifie la version du format de fichier (1.0) Il convient qu'elle suive l'élément d'en-tête. Informe les analyseurs syntaxiques de la version du format d'échange XML utilisée et leur permet de savoir à quels mots-clés s'attendre.	Exigée	<Cem_ver>1.0</Cem_ver>
Filename	Elément Header	Spécifie le nom de fichier. Suit normalement le mot-clé <i>Cem_ver</i> .	Exigée	<Filename> My_file.xml </Filename>
File_ver	Elément Header	Suit le niveau de révision d'un fichier .xml particulier. Le niveau de révision est fixé à la discrétion de l'émetteur du fichier.	Exigée	<File_ver>1.0</File_ver>
Date	Elément Header	La valeur peut contenir des blancs, être d'un format quelconque, mais il convient qu'elle soit limitée à 20 caractères au maximum. Il convient que le mois soit épilé pour plus de clarté. L'analyseur syntaxique considère cette information comme une chaîne de données et ne l'interprète pas.	Facultative	<Date> January 1, 2013 </Date>
Author	Elément Header	La valeur peut contenir des blancs et être d'un format quelconque. L'analyseur syntaxique considère cette information comme une chaîne de données et ne l'interprète pas.	Facultative	<Author> Name 1 Name2 </Author>
Dut	Elément Header	La valeur peut contenir des blancs et être d'un format quelconque. L'analyseur syntaxique considère cette information comme une chaîne de données et ne l'interprète pas.	Facultative	<Dut> 32-bit microcontroller </Dut>
Meas_method	Elément Header	La valeur peut contenir des blancs et être d'un format quelconque. L'analyseur syntaxique considère cette information comme une chaîne de données et ne l'interprète pas.	Facultative	<Meas_method> VNA, S param </Meas_method>
Disclaimer	Elément Header	La valeur peut contenir des blancs et être d'un format quelconque. L'analyseur syntaxique considère cette information comme une chaîne de données et ne l'interprète pas.	Facultative	<Disclaimer> This file contains results of model extraction. Other use is not guaranteed </Disclaimer>
Copyright	Elément Header	La valeur peut contenir des blancs et être d'un format quelconque. L'analyseur syntaxique considère cette information comme une chaîne de données et ne l'interprète pas.	Facultative	<Copyright> Copyright 2013, XYZ Corp., All Rights Reserved </Copyright>

B.3 Mots-clés de la section validité (Validity)

Les mots-clés suivants sont placés dans la balise de début de l'élément racine.

Mot-clé	Parent	Description	Présence	Exemple
Power_supply	Elément Validity	Spécifie les conditions d'alimentation dans lesquelles le macromodèle ICEM-CE est extrait et défini. Il convient qu'elle suive l'élément de validité. Informe l'utilisateur sur le critère de validité du macromodèle ICEM-CE. Une valeur numérique valide avec ou sans unité est autorisée. Voir A.2.5.3 pour plus d'informations.	Exigée	<Power_supply> 9V </Power_supply>
Frequency_range	Elément Validity	Spécifie la plage de fréquences valide dans laquelle le macromodèle ICEM-CE dans le domaine fréquentiel est extrait et défini. Il convient qu'elle suive l'élément de validité. Informe l'utilisateur sur le critère de validité du macromodèle ICEM-CE. Une valeur numérique valide avec ou sans unité est autorisée. Voir A.2.5.3 pour plus d'informations.	Exigée	<Frequency_range> [10kHz-100MHz] </Frequency_range>
Time_range	Elément Validity	Spécifie la plage de temps valide dans laquelle le macromodèle ICEM-CE dans le domaine temporel est extrait et défini. Il convient qu'elle suive l'élément de validité. Informe l'utilisateur sur le critère de validité du macromodèle ICEM-CE. Une valeur numérique valide avec ou sans unité est autorisée. Voir A.2.5.3 pour plus d'informations.	Exigée	<Time_range> [0ms-1ms] </Time_range>
Temperature_range	Elément Validity	Spécifie la plage de températures valide dans laquelle le macromodèle ICEM-CE est extrait et défini. Il convient qu'elle suive l'élément de validité. Informe l'utilisateur sur le critère de validité du macromodèle ICEM-CE. Une valeur numérique valide avec unité est autorisée. Voir A.2.5.3 pour plus d'informations.	Facultative	<Temperature_range> [25Celsius-55Celsius] </Temperature_range>
Phase_origin	Elément Validity	Spécifie l'origine des phases correspondant à la définition de l'IA du macromodèle ICEM-CE. Il convient qu'elle suive l'élément de validité. Informe l'utilisateur sur le critère de validité du macromodèle ICEM-CE. Une valeur numérique valide avec unité est autorisée. Voir A.2.5.3 pour plus d'informations.	Exigée	<Phase_origin> 0degrees </Phase_origin>
Time_origin	Elément Validity	Spécifie l'origine des temps correspondant à la définition de l'IA du macromodèle ICEM-CE. Il convient qu'elle suive l'élément de validité. Informe l'utilisateur sur le critère de validité du macromodèle ICEM-CE. Une valeur numérique valide avec unité est autorisée. Voir A.2.5.3 pour plus d'informations.	Exigée	<Time_origin> 100ps </Time_origin>

B.4 Mots-clés globaux

Les mots-clés suivants peuvent être placés n'importe où dans le fichier, sauf dans un élément XML contenant une valeur.

Mot-clé	Parent	Description	Présence	Exemple
Notes	Tout élément sauf ceux qui contiennent une valeur.	Ajoute facultativement des informations relatives au paramétrage, aux données, etc. La valeur peut contenir des blancs et être d'un format quelconque. Une section de notes peut être insérée n'importe où dans le fichier et le nombre de sections de notes dans le fichier n'est pas limité. L'analyseur syntaxique considère cette information comme une chaîne de données et ne l'interprète pas.	Facultative	<pre><Notes> Use this section for any special notes </Notes></pre>
Documentation	Tout élément sauf ceux qui contiennent une valeur.	Ajoute facultativement les chemins aux fichiers contenant la documentation du projet.	Facultative	<pre><Documentation> Project doc.pdf Model_descr.doc </Documentation></pre>

B.5 Mots-clés de broches (Lead)

Le mot-clé suivant peut être placé dans les sections *Lead_definitions*, *Pdn*, *Ibc* et *Ia* comme élément parent.

Mot-clé	Parent	Description	Présence	Exemple
Lead	Tout élément: <i>Lead_definitions</i> , <i>Pdn</i> , <i>Ia</i> et <i>Ibc</i>	Spécifie l'élément <i>Lead</i> . Les attributs acceptés peuvent varier d'une section à une autre.	Exigée	<pre><Lead> ... </Lead></pre>

B.6 Attributs de la section définition de broche (Lead_definitions)

Le mot-clé *Lead* dans la section *Lead_definitions* comporte les attributs valides suivants.

Mot-clé	Parent	Description	Présence	Exemple
Id	Elément Lead	Spécifie l'identité ou le numéro de l'élément <i>Lead</i> . Une seule valeur est autorisée.	Exigée	<Lead Id="1"> ... </Lead>
Name	Elément Lead	Spécifie le nom l'élément <i>Lead</i> . Il peut s'agir de n'importe quelle chaîne valide.	Facultative	<Lead Id="1" Name="ET1"> ... </Lead>
Mode	Elément Lead	Spécifie le mode dans lequel une broche particulière est utilisée GND, None	Facultative	<Lead Id="1" Name="ET1" Mode="GND"> ... </Lead>
Type	Elément Lead	Spécifie le type dans lequel une broche particulière est utilisée: internal, external	Facultative	<Lead Id="1" Name="ET1" Mode="GND" Type="external"> ... </Lead>

B.7 Attribut de la section macromodèles (Macromodels)

Le mot-clé *Subckt* dans la section *Macromodels* comporte les attributs valides suivants.

Mot-clé	Parent	Description	Présence	Exemple
Name	Elément Subckt	Spécifie le nom du macromodèle.	Exigée	<Subckt Name="MD1" ...> ... </Subckt>
Nodes	Elément Subckt	Spécifie le nœud externe du macromodèle par lequel le circuit secondaire se connecte au circuit principal.	Exigée	<Subckt Name="MD1" Nodes="N1,N2,N3"> ... </Subckt>
Kind	Elément Subckt	Spécifie le genre de liste d'interconnexions utilisée pour définir le circuit secondaire. Voir 7.6 pour plus d'informations.	Facultative	< Subckt Name="MD1" Nodes="N1,N2,N3" Kind="SPICE3"> ... </Subckt>
Data_files	Elément Subckt	Spécifie les noms de chemins des fichiers contenant une liste d'interconnexions de circuit secondaire. Les noms de chemin sont séparés par un espace (" ") ou par une terminaison de ligne. Les noms de fichiers et les chemins doivent être conformes à A.2.4.2 et A.2.4.3. Un seul mot-clé <i>Data_files</i> ou <i>List</i> doit être inclus dans la section <i>Subckt</i> .	Facultative	< Subckt Kind="SPICE3"> ... <Data_files> PDN_Pin1_macro.lib </Data_files> ... </Subckt>

B.8 Mots-clés de la section réseau de distribution passif (Pdn)

B.8.1 Mots-clés de l'élément broche (Lead)

Les données du PDN de chaque broche soumise aux essais défini dans la balise *Lead*. Le mot-clé *Lead* dans la section *Pdn* comporte les attributs valides suivants.

Mot-clé	Parent	Description	Présence	Exemple
Id	Elément Lead	Spécifie l'identité ou le numéro de l'élément <i>Lead</i> . Un seul ou plusieurs identifiants sont autorisés pour différencier les broches à une seule extrémité, différentielles ou à plusieurs extrémités. Pour les broches différentielles ou à plusieurs extrémités, les identifiants doivent être séparés par une virgule ","	Exigée	<Lead Id="1"...> ... </Lead> <Lead Id="1,2,3"...> ... </Lead>
Ground_id	Elément Lead	Spécifie l'identité ou le numéro de l'élément <i>Lead</i> utilisé comme broche de référence. Un seul identifiant est autorisé.	Exigée pour Type="S", "Z" ou "Y"	<Lead Id="1" Ground_id="10"...> ... </Lead>
Blockname	Elément Lead	Spécifie le nom du sous-bloc lorsqu'un modèle de PDN basé sur des blocs est utilisé.	Facultative	<Lead Id="1" Ground_id="10" Blockname="Block1"> ... </Lead>
Type	Elément Lead	Spécifie le type de données du PDN: "S" pour les paramètres S, "Z" pour les paramètres Z, "Y" pour les paramètres Y, "Ckt" pour la définition de circuit. Voir 7.8.2.4	Facultative	<Lead Id="1" Type="Z"> ... </Lead>
Param_order	Elément Lead	Spécifie l'ordre dans lequel les paramètres du PDN sont définis. Des informations détaillées sont données en 7.8.2.5. Non défini, si Data_files au format Touchstone et Netlist sont utilisés	Facultative	<Lead Id="1" Type="Z" Param_order="Freq,Z 11"> ... </Lead>
Format	Elément Lead	Spécifie le format de données du PDN. Voir 7.8.2.6. Non défini, si Data_files au format Touchstone et Netlist sont utilisés	Facultative	<Lead Id="1" Type="Z" Param_order="Freq,Z 11" Format="MA"> ... </Lead>
Meas_type	Elément Lead	Spécifie la méthode mise en œuvre pour réaliser les mesures du PDN. Voir 7.8.2.7. Non défini si Netlist est utilisé	Facultative	<Lead Id="1" Type="S" Param_order="Freq,S 21" Format="MA" Meas_type="1"> ... </Lead>
Reference_impedance	Elément Lead	Spécifie l'impédance de référence utilisée pour les mesures du PDN. Non défini si Netlist est utilisé	Facultative	<Lead Id="1" Type="S" Param_order="Freq,S 11" Format="MA" Reference_impedance="50ohm"> ... </Lead>

Mot-clé	Parent	Description	Présence	Exemple
Use	Elément Lead	Spécifie un des paramètres définis dans Param_order, sauf la fréquence qui doit être utilisée spécifiquement. Voir 7.8.2.9. Non défini si Netlist est utilisé	Facultative	<pre><Lead Id="1" Type="S" Param_order="Freq,S 21" Format="MA" Meas_type="1" Use="S21"> ... </Lead></pre>
Netlist	Elément Lead	Spécifie la connectivité électrique des éléments du PDN sous forme de circuit en utilisant des déclarations de données de type SPICE.	Exigée si Type="Ckt"	<pre><Lead Id="1" Type="Ckt"> ... <Netlist> ... </Netlist> ... </Lead></pre>
Unit_freq	Elément Lead	Spécifie les unités des fréquences utilisées pour spécifier le PDN. La valeur doit être conforme à A.2.5.5. Si ce mot-clé est omis, les unités sont par hypothèse des unités SI. Non défini, si Netlist et Data_files au format Touchstone sont utilisés.	Facultative	<pre><Lead Id="1" Type="S"> <Unit_freq> MHz </Unit_freq> ... </Lead></pre>
Unit_param	Elément Lead	Spécifie les unités des paramètres utilisés pour spécifier le PDN. La valeur doit être conforme à A.2.5.5. Si ce mot-clé est omis, les unités sont par hypothèse des unités SI. Non défini, si Netlist et Data_files au format Touchstone sont utilisés.	Facultative	<pre><Lead Id="1" Type="S"> <Unit_param> dB </Unit_param> ... </Lead></pre>
Power_level	Elément Lead	Spécifie le niveau de puissance utilisé pour obtenir les données du PDN. La valeur doit être conforme à A.2.5.5 et il convient que la valeur et l'unité soient indiquées. En l'absence d'unité, les unités SI sont prises par hypothèse. Non défini si Netlist et Data-files sont utilisés.	Facultative	<pre><Lead Id="1" Type="S"> <Power_level> -10dBm </Power_level> ... </Lead></pre>
Data_files	Elément Lead	Spécifie les noms de chemins des fichiers contenant une liste de données du PDN correspondant à la section <i>Pdn</i> . Les noms de chemin sont séparés par un espace (" ") ou par une terminaison de ligne. Les noms de fichiers et les chemins doivent être conformes à A.2.4.2 et A.2.4.3. Un seul mot-clé <i>Data_files</i> ou <i>List</i> doit être inclus dans la section <i>Pdn</i> .	Exigée si différent de <i>List</i>	<pre><Lead Id="1" Type="S"> ... <Data_files> Pdn_pin1_S11.txt </Data_files> ... </Lead></pre>
List	Elément Lead	Spécifie une liste de données du PDN dans le modèle.	Exigée si différent de <i>Data_files</i>	<pre><Lead Id="1" Type="S"> ... <List> ... </List> ... </Lead></pre>

B.8.2 Mots-clés de la section liste d'interconnexions (Netlist)

Les mots-clés suivants peuvent être placés dans la section *Netlist* sous la balise *Pdn* comme élément parent.

Mot-clé	Parent	Description	Présence	Exemple
Kind	Elément Netlist	Spécifie le genre de liste d'interconnexions utilisée pour définir le PDN. Voir 7.8.3.4 pour plus d'informations.	Facultative	<pre><Netlist Kind="SPICE3"> R1 1 10 1e3 ... </Netlist></pre>
Data_files	Elément Netlist	Spécifie les noms de chemins des fichiers contenant une liste d'interconnexions. Les noms de chemin sont séparés par un espace (" ") ou par une terminaison de ligne. Les noms de fichiers et les chemins doivent être conformes à A.2.4.2 et A.2.4.3. Un seul mot-clé <i>Data_files</i> ou <i>List</i> doit être inclus dans la section <i>Netlist</i> .	Facultative	<pre>< Netlist Kind="SPICE3"> ... <Data_files> PDN_Pin1.net </Data_files> ... </Netlist></pre>

B.9 Mots-clés de la section couplage entre blocs (Ibc)

B.9.1 Mots-clés de l'élément broche (Lead)

Les données de l'IBC de chaque broche soumise aux essais défini dans la balise *Lead*, semblable au PDN. Le mot-clé *Lead* dans la section *Ibc* comporte les attributs valides suivants.

Mot-clé	Parent	Description	Présence	Exemple
Id	Elément Lead	Spécifie l'identité ou le numéro de l'élément <i>Lead</i> . Un seul ou plusieurs identifiants sont autorisés pour différencier les broches à une seule extrémité, différentielles ou à plusieurs extrémités. Pour les broches différentielles ou à plusieurs extrémités, les identifiants doivent être séparés par une virgule ",". Voir 7.8.2.1.	Exigée	<pre><Lead Id="1,2"...> ... </Lead> <Lead Id="1,2,3"...> ... </Lead></pre>
Blockname	Elément Lead	Spécifie le nom du bloc IBC. Voir 7.8.2.3.	Facultative	<pre><Blockname> Block1_IBC </Blockname></pre>
Type	Elément Lead	Spécifie le type de données de l'IBC: "S" pour les paramètres <i>S</i> , "Z" pour les paramètres <i>Z</i> , "Y" pour les paramètres <i>Y</i> , "Ckt" pour la définition de circuit. Voir 7.8.2.4.	Facultative	<pre><Lead Id="1,2" Type="Z"> ... </Lead></pre>
Param_order	Elément Lead	Spécifie l'ordre dans lequel les paramètres de l'IBC sont définis. Des informations détaillées sont données en 7.8.2.5. Non défini, si <i>Data_files</i> au format Touchstone et <i>Netlist</i> sont utilisés	Facultative	<pre><Lead Id="1,2" Type="Z" Param_order="Freq, Z11"> ... </Lead></pre>

Mot-clé	Parent	Description	Présence	Exemple
Format	Elément Lead	Spécifie le format de données de l'IBC. Voir 7.8.2.6. Non défini, si Data_files au format Touchstone et Netlist sont utilisés	Facultative	<pre><Lead Id="1,2" Type="Z" Param_order="Freq, Z11" Format="MA"> ... </Lead></pre>
Meas_type	Elément Lead	Spécifie la méthode mise en œuvre pour réaliser les mesures de l'IBC. Voir 7.8.2.7. Non défini si Netlist est utilisé	Facultative	<pre><Lead Id="1,2" Type="S" Param_order="Freq, S21" Format="MA" Meas_type="1"> ... </Lead></pre>
Reference_impedance	Elément Lead	Spécifie l'impédance de référence utilisée pour les mesures de l'IBC. Non défini si Netlist est utilisé Voir 7.8.2.8.	Facultative	<pre><Lead Id="1,2" Type="S" Param_order="Freq, S11" Format="MA" Reference_impedance="50ohm"> ... </Lead></pre>
Use	Elément Lead	Spécifie un des paramètres définis dans Param_order, sauf la fréquence qui doit être utilisée spécifiquement. Voir 7.8.2.9. Non défini si Netlist est utilisé	Facultative	<pre><Lead Id="1,2" Type="S" Param_order="Freq, S21" Format="MA" Meas_type="1" Use="S21"> ... </Lead></pre>
Netlist	Elément Lead	Spécifie la connectivité électrique des éléments de l'IBC sous forme de circuit en utilisant des déclarations de données de type SPICE. Voir 7.8.2.10.	Exigée si Type="Ckt"	<pre><Lead Id="1,2" Type="Ckt"> ... <Netlist> </Netlist> ... </Lead></pre>
Unit_freq	Elément Lead	Spécifie les unités des fréquences utilisées pour spécifier l'IBC. La valeur doit être conforme à A.2.5.5 Si ce mot-clé est omis, les unités sont par hypothèse des unités SI. Non défini, si Netlist et Data_files au format Touchstone sont utilisés.	Facultative	<pre><Lead Id="1,2" Type="S"> <Unit_freq> MHz </Unit_freq> ... </Lead></pre>
Unit_param	Elément Lead	Spécifie les unités des paramètres utilisés pour spécifier l'IBC. La valeur doit être conforme à A.2.5.5 Si ce mot-clé est omis, les unités sont par hypothèse des unités SI. Non défini, si Netlist et Data_files au format Touchstone sont utilisés.	Facultative	<pre><Lead Id="1,2" Type="S"> <Unit_param> dB </Unit_param> ... </Lead></pre>
Power_level	Elément Lead	Spécifie le niveau de puissance utilisé pour obtenir les données de l'IBC. La valeur doit être conforme à A.2.5.5 et il convient que la valeur et l'unité soient indiquées. En l'absence d'unité, les unités SI sont prises par hypothèse. Non défini si Netlist et Data-files sont utilisés.	Facultative	<pre><Lead Id="1,2" Type="S"> <Power_level> -10dBm </Power_level> ... </Lead></pre>

Mot-clé	Parent	Description	Présence	Exemple
Data_files	Elément Lead	Spécifie les noms de chemins des fichiers contenant une liste de données de l'IBC correspondant à la section <i>Ibc</i> . Les noms de chemin sont séparés par un espace (" ") ou par une terminaison de ligne. Les noms de fichiers et les chemins doivent être conformes à A.2.4.2 et A.2.4.3. Un seul mot-clé <i>Data_files</i> ou <i>List</i> doit être inclus dans la section <i>Ibc</i> .	Exigée si différent de <i>List</i>	<pre><Lead Id="1,2" Type="S"> ... <Data_files> Ibc_pin1_S11.txt </Data_files> ... </Lead></pre>
List	Elément Lead	Spécifie une liste de données de l'IBC dans le modèle.	Exigée si différent de <i>Data_files</i>	<pre><Lead Id="1,2" Type="S"> ... <List> ... </List> ... </Lead></pre>

B.9.2 Mots-clés de la section liste d'interconnexions (Netlist)

Les mots-clés suivants peuvent être placés dans la section *Netlist* sous la balise *Ibc* comme élément parent.

Mot-clé	Parent	Description	Présence	Exemple
Kind	Elément Netlist	Spécifie le genre de liste d'interconnexions utilisée pour définir l'IBC. Voir 7.8.3.4 pour plus d'informations.	Facultative	<pre><Netlist Kind="SPICE3"> R1 1 10 1e3 ... </Netlist></pre>
Data_files	Elément Netlist	Spécifie les noms de chemins des fichiers contenant une liste d'interconnexions. Les noms de chemin sont séparés par un espace (" ") ou par une terminaison de ligne. Les noms de fichiers et les chemins doivent être conformes à A.2.4.2 et A.2.4.3. Un seul mot-clé <i>Data_files</i> ou <i>List</i> doit être inclus dans la section <i>Netlist</i> .	Facultative	<pre>< Netlist Kind="SPICE3"> ... <Data_files> IBC_IT1_IT0.net </Data_files> ... </Netlist></pre>

B.10 Mots-clés de la section activité interne (Ia)

B.10.1 Mots-clés de l'élément broche (Lead)

Les données de l'IA de chaque broche soumise aux essais définie dans la balise *Lead*. Le mot-clé *Lead* dans la section *Ia* comporte les attributs valides suivants.

Mot-clé	Parent	Description	Présence	Exemple
Id	Elément Lead	Spécifie l'identité ou le numéro de l'élément <i>Lead</i> . Seuls deux identifiants sont admis par définition de broche puisque la source d'IA est toujours connectée entre deux nœuds. Les identifiants doivent être séparés par une virgule ",".	Exigée	<pre><Lead Id="1,2"...> ... </Lead> <Lead Id="1,2,3"...> ... </Lead></pre>
Type	Elément Lead	Spécifie le type de données de l'IA: "freq" pour les données du domaine fréquentiel et "time" pour les données du domaine temporel.	Exigée	<pre><Lead Id="1,2" Type="freq"> ... </Lead></pre>
Blockname	Elément Lead	Spécifie le nom du sous-bloc lorsqu'un modèle d'IA basé sur des blocs est utilisé.	Facultative	<pre><Lead Id="1,2" Type="freq" > <Blockname> Block1 </Blockname> ... </Lead></pre>
Voltage	Elément Lead	Spécifie les données de l'IA comme une tension. Voir B.10.2	Exigée si différent de <i>Current</i>	<pre><Lead Id="1,2" Type="freq" > <Voltage...> ... </Voltage> ... </Lead></pre>
Current	Elément Lead	Spécifie les données de l'IA comme un courant. Voir B.10.3	Exigée si différent de <i>Voltage</i>	<pre><Lead Id="1,2" Type="freq" > <Current...> ... </Current> ... </Lead></pre>

B.10.2 Mots-clés de la section tension (Voltage)

Les mots-clés suivants peuvent être placés dans la section *Voltage* sous la balise *Lead* comme élément parent (à l'intérieur de la section *la*).