



BSI Standards Publication

Format for LSI-Package-Board Interoperable design

National foreword

This British Standard is the UK implementation of IEC 63055:2023. It supersedes BS IEC 63055:2016, which is withdrawn.

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

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

Contractual and legal considerations

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

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

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

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

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 27840 8

ICS 31.180; 31.200; 35.060

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2023.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

INTERNATIONAL IEEE Std 2401™ STANDARD



Format for LSI-Package-Board Interoperable design

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 31.180, 31.200, 35.060

ISBN 978-2-8322-7515-3

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

Contents

1. Overview	11
1.1 Scope	11
1.2 Purpose	11
1.3 Key characteristics of the LSI-Package-Board Format	11
1.4 Contents of this standard	12
1.5 Word usage	13
2. Normative references	13
3. Definitions, acronyms, and abbreviations	13
3.1 Definitions	13
3.2 Acronyms and abbreviations	16
4. Concept of the LPB Format	18
4.1 Technical background	18
4.2 Conventional design	18
4.3 Common problems at the design site	19
4.4 Concept of LPB interoperable design	19
4.5 Value creation by LPB interoperable design	19
4.6 LPB Format	21
4.7 Summary of LPB Format files	21
5. Language basics	27
5.1 General	27
5.2 Typographic and syntax conventions	27
5.3 Character information	28
5.4 Notation of the floating point number	28
5.5 File naming definitions	28
6. Common elements in M-Format, C-Format, and R-Format	29
6.1 General	29
6.2 The <extensions> element	29
6.3 The <header> element	30
6.4 The <global> element	32
7. M-Format	48
7.1 M-Format file structure	48
7.2 The <include> element	49
7.3 The <current_phase> element	49
7.4 The <class> element	50
8. C-Format	56
8.1 C-Format file structure	56
8.2 The <module> element	56
8.3 The <component> element	161
9. R-Format	165
9.1 R-Format file structure	165
9.2 The <Physicaldesign> element	166
9.3 The <Constrainrule> element	201

10. N-Format	211
10.1 Purpose of the N-Format file	211
10.2 How to identify the power/ground network	211
10.3 Example	212
11. G-Format	212
11.1 Language basics of G-Format	212
11.2 Structure	213
11.3 Header section	214
11.4 Material section	215
11.5 Layer section	215
11.6 Shape section	216
11.7 Board geometry section	220
11.8 Padstack section	221
11.9 Part section	222
11.10 Component section	224
11.11 Net attribute section	225
11.12 Netlist section	225
11.13 Via section	227
11.14 Bondwire section	228
11.15 Route section	229
Annex A (informative) Bibliography	233
Annex B (informative) Examples of utilization	235
B.1 Understanding the function of the LPB Format	235
B.2 Test bench	235
B.3 Design flow example	237
B.4 Growth of the sample files in the LPB Format	269
B.5 Simulations using the sample files in the LPB Format	273
Annex C (informative) XML Encryption	275
Annex D (informative) MD5 checksum	278
Annex E (informative) Chip-Package Interface Protocol	279
E.1 General	279
E.2 Comparison of C-Format with Chip-Package Interface Protocol	279
Annex F (informative) M-Format usage examples	284
F.1 General	284
F.2 Bundling files for a product	284
F.3 Bundling files for a library	285
F.4 Managing design versions using M-Format	287
Annex G (informative) Example of how to express the connector parts as terminals	288
G.1 General	288
Annex H (informative) Participants	292

FORMAT FOR LSI-PACKAGE-BOARD INTEROPERABLE DESIGN –

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC document(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation.

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

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

- 2) The formal decisions of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE SA) Standards Board.
- 3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.
- 8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility.

IEC 63055/IEEE Std 2401 was processed through IEC technical committee 91: Electronics assembly technology, under the IEC/IEEE Dual Logo Agreement. It is an International Standard.

IEEE Std	FDIS	Report on voting
2401 (2019)	91/1869/FDIS	91/1887/RVD

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

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

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

- reconfirmed,
- withdrawn, or
- revised.

IMPORTANT – The "colour inside" logo on the cover page of this document 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.

IEEE Standard Format for LSI-Package-Board Interoperable Design

Developed by the

Design Automation Standards Committee
of the
IEEE Computer Society

Approved 7 November 2019

IEEE SA Standards Board



Documents

Document Sharing Hub

Published by IEC under licence from IEEE. © 2019 IEEE. All rights reserved.

Copyrights and permissions

All figures in this standard are reprinted with permission from JEITA.¹

Abstract: A method is provided for specifying a common interoperable format for electronic systems design. The format provides a common way to specify information/data about the project management, netlists, components, design rules, and geometries used in the large-scale integration-package-board designs. The method provides the ability to make electronic systems a key consideration early in the design process; design tools can use it to seamlessly exchange information/data.

Keywords: common interoperable format, components, design analysis, design rules, geometries, IEEE 2401™, large-scale integration (LSI), netlists, packages for LSI circuits, printed circuit board, project management, Verilog-HDL



Important notices and disclaimers concerning IEEE Standards documents

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

Notice and disclaimer of liability concerning the use of IEEE Standards documents

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

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

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

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

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

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

Translations

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

Official statements

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

Comments on standards

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

Comments on standards should be submitted to the following address:

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

Laws and regulations

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

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

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

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

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

Errata

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

Patents

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

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IEEE Introduction

This introduction is not part of IEEE Std 2401-2019, IEEE Standard Format for LSI-Package-Board Interoperable Design.

To deal with the increasing difficulty of design and the cost competitiveness of the global market, and to shorten the development term, innovative design methodologies should be implemented. It has been difficult to achieve the optimization of an entire set of large-scale integration (LSI), package, and board (LPB) using individual design processes for each LPB part.

One possibility for optimization is to have a certain section design the whole LPB; however, gathering knowledge and integrating the design environment of each LPB part is difficult. Dedicated professional technicians of individual LPB parts, who have the best knowledge and performance of their own part's design tools, intend to create design optimization by having proper interoperable information exchanges among all LPB parties. In order to achieve a design that optimizes the balance between cost and performance, information about and the results of design should be well shared among cooperating LPB design sections.

The standard format of LSI package board interoperable design (hereinafter called *LPB Format*) was developed to make it easier to exchange information among LPB design departments, so that optimal design will be carried out quickly.

The LPB interoperable design process has the following issues:

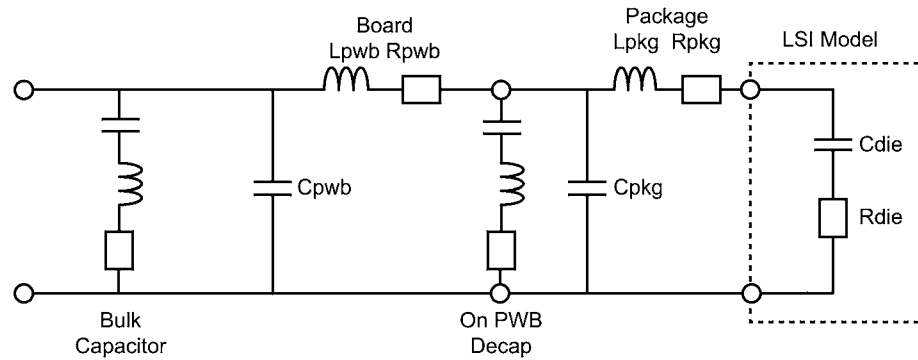
- Netlist not unified on each LPB
- Complexity of the representation of the relationship as a whole arrangement of the LPB
- Differences in how to give the design constraints, lack of design information, and many discrepancies in design rules
- Databases not unified in each LPB, or among different vendors
- No unified terms

Various problems caused by these issues include the following:

- A large effort is required for conversion of formats.
- The occurrence of conversion errors and connection errors is difficult to detect because there is a lack of the information needed to do so.
- It takes a long time to gather information, resulting in a long period of design and analysis.
- It is difficult to make optimal design changes because the entire verification process is difficult.
- Electronic design automation (EDA) tool cost increase because of additional development required to support multiple formats.
- It is time consuming for designers to communicate their intentions in a way that others understand.

Based on this analysis, the interface LPB Format has been developed which can address these issues.

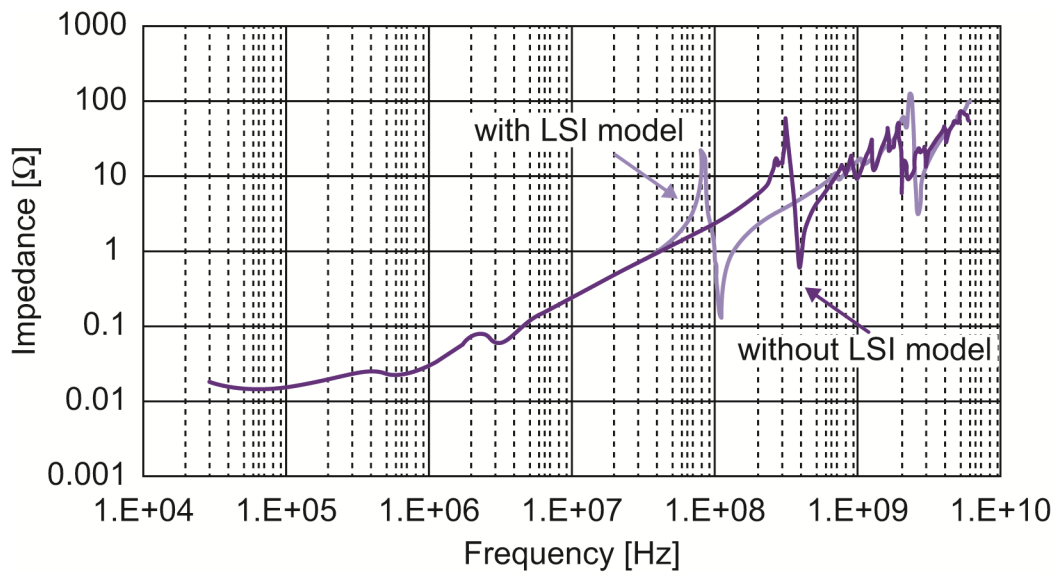
As the one of the case studies of the LPB interoperable design process, the power distribution network (PDN) should be designed with information about the other LPB parts to reduce the noise (see Figure i).



Reprinted with permission from JEITA.

Figure i—Power distribution network

Resonance is caused by a capacitance and inductance present in the various parts in the LPB PDN. Impedance at the resonant frequency will be extremely large. If each part of the overall LPB design is not accurately simulated in the PDN model, the power supply circuit cannot be correctly designed (see Figure ii).



Reprinted with permission from JEITA.

Figure ii—Example of PDN impedance

In order to run properly, this simulation should align a variety of information, such as the circuit model of power distribution network (PDN) of LSI, shape information about the package and board, electrical parameters of materials, and models of the components. It is difficult to make an efficient design when the specification or format of the design information is different in each part of the LPB, and the necessary parameters are not shared. When the format of the interface methods and models of the simulation are not consistent, the setup time and the cost of design/verification are enormous, which has become a barrier to cooperation in LPB design. The LPB Format will be evolved to create a mutual interface to enable a more efficient co-design environment.

IEEE Standard Format for LSI-Package-Board Interoperable Design

1. Overview

1.1 Scope

This standard defines a common interoperable format that will be used for the design of a) large-scale integration (LSI), b) packages for such LSI, and c) printed circuit boards on which the packaged LSIs are interconnected. Collectively, such designs are referred to as *LSI-Package-Board (LPB)* designs. The format provides a common way to specify information/data about the project management, netlists, components, design rules, and geometries used in LPB designs.

1.2 Purpose

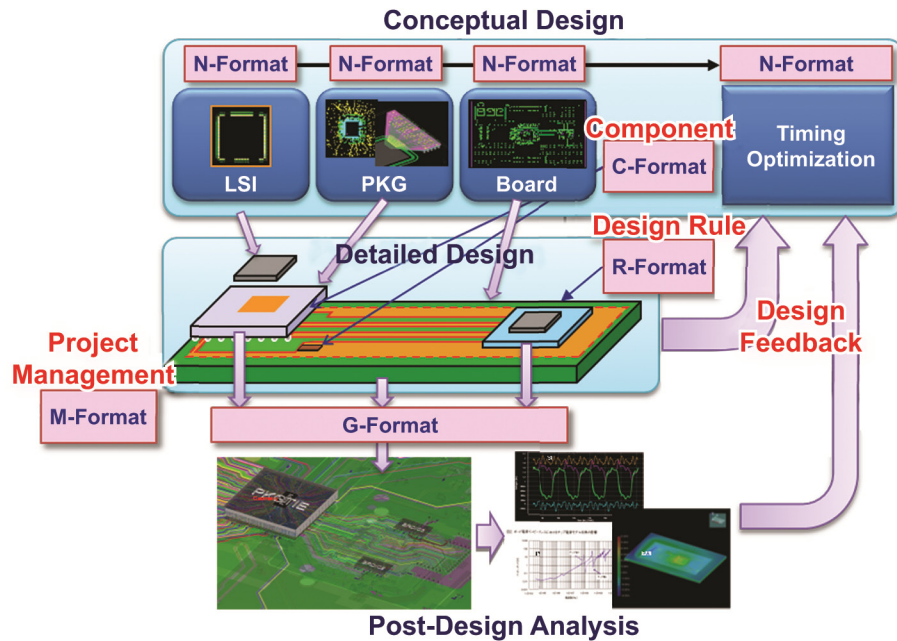
The general purpose of this standard is to develop a common format that LPB design tools can use to exchange information/data seamlessly, as opposed to having to work with multiple different input and output formats.

1.3 Key characteristics of the LSI-Package-Board Format

The LPB Format will facilitate the exchange of design information. This functionality provides the ability to plan the entire design at an early stage. In effect, post-design analysis will be possible throughout the entire LPB design process. Analysis of each part of the design can be examined in relation to all other parts of the design to determine the optimal point to give feedback for appropriate design changes throughout the LPB. This will promote the overall optimization of the design process.

The LPB Format is constructed out of the following five formats (see Figure 1):

- a) Project Management (M-Format)
- b) Netlist (N-Format)
- c) Component (C-Format)
- d) Design Rule (R-Format)
- e) Geometry (G-Format)



Reprinted with permission from JEITA.

Figure 1—LPB Format

Design time can be shortened by using the LPB Format. Traditionally, design starts immediately after separate planning for each individual component of the LPB. Therefore, information exchange among the separate design processes is limited. Trying to adjust the detailed design of one component to the detailed design of another component makes the entire design period take longer. Optimization also tends to be a separate process for each component of the LPB. By using the LPB Format for distributed information, each LPB engineer will be able to have the same understanding of the challenges at an early stage. As a result, adjustments at the conceptual design stage can be made, before detailed designs are developed. By making clear the overall LPB product specifications, the design target can be decided, and so the duration of individual designs can be shortened. Use of the LPB Format also helps to reduce the number of design iterations, because the design quality is enhanced. The designers can collect all information for simulation using the LPB formats, thereby reducing production time. The LPB Format can enable the entire analysis easily, so that sufficient verification can be done and the quality of the products can be improved. As a result, the period of adjustment in the set can be shortened and the time to market can be accelerated. With the LPB Format, the design method for one product can be applied to the design environment for next product in development.

1.4 Contents of this standard

The organization of the remainder of this standard is as follows:

- Clause 2 provides references to other applicable standards that are presumed or required for this standard.
- Clause 3 defines terms and acronyms used throughout the different specifications contained in this standard.
- Clause 4 describes the concepts of the LPB Format.
- Clause 5 describes the language basics for the LPB Format and its commands.
- Clause 6 describes common elements in the M-Format, C-Format, and R-Format.

- Clause 7 describes the M-Format.
- Clause 8 describes the C-Format.
- Clause 9 describes the R-Format.
- Clause 10 describes the N-Format.
- Clause 11 describes the G-Format.

1.5 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (shall equals is required to).^{1, 2}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (should equals is recommended that).

The word *may* is used to indicate a course of action permissible within the limits of the standard (may equals is permitted to).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (can equals is able to).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1364™, IEEE Standard for Verilog® Hardware Description Language.^{3, 4}

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁵

antipad: The clearance **hole** between a **via** and a no-connect metal layer, mainly used on the printed circuit **board** and large-scale integration (LSI) **package**. The shape of the antipad is mainly determined by the limit on the printed circuit board or LSI package manufacturing and is defined by the **padstack** in the R-Format file.

¹ The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

² The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

³ IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁴ The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

⁵ *IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>.

ball: *See:* **solder ball**.

ball grid array (BGA) package: A type of surface-mount **package** with one face covered (or partly covered) with **solder balls** arranged in a grid pattern.

board: The printed circuit board or printed wiring board.

bonding finger: The metal electrode on the surface of an large-scale integration (LSI) **package**. It connects the **bonding wire** to the routing pattern on the LSI package. In LSI-Package-Board (LPB) Format files, the shape of the bonding finger is defined by the **padstack** in the C-Format file.

bonding wire: A metal wire for connecting the **die** and **bonding finger**. In large-scale-integration-Package-Board (LPB) Format files, the shape of the bonding wire is defined in the R-Format file.

clock: The signal used in a synchronous circuit. All synchronous circuits use a clock signal to synchronize different parts of the circuit. In most cases, the part is register or flip-flop. The clock is distributed from a single source to registers or flip-flops and is required to arrive at all such parts at the same time.

common mode impedance: The impedance of a single transmission line when the two lines in a pair are driven with signals of the same amplitude and same polarity. Common (even) and differential (odd) modes are the two main modes of propagation of the signal through a coupled line pair.

component: A physical and logical construction having inputs to outputs. Large-scale integration (LSI) package or die and passive parts such as capacitors and connectors are called components.

component hole: A **hole** used for the attachment of component terminations to the printed **board** as well as for any electrical connection to the conductive pattern. *See also:* **hole**.

delay: The time interval between a step function change of the input signal level and the instant at which the magnitude of the output signal passes through a specified value that is close to its initial value. The switching time of the transistor and the propagation time of the signal through wiring.

die: A separated part (or whole) of a wafer intended to perform a function or functions in a device. A small block of semiconducting material, on which a given functional circuit is fabricated.

differential mode impedance: The impedance of a single transmission line when the two lines in a pair are driven with signals of the same amplitude and opposite polarity. Common (even) and differential (odd) modes are the two main modes of propagation of the signal through a coupled line pair.

differential signal: Differential signaling. A method of transmitting information electrically with two complementary signals sent on two paired wires, called a *differential pair*.

drill: The drill to be used when drilling the **via** hole connecting the layers of a multilayer printed circuit board.

driver: *See:* **sender**.

finger: *See:* **bonding finger**.

flipchip: A die that is flipped over so that its metal wiring faces down in order to mount the die to external circuitry (e.g., a circuit board or another chip or wafer).

flipchip pad: The contact pad of the **flipchip** surface.

guard shield: A barrier or enclosure provided for mechanical protection, which may also have the function of a screen, called a *GND shield* when put on a ground (GND) conductor. Its purpose is to limit the electromagnetic interference from other signals.

hole: Used for the conductive connection between each layers and for mounting components. *See also:* **component hole**, **landless hole**, **mounting hole**, **plated-through hole**, and **via hole**.

inout: A **port** having the function of both input and output. It is an input port where electromagnetic energy or signals may be received from an external circuit or device. It is an output port where electromagnetic energy or signals may be supplied to an external circuit or device.

land: The conductive pattern used for joining and connecting parts, the conductive pattern for surface mount pads and hole-mounted components, and the conductive pattern that covers a via hole.

landless hole: A **plated-through hole** without **land**. *See also:* **hole**.

line: A device connecting two points for the purpose of conveying electromagnetic energy between them. Electromagnetic energy may be extracted from or supplied to a line at an intermediate point. Examples of lines are two-wire line, polygon line, coaxial line, and waveguide.

mounting hole: A **hole** used for the mechanical mounting of a printed **board** or for mechanical attachment of components to the printed board. *See also:* **hole**.

net: The relative position of the ideal elements representing an electric network. The label in between interconnection of terminals. Although it is defined on the same hierarchy, there are also cases that indicate the connection regardless of hierarchy (for example, global net/definition).

package: An enclosure for one or more die, film elements, or other components, that allows electrical connection and provides mechanical and environmental protection. Types of packages include quad flat package (QFP), ball-grid array (BGA), wafer-level chip-scale package (WLCSP), multi-chip module (MCM), package on package (PoP), etc.

package mold: Protection of die by resin against stress, external force, water, static electricity, and foreign substances. A package mold contains resin, silica, carbon, and flame-retardant material.

package substrate: The same as that of a printed circuit board. It carries large-scale integration (LSI) and electrically connects **solder balls** with LSI.

pad: A metal electrode on the surface of a die, large-scale integration (LSI) package, or printed circuit board.

padstack: The combination of layers that constitute a **pad**.

physical design rule: A series of parameters provided by semiconductor manufacturers that enable the designer to verify the correctness of a mask set. A design rule set specifies certain geometric and connectivity restrictions to ensure sufficient margins to account for variability in semiconductor manufacturing processes, so as to ensure that most of the parts work correctly.

pin: A contact element intended to make electric engagement on its outer surface for mating with the inner surface of another contact element.

plated-through hole: A **hole** in which metal is deposited on the wall. *See also:* **hole**.

port: Access point or points to a module where signals may be supplied or received. The port may be physical, or conceptual as in circuit nodes.

power domain: A collection of instances that are treated as a group for power-management purposes. The instances of a power domain typically, but do not always, share a primary supply set. A power domain may also have additional supplies, including retention and isolation supplies.

receiver: A device that receives signals for interpretation and action.

reference point: A location with respect to which the dimension coordinates and placement of a shape are fixed.

sender: A device that generates and terminates signals. *Syn:* **driver**.

single-ended signal: A signal used for single-ended signaling, which is the method of transmitting signals over electrical connections. One electrical connection carries a varying voltage that represents the signal.

skew: A variation of a delay time by propagation of a signal, or an amount of gaps of the delay time between a reference signal and an object signal.

solder ball: A spherical solder that is used in the large-scale integration (LSI) package of a **ball grid array (BGA)** type and provides the contact between the **package** and the printed circuit **board**. The **pins** of the BGA package are placed in a grid pattern. Balls are mounted on each pin and used to solder the BGA to the printed circuit board. The internal circuit in a BGA exchanges signals and power with an external circuit on a printed circuit board through the ball. The shape of the ball is defined in the R-Format file.

stacked via: A structure that places a **via** on another via. *See also:* **via**.

sub-circuit: A sub-circuit expresses a specific circuit as one unit.

terminator: A device fitted to the end of a cable to ensure electrical connection with other parts of the system and to maintain the insulation up to the point of connection.

typ: An abbreviation of “typical,” used to express a representative or standard value.

via: One of the conductive parts forming a contact in between layers.

via hole: The conduction connectivity made through a **hole** in between layers. *See also:* **hole**.

void: A **hole** or cutout on a plane.

3.2 Acronyms and abbreviations

ACIS	A generic 3D modeling kernel developed by Spatial Corporation
ASCII	American Standard Code for Information Interchange
BGA	ball grid array
CPIP	Chip-Package Interface Protocol
dc	direct current
DDR	double data rate
DDR-SDRAM	double data rate synchronous dynamic random-access memory

DEF	Design Exchange Format
DXF	Drawing Exchange Format
EDA	electronic design automation
EMI	electromagnetic interference
GDS (GDS II)	Graphic Database System
GND	ground
HDL	hardware description language
IBIS	Input/output Buffer Information Specification
IBIS-ISS	IBIS Interconnect SPICE Subcircuit
IC	integrated circuit
ICEM	integrated circuit emission model
ICIM	integrated circuit immunity model
IEC	International Electrotechnical Commission
IGES	Initial Graphics Exchange Specification
I/O	input output
JEDEC	Joint Electron Device Engineering Council
JEITA	Japan Electronics and Information Technology Industries Association
JTAM	Japan Electronics and Information Technology Industries Association (JEITA) Thermal Accurate Model
LPB	LSI, package, and board; LSI-Package-Board
LSI	large-scale integration
MD5	message digest algorithm 5
NG	no good
PCB	printed-circuit-board (adjective)
PCIe	Peripheral Component Interconnect Express
PI	power integrity
PKG	package
PKI	public key infrastructure



POP, PoP	package on package
PWB	printed wiring board
RMS	root mean square
SAT	Standard ACIS Text file
Si2	Silicon Integration Initiative
SI	signal integrity
SMA	subminiature version A
SMD	surface mount device
SoC	System on Chip
SPICE	Simulation Program with Integrated Circuit Emphasis
STEP	Standard for the Exchange of Product model data
URI	Uniform Resource Identifier
VHDL	Very High Speed Integrated Circuits Hardware Description Language
XML	Extensible Markup Language
Xtal	crystal

4. Concept of the LPB Format

4.1 Technical background

The design margin for timing and noise is decreasing due to the higher speed of systems and the low voltage of the interface and power supply. Also, balancing design for both cost and performance is increasingly becoming important for cost competitiveness. In a conventional design, the large-scale integration (LSI), package and board (LPB) are designed with margin in accordance with individual design guidelines. However, it becomes difficult to provide design guidelines for each LPB part separately with the decreased design margin. Therefore, deciding the design target needs the cooperation of the designers of each part of the LPB. In other words, the innovation of deciding design guidelines by using simulation technology is needed in the system design process. To perform this task, a rapid and accurate simulation environment is necessary.

4.2 Conventional design

In conventional design, LPB design and sign-off analysis were performed for each design criterion. In other words, only partial optimization was done, so whole-system optimization and analysis was not performed at the initial design stage. The result was a lack of observation of physical phenomena throughout the whole system, appropriate design changes on each design site separately, and excessive estimation for design margin. These significantly influenced the quality and cost of the final product.

4.3 Common problems at the design site

4.3.1 Misunderstanding among designers

Misunderstanding among designers can occur when different designers perceive the same word to have different meanings, designers use the different words to describe the same phenomenon, and designers have different subjectivities, such as viewing the design from the top or bottom. These cases occur even if designers are in the same office, with the result that many designers have a bitter experience.

4.3.2 Lack of information for system design

Limitations and design margins among the LSI, package, and board may not be assumed at the time of the design of the individual components. Also, designers may face big obstacles after connecting individual components because they are uncertain about the limitations and margins of the others. For example, designing the package or board (wiring) will be difficult because of lack of information about the LSI pad assignment, the package ball assignment, and the part location on the board.

4.3.3 Waste of time that should be used for design

Because input and output formats are different for each electronic design automation (EDA) vendor, in some cases the designer needs to convert the format at the time of the delivery of the data from the different EDA vendors. Problems occur when the designer lacks information at the time of format conversion and has to make up for its lack later. In addition, the pin name and net names may be different for every LSI circuit, package, and board. The designer needs to adjust those data and spend painful time in correction.

4.4 Concept of LPB interoperable design

Interoperable design is the solution method for the problems with using conventional individual design methods for each LPB part. It is defined as the style that each LPB section uses to cooperate and design. At first, the designer makes minimum design guidelines to satisfy the product's performance standards by designing through the whole LPB using the interoperable design. After that, the designer designs each part individually in accordance with the design guidelines. After each individual design is completed, the designer performs a simulation for integrating that part with the whole LPB, confirms the performance of all products, and provides feedback on the designs for all other parts accurately.

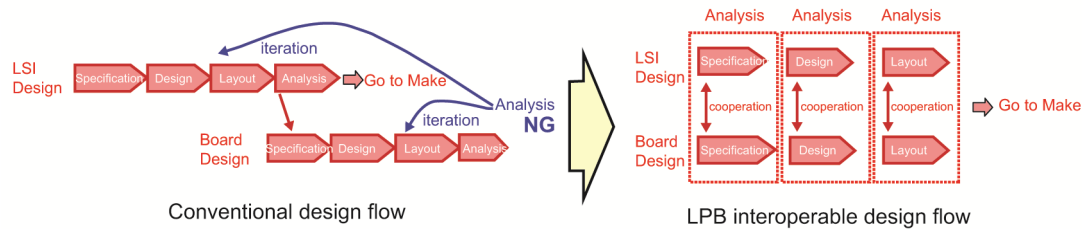
4.5 Value creation by LPB interoperable design

4.5.1 Effects of LPB interoperable design

LPB interoperable design is an appropriate method of design throughout a whole system for signal integrity and power integrity among LPB components. This technology is a solution for the problems discussed in 4.3. Use of LPB interoperable design makes it is possible to get the effect described in the following subclauses.

4.5.2 Reduction of development cost and time by design flow without iteration

When the designer finally connects the parts designed by partial optimization, an unexpected lack of design margin sometimes occurs. The designer then needs to discover which part of the LPB is causing the margin bottleneck. As a result, the designer may not only spend a lot of time but also need additional recovery costs for creating the needed design margin. In such cases, it is possible to reduce the development cost and time by starting LSI design, package design, and board design with each designer sharing information and cooperating from the point that the LSI design begins. Figure 2 compares conventional design flow and LPB interoperable design flow. The LPB interoperable flow reduces development costs and time.

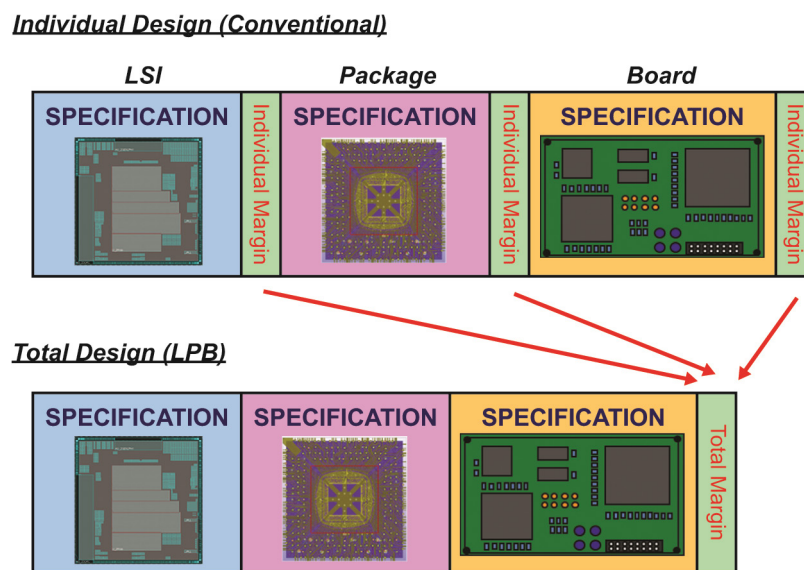


Reprinted with permission from JEITA.

Figure 2—Conventional design flow versus LPB interoperable design flow

4.5.3 Reduction of system total cost

In conventional design, because a designer designs each part of the LSI, package, and board separately with individual margins, the design margin of the product as whole system sometimes becomes excessive when all the parts are finally connected. It is possible to reduce the total cost using LPB interoperable design because the design margin for the whole design is controlled and held in total, including reduction of the number and cost of the parts used in the design. Figure 3 compares the design margins of conventional design flow and LPB interoperable design flow.



Reprinted with permission from JEITA.

Figure 3—LPB interoperable design can control design margin

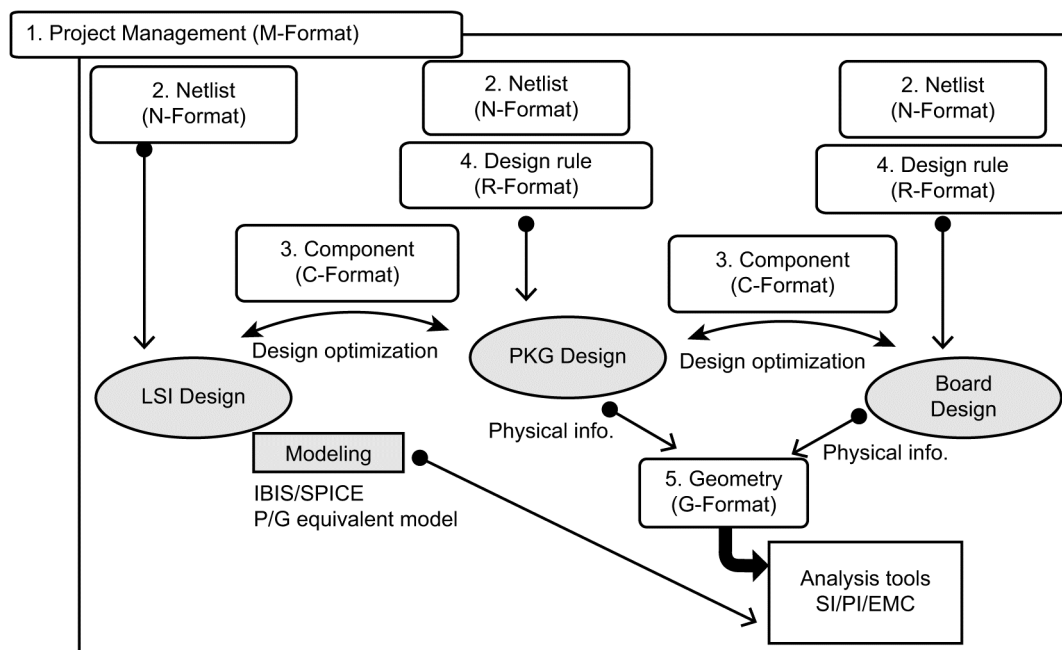
4.6 LPB Format

LPB Format solves the problems in conventional design methodologies, improves product quality, and reduces design time. LPB Format is a common language for describing the information required for design and verification. It reduces the preparation time of EDA tools in design teams. Furthermore, LPB can enable the sharing of ideas beyond the design teams. It can become a medium for information distribution in the supply chain. It prevents misunderstanding of design information and improves the information flow in the whole industry. As the result, quality, cost, and delivery time (QCD) will be improved.

4.7 Summary of LPB Format files

4.7.1 General

Figure 4 shows one example of how design information is exchanged using the LPB Format. In this example, three types of designers exist: the LSI designer, the package designer, and the board designer. All designers exchange design information using LPB Format standard files. By unifying the notation of the files for exchanging design information, it is possible to prevent misunderstandings and to automate the design tool settings.



Reprinted with permission from JEITA.

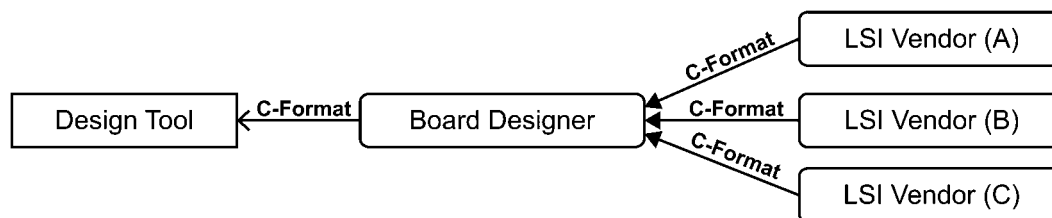
Figure 4—Example of exchanging design information using the LPB Format

4.7.2 C-Format

The first purpose of C-Format is to give a unified notation to the outer specifications for parts, such as the LSI, package, socket, and so forth. The outer specification is the information needed in order to use the parts, as follows:

- Physical shape of the parts
- Name and type of signal to be input to and output from the parts
- Input/output (I/O) specification, such as the physical shape and position of pins, or swappable pin definitions
- Design constraints, such as the upper limit of the delay or skew
- Design specifications, such as the input impedance of pins or power consumption

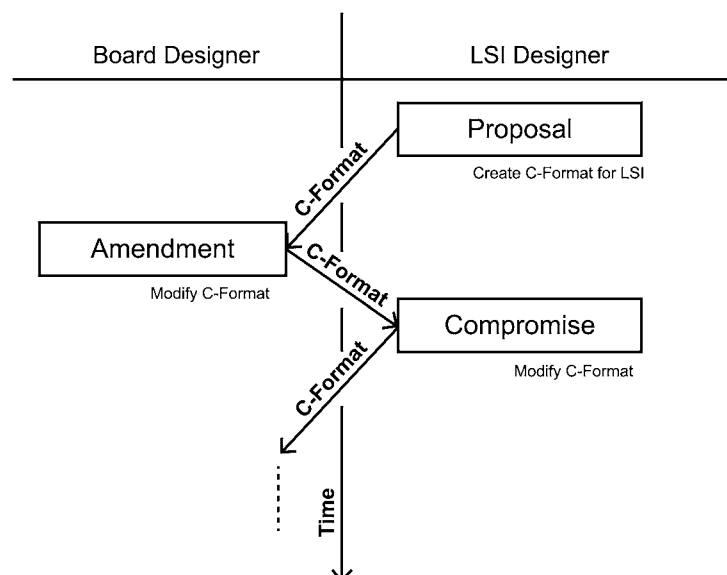
By unifying the notation with C-Format, it is possible to exchange information without misunderstanding. For example, a board designer should understand the specifications delivered from several LSI vendors and should set up the design tools based on this understanding. Human error may occur during this process if the specifications are described by each vendor's own notation. Using C-Format instead of the vendor's own notations, it is possible to set up the design tool automatically and to prevent human error due to misunderstanding. Figure 5 shows an example of the information flow of LSI specifications using C-Format. The LSI specifications are all provided by C-Format even if the vendors are different. The board designer can understand the LSI specifications by reading only one notation, and design tools can automatically set up themselves by entering C-Format.



Reprinted with permission from JEITA.

Figure 5—Example of information flow of LSI specifications using C-Format

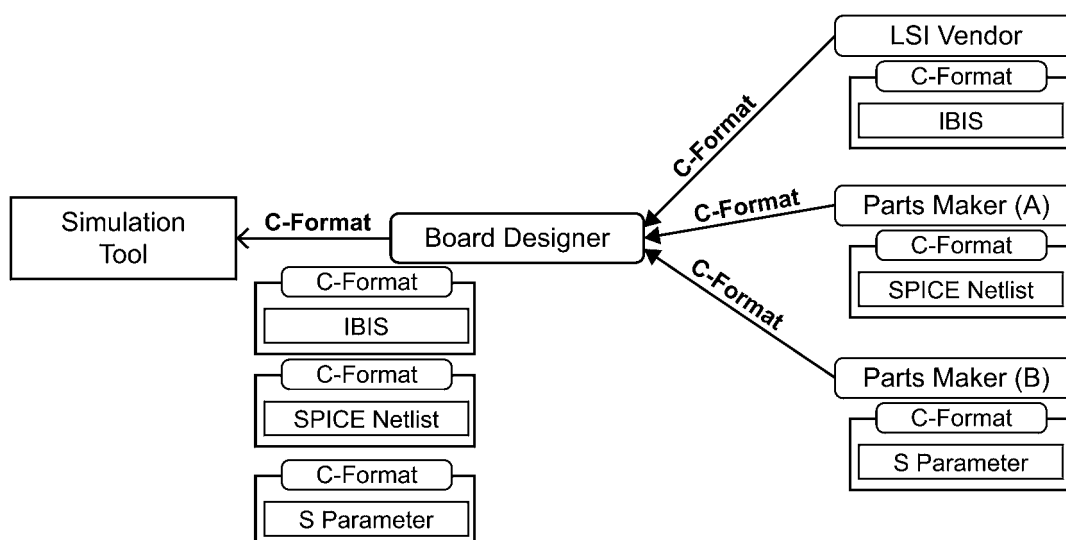
Unified notation also smoothly circulates information about specification changes. Figure 6 shows an example of interoperable design flow. A board designer and a LSI designer are collaborating. The information about the pin assignment of the LSI package is provided by the LSI designer. When the board designer wants to change the pin assignment of the LSI package, he or she modifies the provided C-Format file and returns it to the LSI designer. The use of C-Format prevents misunderstanding about the specification change and prevents human error that can occur due to specification changes.



Reprinted with permission from JEITA.

Figure 6—Example of interoperable design flow with C-Format

The second purpose of C-Format is to provide a unified interface to a simulation model such as a Simulated Program with Integrated Circuit Emphasis (SPICE) netlist, Input/output Buffer Information Specification (IBIS), S-parameters, etc. C-Format wraps these simulation model files and gives cross-references between the nodes of the simulation model and the physical ports of the parts. Simulation tools can plug in the simulation model automatically by entering the wrapped model file. Figure 7 shows an example of exchanging simulation models. Model files are provided with the C-Format file.

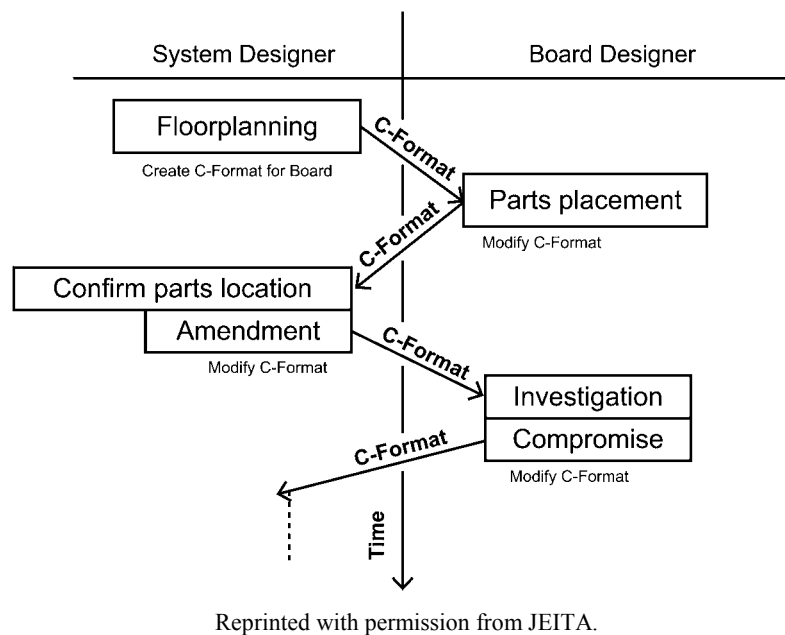


Reprinted with permission from JEITA.

Figure 7—Example of exchanging simulation models with C-Format

The third purpose of C-Format is to provide connection points to merge multiple layout data. C-Format wraps layout data files, such as Graphic Database System (GDS) II, and gives cross-references between ports in the C-Format file and objects in the layout data file. For example, when the C-Format file wraps a GDS II file, ports in C-Format are associated with the coordinates/layers in the GDS II file. When the C-Format file wraps a G-Format file, ports in the C-Format file are associated to the pin in the G-Format file. This feature can be used to merge two or more layout data designed by different design houses. Such situations can occur when the board designer wants to analyze the board with package layout data. The board designer receives the wrapped package layout data from the package designer and enters it in the analysis tools. The analysis tools can find the connection points using the cross-references defined in the C-Format file and merge the layout data of the printed circuit board and the package.

The fourth purpose of C-Format is to support the floorplan of the printed circuit board and package. C-Format includes placement information about parts. This feature can be used to communicate the floorplan information from the system designer to the board designer. Figure 8 shows an example of design flow when the system designer requests a printed circuit board design to the board designer. At the beginning of the work, the printed circuit board designer can set up the design tools based on the provided C-Format file. By unifying the notation of the floorplan, it is possible to exchange floorplan information without misunderstanding, and to set up the design tool automatically.



Reprinted with permission from JEITA.

Figure 8—Example of design flow between system designer and board designer

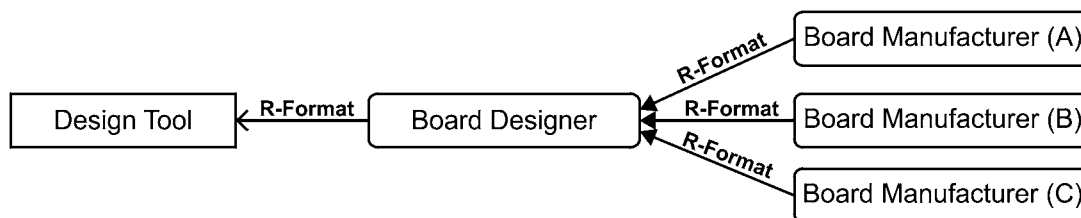
4.7.3 R-Format

The first purpose of R-Format is to unify the notation of the design rules of the printed circuit board and the package. The following design rules are defined in R-Format:

- Layer stackup of the package and the printed circuit board
- The thickness of the conductive layer/insulating layer
- The material used for each layer
- The material parameters, such as conductivity, dielectric constant, or loss tangent

- Line width and line space
- Via spacing
- Shape of vias

Normally, manufacturers of printed circuit boards and packages provide the design rules using their own notations. The designer has to understand the design rules described in several notations and set up the design tools, with the risk that human error may occur. However, using a unified notation such as R-Format makes it possible to set up the design tool automatically and to prevent misunderstanding. Figure 9 shows an example of exchanging design rules using R-Format. All manufacturers provide design rules using the R-Format unified notation. Design tools can set up themselves automatically by entering R-Format.



Reprinted with permission from JEITA.

Figure 9—Example of exchanging design rules with R-Format

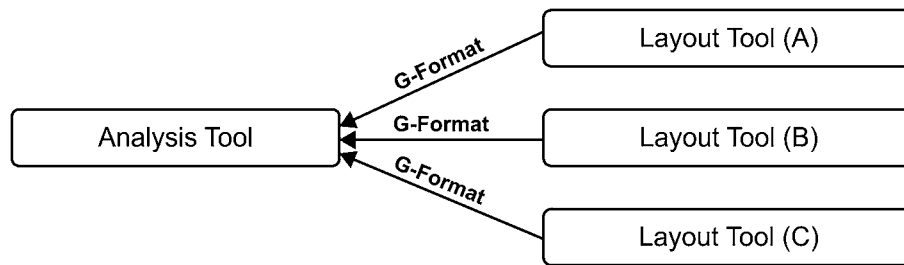
The second purpose of R-Format is to define the physical design constraints of the printed circuit board and the package. *Physical design constraints* mean the height limitation of mounted parts and the non-default design rule area.

4.7.4 G-Format

The purpose of G-Format is to unify the notation of the layout data of the layer stack-up structure, such as the printed circuit board or package. It is possible to exchange the layout data seamlessly between analysis tools and layout tools by the unified notation of the layout data. The G-Format file includes the following geometric information:

- Printed circuit board
- Layer stackup and physical parameters of the material
- Shape and location of the mounted parts
- Shape and location of the pins
- Route or pattern of the nets
- Shape and location of the vias
- Shape of the bonding wires

Figure 10 shows an example of exchanging layout data using G-Format. Analysis tools can set up by only entering G-Format if all layout tools output G-Format files.



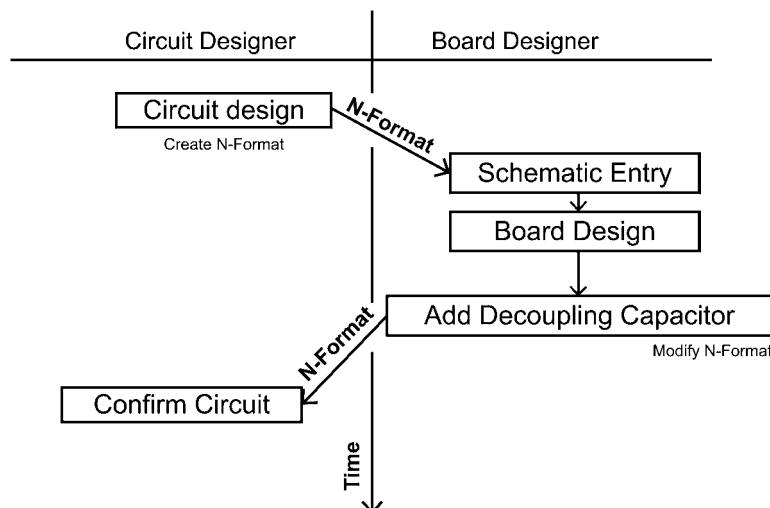
Reprinted with permission from JEITA.

Figure 10—Example of exchanging layout data with G-Format

4.7.5 N-Format

The purpose of N-Format is to unify the notation of the netlist that is used to design the printed circuit board and the package, where *the netlist* means the connectivity information between parts mounted on a printed circuit board or package. N-Format conforms to Verilog hardware description language (HDL) (IEEE Std 1364™⁶) and adds keywords to identify power and ground nets. Unifying notation of the netlist makes it possible to exchange the netlist seamlessly between circuit design tools and layout tools. For example, when the circuit designer orders the board design from the board designers, the circuit designers should exchange the netlist.

If the netlist is represented using several notations, there is a risk that human error may occur when entering the connectivity between parts into the design tools. However, by using the unified notation for netlist, it is possible to prevent human errors when setting up the design tools because all design tools can set up the connectivity between parts by supporting only one notation for the netlist file. Figure 11 shows an example of design flow using the N-Format file. The circuit modification is exchanged using N-Format.



Reprinted with permission from JEITA.

Figure 11—Example of design flow with N-Format

⁶ Information on references can be found in Clause 2.

4.7.6 M-Format

LPB Format files are continuously updated in accordance with the design progress. The purpose of M-Format is to manage the version of each file to prevent errors when exchanging LPB Format files.

5. Language basics

5.1 General

This subclause describes the conventions used in the syntax definitions of the LPB M-Format, C-Format, and R-Format.

5.2 Typographic and syntax conventions

The following list describes the syntax conventions:

`text`

The `monospace` font is used to indicate the attributes or elements that shall be typed literally.

italic

The *italic* font is used to indicate the user-defined information for which shall be substituted a name or value.

|

Vertical bars separate possible choices for a single attribute or element. They take precedence over any other character.

[]

Brackets denote optional attributes or elements. When used with vertical bars, they enclose a list of choices.

[]...

Brackets followed by three dots indicate that there shall be specified zero or more attributes or elements. When used with vertical bars, they enclose a list of choices.

{ }...

Braces followed by three dots indicate that there shall be specified one or more attributes or elements. When used with vertical bars, they enclose a list of choices.

...

Three dots indicate that the previous value could be repeated.



All of the strings in the M-Format, C-Format, and R-Format are case sensitive.

NOTE—All code examples in this standard are written in monospace font.⁷

5.3 Character information

Unless otherwise noted, only ASCII characters, as defined in ANSI Standard X3.4-1986, shall be used in LPB Format file. The use of characters with codes greater than hexadecimal 0x7E is not allowed. Also, ASCII control characters (those numerically less than hexadecimal 0x20) are not allowed, except for tabs or in a line termination sequence.

5.4 Notation of the floating point number

The following format is used to denote floating point numbers.

```
[ -|+ ] <mantissa> [ { e | E } [ -|+ ] <exponent> ]
```

<mantissa>

An integer or fractional number. Only numeric character (0 to 9) and decimal separator character (period) shall be used.

<exponent>

Only integer shall be used for the exponent (the power of 10). Shall not contain a decimal separator character.

5.5 File naming definitions

The following defines a rule of file referencing.

- **directory**: A location for storing files on a hierarchical file system.
- **root directory**: A standard designation for the root of a file system. In Windows, it includes the driver letter (line C:) and in Linux, it includes the root slash (/).
- **current directory**: A directory that holds current LPB Format file.
- **path**: A route to a specific file that is represented by a sequence of root directory (optional) and directories.
- **absolute path**: An absolute path is a path that starts from the root directory of a file system. It locates the specified file regardless of the current directory.
- **relative path**: The relative path is not starting from root directory. It locates a file relative to the current directory.
- **file reference**: A reference to a file, expressed as either a simple file name or including relative or absolute path. Simple file name means that the file is located in the current directory.

⁷ Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

6. Common elements in M-Format, C-Format, and R-Format

6.1 General

This clause provides commonly used elements in M-Format, C-Format, and R-Format.

M-Format, C-Format, and R-Format use the World Wide Web Consortium standard for the XML version 1.0 [B21]. M-Format, C-Format, and R-Format are structured by several elements. The name of an element is enclosed in triangular brackets < >.

```
<element_name>
```

An element can contain multiple elements as its children. An element start with <element_name> and end with </element_name> if the element contains children.

```
<element_name>...</element_name>
```

If an element does not contain children, the element simply expresses as <element_name/>. The order of elements is defined by syntax rule of M-Format, C-Format, and R-Format.

An element can have one or more attributes. An attribute is represented by name, and a value enclosed in double quotation. Attribute values shall be numeric value or string value. Numeric value shall be notated according to the rules defined in 5.4. Unless restricted by keyword, string value is free strings.

```
attribute_name="attribute_value"
```

The attributes can appear in any order and shall be unique.

6.2 The <extensions> element

6.2.1 General

The <extensions> element is a container of the user's own extension. It is used to add the user's own specifications to M-Format, C-Format, and R-Format. All user extensions are defined using the namespace (Name spaces in XML 1.0 [B16]). Any description can be added to user extension area as long as they accord to the XML format. The user extension declaration has the following syntax.

```
<extensions xmlns:user_prefix="URI">  
  {<user_prefix:user_own_tag {user_own_attribute="any_string"}... />}...  
</extensions>
```

If multiple users add the their own specifications individually, the keywords might conflict with each other. Use the namespace identified by URI reference to prevent the keyword conflicting. The most important thing of URI is that it is unique. The users who use user extension area can guarantee the uniqueness of the URI by registering the domain name included in URI to internet domain management organization. It is the responsibility of the user extension to guarantee the uniqueness of the string used after the domain name. The LPB Format use the URI as an identifier. Including a specific domain name is an idea to ensure uniqueness of URI strings. The URI specified here can be fictitious as long as it is unique. Only graphic characters in ANSI Standard X3.4-1986 (codes 0x21 to 0x7E) shall be used for URI.

6.2.2 Example

The following is an example of the <extensions> element in use.

```
<header project="test project" design_revison="1.1">
  <extensions xmlns:A="http://www.A.com">
    <A:own_user_area1 myattribute1="xxx" myattribute2="yyy"/>
    <A:own_user_area2 myattribute3="xxx" myattribute4="yyy"/>
  </extensions>
  <extensions xmlns:B="http://www.B.com">
    <B:own_user_area1 myattribute1="xxx" myattribute2="yyy"/>
    <B:own_user_area2 myattribute3="xxx" myattribute4="yyy"/>
  </extensions>
</header>
```

6.3 The <header> element

6.3.1 General

The <header> element defines the management information of the M-Format, R-Format, and C-Format files.

```
<header
    project="project_name"
    design_revison="revision_number"
    [permission="edit_permission"]
    [date="date"]
    [author="owner_of_this_document"]
    [email="email_address"]
    [company="company_name"]
    [comment="any_comment"]
>
    [<extensions> element]...
</header>
```

By processing the <header> element, you can get the project name (project), creation date (date), information about the author (author, email, and company), and a control number (design_revison).

6.3.2 Attribute definitions

The attributes of the <header> element are defined as follows.

project

This attribute specifies the name of the project. All LPB Format files in a project shall have the same project name.

design_revison

This attribute specifies the revision number of each LPB Format file. The revision number shall be increased appropriately when modifying the LPB Format file.

permission

This attribute specifies the edit permission of each LPB Format file. The value shall be one of the following:

EDITABLE	Permit editing.
LOCKED	Prohibit editing.

If permission attribute is not specified, EDITABLE is set as the default.

date

This attribute specifies the creation or modification date of the LPB Format file.

author

This attribute specifies the name of the author of the LPB Format file.

email

This attribute specifies the email address of the author.

company

This attribute specifies the organization to which the author of this design file belongs.

comment

This attribute specifies an arbitrary text string used as a comment.

6.3.3 Element content

The <header> element can contain the following element:

<extensions>

6.3.4 Example

The following is an example of the <header> element in use.

```
<header
  project="JEITA_LPB_SAMPLE_PROJECT"
  design_revision="1.3"
  date="20190101"
  permission="LOCKED"
  author="yyyyy xxxxx"
  email="xxxx@jeita.jp"
  company="JEITA"
  comment="This is a sample code of LPB format."
>
<extensions xmlns:LPB_FORUM="http://www.lpb-forum.com">
  <LPB_FORUM:header_ext comment="this is a sample of user extension area"/>
</extensions>
</header>
```

6.4 The <global> element

6.4.1 General

The <global> element defines the unit system, basic shapes, and padstack to be used throughout the R-Format and C-Format files.

```
<global>
    <unit> element
    [<shape> element]...
    [<padstack_def> element]...
    [<extensions> element]...
</global>
```

The scope of the defined variables is limited to the file in which it is declared. The content of the <global> element consists of one <unit> element and one or zero <shape> and <padstack_def> elements.

6.4.2 Element contents

The <global> element can contain the following elements:

```
<unit>
<shape>
<padstack_def>
<extensions>
```

6.4.3 The <unit> element

The <unit> element defines the unit system to be used in LPB Format file.

```
<unit>
    [<distance> element]
    [<angle> element]
    [<area> element]
    [<time> element]
    [<resistance> element]
    [<capacitance> element]
    [<resistivity> element]
    [<temperature> element]
    [<voltage> element]
    [<current> element]
    [<power> element]
    [<inductance> element]
    [<frequency> element]
    [<impedance> element]
    [<thermal_conductivity> element]
    [<specific_heat_capacity> element]
    [<density> element]
    [<thermal_diffusivity> element]
    [<coefficient_of_thermal_expansion> element]
    [<dynamic_viscosity> element]
    [<extensions> element]...
</unit>
```

The value for each indicates how the numbers in the LPB Format file shall be interpreted.

6.4.3.1 Element content

6.4.3.1.1 The <distance> element

The <distance> element defines length units to be used in the LPB Format file.

```
<distance unit="length_unit"/>
```

The *length_unit* shall be one of the following values:

pm	picometers
nm	nanometers
um	micrometers
mm	millimeters
m	meters
inch	inches
mil	mil

If <distance> element is not specified, m is set as the default.

6.4.3.1.2 The <angle> element

The <angle> element defines a unit system for angles.

```
<angle unit="unit_of_angle"/>
```

The *unit_of_angle* shall be either of the following values:

degree
radian

If <angle> element is not specified, degree is set as the default.

6.4.3.1.3 The <area> element

The <area> element defines a unit of area.

```
<area unit="unit_of_area" />
```

The *unit_of_area* shall be one of the following values:

pm2	square picometer
nm2	square nanometer
um2	square micrometer
mm2	square millimeter
m2	square meter
i2	square inch

If <area> element is not specified, m2 is set as the default.

6.4.3.1.4 The <time> element

The <time> element defines a unit of time in seconds.

```
<time unit="unit_of_time" />
```

The *unit_of_time* shall be one of the following values:

ps	picosecond
ns	nanosecond
us	microsecond
ms	millisecond
s	second

If <time> element is not specified, s is set as the default.

6.4.3.1.5 The <resistance> element

The <resistance> element defines a unit of resistance.

```
<resistance unit="unit_of_resistance" />
```

The *unit_of_resistance* shall be one of the following values:

fohm	femtoohm
pohm	picoohm
nohm	nanoohm
uohm	microohm
mohm	milliohm
ohm	ohm
kohm	kiloohm
Mohm	megaohm
Gohm	gigaohm

If <resistance> element is not specified, ohm is set as the default.

6.4.3.1.6 The <capacitance> element

The <capacitance> element defines a unit of capacitance.

```
<capacitance unit="unit_of_capacitance" />
```

The *unit_of_capacitance* shall be one of the following values:

fF	femtofarad
pF	picofarad
nF	nanofarad
uF	microfarad
mF	millifarad
F	farad
kF	kilofarad

MF	megafarad
GF	gigafarad

If <capacitance> element is not specified, F is set as the default.

6.4.3.1.7 The <resistivity> element

The <resistivity> element defines a unit of resistivity.

```
<resistivity unit="unit_of_resistivity" />
```

The *unit_of_resistivity* shall be one of the following values:

fohmm	femtoohm meter
nohmm	nanooohm meter
pohmm	picoohm meter
uohmm	microohm meter
mohmm	milliohm meter
ohmm	ohm meter
kohmm	kiloohm meter
Mohmm	megaohm meter
Gohmm	gigaohm meter

If <resistivity> element is not specified, ohmm is set as the default.

6.4.3.1.8 The <temperature> element

The <temperature> element defines a unit system for temperature.

```
<temperature unit="unit_of_temperature" />
```

The *unit_of_temperature* shall be either of the following values:

C	celsius
K	kelvin

If <temperature> element is not specified, C is set as the default.

6.4.3.1.9 The <voltage> element

The <voltage> element defines a unit of voltage.

```
<voltage unit="unit_of_voltage" />
```

The *unit_of_voltage* shall be one of the following values:

pV	picovolt
nV	nanovolt
uV	microvolt
mV	millivolt

V	volt
kV	kilovolt

If `<voltage>` element is not specified, V is set as the default.

6.4.3.1.10 The `<current>` element

The `<current>` element defines a unit of electric current.

```
<current unit="unit_of_current" />
```

The `unit_of_current` shall be one of the following values:

pA	picoampere
nA	nanoampere
uA	microampere
mA	milliampere
A	ampere
kA	kiloampere

If `<current>` element is not specified, A is set as the default.

6.4.3.1.11 The `<power>` element

The `<power>` element defines a unit of power consumption.

```
<power unit="unit_of_power" />
```

The `unit_of_power` shall be one of the following values:

pW	picowatt
nW	nanowatt
uW	microwatt
mW	milliwatt
W	watt
kW	kilowatt

If `<power>` element is not specified, W is set as the default.

6.4.3.1.12 The `<inductance>` element

The `<inductance>` element defines a unit of inductance.

```
<inductance unit="unit_of_inductance" />
```

The `unit_of_inductance` shall be one of the following values:

fH	femtohenry
pH	picohenry
nH	nanohenry

uH	microhenry
mH	millihenry
H	henry
kH	kiloHenry

If `<inductance>` element is not specified, H is set as the default.

6.4.3.1.13 The `<frequency>` element

The `<frequency>` element defines a unit of frequency.

```
<frequency unit="unit_of_frequency" />
```

The `unit_of_frequency` shall be one of the following values:

uHz	microhertz
mHz	millihertz
Hz	hertz
kHz	kilohertz
MHz	megahertz
GHz	gigahertz

If `<frequency>` element is not specified, Hz is set as the default.

6.4.3.1.14 The `<impedance>` element

The `<impedance>` element defines a unit of impedance.

```
<impedance unit="unit_of_impedance" />
```

The `unit_of_impedance` shall be one of the following values:

fohm	femtoohm
pohm	picoohm
nohm	nanoohm
uohm	microohm
mohm	milliohm
ohm	ohm
kohm	kiloohm
Mohm	megaohm

If `<impedance>` element is not specified, ohm is set as the default.

6.4.3.1.15 The `<thermal_conductivity>` element

The `<thermal_conductivity>` element defines a unit of thermal conductivity.

```
<thermal_conductivity unit="unit_of_thermal_conductivity" />
```

The *unit_of_thermal_conductivity* shall be one of the following values:

$$\begin{array}{l} \text{W/ (m}\cdot\text{K)} \quad \frac{\text{watt}}{(\text{meter} \times \text{kelvin})} \\ \text{W/ (cm}\cdot\text{K)} \quad \frac{\text{watt}}{(\text{centimeter} \times \text{kelvin})} \end{array}$$

If <thermal_conductivity> element is not specified, W/ (m*K) is set as the default.

6.4.3.1.16 The <specific_heat_capacity> element

The <specific_heat_capacity> element defines a unit of specific heat capacity.

```
<specific_heat_capacity unit="unit_of_thermal_specific_heat_capacity" />
```

The *unit_of_specific_heat_capacity* shall be one of the following values:

$$\begin{array}{l} \text{J/ (g}\cdot\text{K)} \quad \frac{\text{joule}}{(\text{gram} \times \text{kelvin})} \\ \text{J/ (kg}\cdot\text{K)} \quad \frac{\text{joule}}{(\text{kilogram} \times \text{kelvin})} \end{array}$$

If <specific_heat_capacity> element is not specified, J/ (kg*K) is set as the default.

6.4.3.1.17 The <density> element

The <density> element defines a unit of density.

```
<density unit="unit_of_density" />
```

The *unit_of_density* shall be one of the following values:

$$\begin{array}{l} \text{kg/m}^3 \quad \frac{\text{kilogram}}{\text{cubic meter}} \\ \text{g/m}^3 \quad \frac{\text{gram}}{\text{cubic meter}} \\ \text{g/cm}^3 \quad \frac{\text{gram}}{\text{cubic centimeter}} \end{array}$$

If <density> element is not specified, kg/m³ is set as the default.

6.4.3.1.18 The <thermal_diffusivity> element

The <thermal_diffusivity> defines the unit of the coefficient of thermal diffusivity.

```
<thermal_diffusivity unit="unit_of_thermal_diffusivity">
```

The *unit_of_thermal_diffusivity* shall be one of the following values:

m ² /s	$\frac{\text{square meter}}{\text{seconds}}$
cm ² /s	$\frac{\text{square centimeter}}{\text{seconds}}$

If <thermal_diffusivity> element is not specified, m²/s is set as the default.

6.4.3.1.19 The <coefficient_of_thermal_expansion> element

The <coefficient_of_thermal_expansion> defines the unit of the coefficient of thermal expansion.

```
<coefficient_of_thermal_expansion
unit="unit_of_coefficient_of_thermal_expansion">
```

The *unit_of_coefficient_of_thermal_expansion* shall be one of the following values:

1/K	$\frac{1}{\text{kelvin}}$
1/C	$\frac{1}{\text{celcius}}$

If <coefficient_of_thermal_expansion> element is not specified, 1/K is set as the default.

6.4.3.1.20 The <dynamic_viscosity> element

The <dynamic_viscosity> defines the unit of dynamic viscosity.

```
<dynamic_viscosity unit="unit_of_dynamic_viscosity">
```

The *unit_of_dynamic_viscosity* shall be one of the following values:

kg/ms	$\frac{\text{kilogram}}{\text{meter seconds}}$
Pas	pascal seconds $\left(\text{Pas} = \frac{\text{kilogram}}{\text{meter seconds}} \right)$
P	poise $\left(\text{P} = \frac{0.1 \text{ kilogram}}{\text{meter seconds}} \right)$
cP	centi poise $\left(\text{cP} = \frac{0.001 \text{ kilogram}}{\text{meter seconds}} \right)$

If <dynamic_viscosity> element is not specified, kg/ms is set as the default.

6.4.3.2 Example

The following is an example of the `<unit>` elements in use.

```
<unit>
  <distance unit="um" />
  <angle unit="degree" />
  <area unit="um2" />
  <time unit="ps" />
  <resistivity unit="ohmm"/>
  <temperature unit="C" />
  <voltage unit="V" />
  <capacitance unit="pF" />
  <resistance unit="mohm" />
  <power unit="mW" />
  <inductance unit="nH" />
  <frequency unit="MHz" />
  <impedance unit="ohm" />
  <thermal_conductivity unit="W/(m*K)" />
  <specific_heat_capacity unit="J/(kg*K)" />
  <density unit="kg/m3" />
</unit>
```

6.4.4 The `<shape>` element

6.4.4.1 General

The `<shape>` element defines basic shapes those are referenced by other attributes and elements, such as a `<padstack_def>` element.

```
<shape>
    [<rectangle> element]...
    [<circle> element]...
    [<polygon> element]...
    [<extensions> element]...
</shape>
```

The scope of the defined shapes is limited to the file in which it is declared. The content of the `<shape>` element consists of zero or more `<rectangle>`, `<circle>`, and/or `<polygon>` elements.

6.4.4.2 Element content

The `<shape>` element can contain the following elements:

```
<rectangle>
<circle>
<polygon>
<extensions>
```

6.4.4.3 The <rectangle> element

6.4.4.3.1 General

The <rectangle> element defines the shape of the rectangle with width, height, and rotation angle.

```
<rectangle
    id="identifier"
    width="width"
    height="height"
    [angle="rotation_angle"]
/>
```

The reference point of the defined rectangle is the center of rectangle. It is used for rotation and placement position of the rectangle.

6.4.4.3.2 Attribute definitions

The attributes of the <rectangle> element are defined as follows.

id

This attribute specifies the unique identifier that is used to reference the shape separately from other attributes and elements.

width

This attribute specifies the width of the rectangle. The unit of distance is defined by the <distance> element in the <unit> element.

height

This attribute specifies the height of the rectangle. The unit of distance is defined by the <distance> element in the <unit> element.

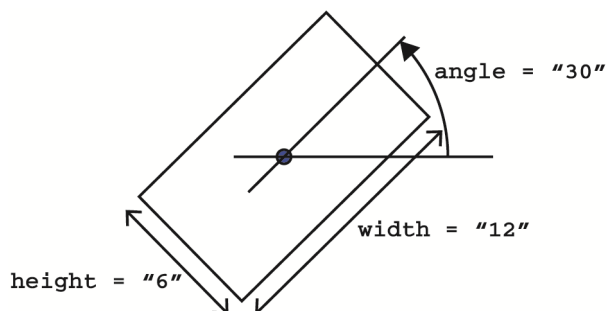
angle

This attribute specifies the angle of the counterclockwise rotation with respect to the center of the rectangle. If an angle is not specified, zero is set as the default. The unit of the rotation angle is defined by the <angle> element in the <unit> element.

6.4.4.3.3 Example

The rectangle shape in Figure 12 is represented by the following code:

```
<rectangle id="bfrect10" width="12" height="6" angle="30"/>
```



Reprinted with permission from JEITA.

Figure 12—Example of a rectangle shape

6.4.4.4 The <circle> element

6.4.4.4.1 General

The <circle> element defines the shape of a circle by diameter.

```
<circle
    id="identifier"
    diameter="diameter"
/>
```

The reference point of the circle is at the center.

6.4.4.4.2 Attribute definitions

The attributes of the <circle> element are defined as follows.

id

This attribute specifies the unique identifier that is used to reference the shape separately from other attributes and elements.

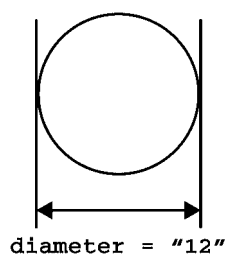
diameter

This attribute specifies the diameter of the circle. The unit of distance is defined by the <distance> element in the <unit> element.

6.4.4.4.3 Example

The circle shape in Figure 13 is represented by the following code:

```
<circle id="viapad12" diameter="12"/>
```



Reprinted with permission from JEITA.

Figure 13—Example of a circle shape

6.4.4.5 The <polygon> element

6.4.4.5.1 General

The <polygon> element defines the shape of a closed polygon.

```
<polygon
    id="Identifier"
    points="x1, y1, x2, y2, x3, y3, x4, y4 ..."
    [angle="rotation_angle"]
/>
```

The reference point of the polygon is set to (0, 0).

6.4.4.5.2 Attribute definitions

The attributes of the <polygon> element are defined as follows.

id

This attribute specifies the unique identifier that is used to reference the shape separately from other attributes and elements.

points

This attribute specifies a sequence of at least four points to generate a closed polygon. The last point and first point shall be the same. Points are separated by commas (,), and each point is a pair of XY coordinates separated by a comma (,). The unit of distance is defined by the <distance> element in the <unit> element.

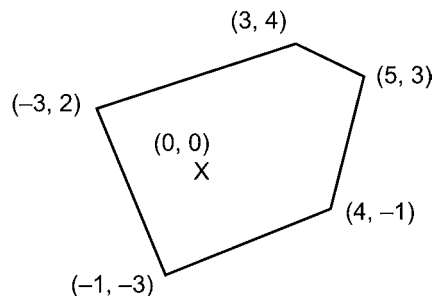
angle

This attribute specifies the angle of the counterclockwise rotation with respect to the reference point of the polygon. If the angle is not specified, zero is set as the default. The unit of the rotation angle is defined by the <angle> element in the <unit> element.

6.4.4.5.3 Example

The polygon shape in Figure 14 is represented by the following code:

```
<polygon id="X1" points="5,3,3,4,-3,2,-1,-3,4,-1,5,3"/>
```

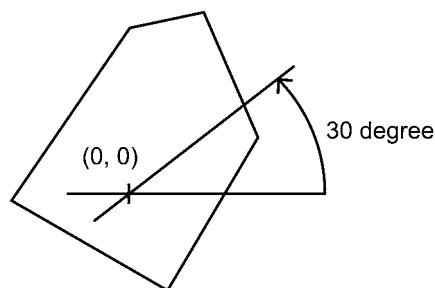


Reprinted with permission from JEITA.

Figure 14—Example of a polygon shape

The polygon shape in Figure 15 is represented by the following code:

```
<polygon id="X2" points="5,3,3,4,-3,2,-1,3,4,-1,5,3" angle="30"/>
```



Reprinted with permission from JEITA.

Figure 15—Example of the rotation of a polygon shape

6.4.4.6 Example

The following is an example of the <shape> elements in use.

```
<shape>
  <rectangle id="R1" width="100" height="10" />
  <circle id="C1" diameter="20" />
  <polygon id="P1" points="-10,-10,10,-10,10,10,-10,10,-10,-10" />
  <circle id="VLLPAD" diameter="500" />
  <circle id="VSLPAD" diameter="400" />
  <circle id="VLSPAD" diameter="700" />
  <circle id="VSAPAD" diameter="600" />
  <circle id="SHP.10" diameter="200" />
  <rectangle id="SHP.21" width="7500" height="10900" />
  <polygon id="SQRT" points="10,10,20,10,20,20,10,20,10,10" />
</shape>
```

6.4.5 The <padstack_def> element

6.4.5.1 General

The <padstack_def> element defines padstacks that are used by other attributes and elements.

```
<padstack_def>
    {<padstack> element}...
    [<extensions> element]...
</padstack_def>
```

The <padstack_def> element consists of one or more <padstack> elements and zero or more <extensions> elements.

6.4.5.2 Element contents

The <padstack_def> element can contain the following elements:

```
<padstack>
<extensions>
```

6.4.5.3 The <padstack> element

6.4.5.3.1 General

The <padstack> element defines the geometrical description of each pin or via of the printed wiring board (PWB) or package.

```
<padstack
    id="identifier"
>
    {<ref_shape> element}...
    [<extensions> element]...
</padstack>
```

The reference point of the padstack is at local origin (0, 0). The scope of the padstack definitions is limited to the file in which it is declared. The content of the <padstack_def> element consists of one or more <ref_shape> element and zero or more <extensions> elements.

6.4.5.3.2 Attribute definition

The attribute of the <padstack> element is defined as follows.

id

This attribute specifies the unique identifier that is used to reference the padstack from other attributes and elements.

6.4.5.4 Element contents

The <padstack> element can contain the following elements:

```
<ref_shape>  
<extensions>
```

6.4.5.4.1 The <ref_shape> element

6.4.5.4.1.1 General

The <ref_shape> element references the shape that constructs a padstack.

```
<ref_shape  
    shape_id="identifier_of_referenced_shape"  
    [type="shape_type"]  
    [x="x_coordinate" y="y_coordinate"]  
    [angle="rotation_angle"]  
    [layer="shape_placed_layer"]  
    [pad_layer="pad_placed_layer"]  
>
```

The referenced shape shall be defined at the <shape> element in the same file.

6.4.5.4.1.2 Attribute definitions

The attributes of the <ref_shape> element are defined as follows.

shape_id

This attribute specifies the identifier of the predefined shape that is defined at the <shape> element. The referenced shape shall be defined at the <shape> element in the same file.

type

This attribute specifies how the shape is used. Figure 16 shows an example when the shape is used for via structure. The value shall be one of the followings:

Antipad	Used as a shape of clearance, or antipad
NonConnection	Used as a shape of the nonconnection land
Land	Used as a shape of the normal land
Drill	Used as a shape of the drill; the outside diameter of the via
Hole	Used as a shape of the hole; the inside diameter of the via
SolderMask	Used as a shape of the solder mask
Resist	Used as a shape of solder resist