Soft IP Tagging
Version 1.0
February 2013

Accellera Systems Initiative, 1370 Trancas Street #163, Napa, CA 94558, USA.

Notices

Accellera Systems Initiative Standards documents are developed within Accellera Systems Initiative and the Technical Committees and Working Groups of Accellera Systems Initiative, Inc. Accellera Systems Initiative develops its standards through a consensus development process, approved by its members and board of directors, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of Accellera and serve without compensation. While Accellera Systems Initiative administers the process and establishes rules to promote fairness in the consensus development process, Accellera Systems Initiative does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an Accellera Systems Initiative Standard is wholly voluntary. Accellera Systems Initiative disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other Accellera Systems Initiative Standard document.

Accellera Systems Initiative does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or suitability for a specific purpose, or that the use of the material contained herein is free from patent infringement. Accellera Systems Initiative Standards documents are supplied "**AS IS**."

The existence of an Accellera Systems Initiative 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 an Accellera Systems Initiative Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change due to developments in the state of the art and comments received from users of the standard. Every Accellera Systems Initiative Standard is subjected to review periodically for revision and update. Users are cautioned to check to determine that they have the latest edition of any Accellera Systems Initiative Standard.

In publishing and making this document available, Accellera Systems Initiative is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is Accellera Systems Initiative undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other Accellera Systems Initiative Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of Accellera Systems Initiative, Accellera Systems Initiative will initiate reasonable action to prepare appropriate responses. Since Accellera Systems Initiative Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, Accellera Systems Initiative and the members of its Technical Committees and Working Groups are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of Accellera Systems Initiative Standards are welcome from any interested party, regardless of membership affiliation with Accellera Systems Initiative. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Accellera Systems Initiative
1370 Trancas Street #163
Napa, CA 94558
USA

Note: 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 with respect to the existence or validity of any patent rights in connection therewith. Accellera Systems Initiative shall not be responsible for identifying patents

for which a license may be required by an Accellera Systems Initiative standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Accellera Systems Initiative is the sole entity that may authorize the use of Accellera Systems Initiative -owned certification marks and/or trademarks to indicate compliance with the materials set forth herein.

Suggestions for improvements to the Soft IP Tagging are welcome. They should be sent to the group's email Reflector:

tagging@lists.accellera.org

The current Soft IP Tagging web page is:

www.accellera.org/activities/committees/ip-tagging

1. Overview

Expanded use of Soft IP allows more flexible use of design libraries, foundry sources, and methodologies. However, this flexibility also creates the problem of an IP becoming absorbed into the overall chip design at some stage of the design process. For example, a Verilog file (as an IP) might be presented as source to a customer. This Verilog file could then be synthesized, timed, placed, and routed along with the customer's own logic. When viewing the final netlist or GDS II files, there would be no means of discerning the source, version, date, manufacturer, or even the inclusion of the IP in the SoC. This Soft IP Tagging specification provides a means of retaining information about the IP all the way through the design process, from source to GDSII. It takes identification information from the source file, for example in Verilog, VHDL, SystemVerilog, or SystemC, and passes it along through each of the design steps. By this means, customers can determine if an IP is contained within the chip and if it is the correct version.

1.1.Scope

This specification updates the existing soft IP tagging standard to align with the hard IP tagging approach currently in wide use such that changes to EDA tools are not required to enable data-driven tracking of all IP through the manufacturing process. The soft IP Tagging structure and requirements are documented. A methodology to propagate the tags through the design process, which can include editing, synthesis, timing, placement, wiring, and other steps leading to GDS II generation, is also described. Semiconductor foundries, providers of IP, and IP integrators can use the methods described in this specification to track IP identification information.

1.2.Purpose

This specification provides a way to tag soft silicon intellectual property (IP), that is, intellectual property that has been delivered to a customer in RTL, gate level netlist, or script form, such that the IP can be tracked in a data driven manner for royalty calculations or other contractual IP usage obligations, as well as "where used" instantiated version linkage for bugs and errata, among other tracking applications. Tags are stored in the GDS II file as a series of formatted text lines. Previous IP Tagging implementations have been specific to cells and hard IP. Text tags instantiated in soft IP have not been recognized or carried forward by traditional EDA tools and are therefore not currently available in the GDSII database to verify actual implementation and usage. The tracking mechanism herein is not secure. It is susceptible to tampering and is intended only to facilitate the passing, use and sharing of information among honest IP users and IP providers; nevertheless, its existence does afford a very low-level form of security.

2. Normative References

The following referenced documents are indispensable for the application of this specification (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.

GDSII- Stream format, which is owned by Cadence Design Systems, Inc.

VSI Alliance, Virtual Component Identification Physical Tagging Standard (IPP 1 2.0DWG), June 2004[1]

VSI Alliance, Virtual Component Identification Soft IP Tagging Standard (IPP 4 2.0), September 2006

3. Definitions, Acronyms, and Abbreviations

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary: Glossary of Terms & Definitions*[2] should be referenced for terms not defined in this chapter.

3.1. Definitions

Foundry: Semiconductor manufacturing facility that creates integrated circuits for other Companies

Metric: A parameter used to judge the value of an IP. This normally refers to area (in square microns), but may be any other parameter related to cost or value as selected by a vendor and/or foundry.

Soft IP: Intellectual property for electronic design that is delivered without semiconductor device layout information.

Tag: Text data embedded in a file that provides additional information required for business reporting (rather than manufacturing).

Tagging: The process of annotating a GDSII-Stream file with a Tag.

3.2. Acronyms and Abbreviations

CAD: Computer Aided Design

EDA: Electronic Design Automation

IP: Silicon Intellectual Property

4. IP Tagging

4.1. Background

The VSIA Physical Tagging Standard states that information must be encoded into the physical description of an IP so the semiconductor foundry can produce a report of the parameters indicating the ownership of the IP in the design. The GDS II-Stream format is used for this purpose. This format's encoding mechanism allows for an arbitrary number of fields. The encoding must have at least the fields indicated in Table 1. These fields provide anyone reading the tags with the basis to produce a report containing the minimum required information prescribed by this standard, that is, a report of Vendor, Product, (product) Version, and Metric.

---

[1] VSI Alliance publications are available at http://vsi.org

[2] *The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

The VSIA Physical Tagging Standard supports a text string that serves as a tag and resides within a specified level of the GDS II file. The format of the text string is a reserved ampersand (&) character followed by a space. The expected sequence following the space is keyword, space, value. An example of this format is:

& Vendor company_name
& Product product_name
& Version string1
& Metric string2

The keywords supported by the current version of the VSIA Physical Tagging Standard are indicated in Table 1. The ampersand (&) character identifies the start of a keyword in the Hard IP tag.

Table 1Currently Supported GDS II Hard IP Tagging Keywords

Keyword	Argument(s)	Example
Vendor	String	& Vendor CompanyA
Product	String	& Product NewIP
Version	String	& Version 1.3a
Metric	String	& Metric 47.3
IP_Owner	String	& IP_Owner GroupB
Techno	String	& Techno CMOS090GP
Area	Floating	& Area 10.514
Celltype	LIB or IP obligatory LEAF optional Empty also allowed	& Celltype IP
Cell_Id	String	& Cell_Id nd2x1
Signature	String	& Signature <tbd>
Tag_Spec	String	& Tag_Spec 3.0
Date_Time	String	& Date_Time 20111014

The GDS II-Stream specification limits the string length to a maximum of 512 characters. This limitation includes the keyword, spaces, and the special ampersand (&) character. The VSIA Physical Tagging Standard specifies only the information about the physical design step to be contained within the GDS II file. This Soft IP Tagging Specification is intended to be seamlessly integrated with the existing Physical Tagging Standard.

4.2.Soft IP Tagging Overview

This new Soft IP Tagging specification takes the VSIA Physical Tagging Standard keywords, delimiters, and structure as an example. It then applies them to soft IP. The Soft IP Tags described in this specification all have % as the delimiter. Hard Tags as described in the referenced specification, all use & as the delimiter. The two different characters for tags will be in the same files at the GDSII level. Tag Readers that will support the capability of the soft tags must be configured to read the text structures with % in them in addition to &.

This specification details the implementation details of this new Soft IP tag. The details are described in the following three sections: “System Level Requirements,” “Tag Requirements,” and “IP Requirements.” Refer to Figure 1 for a design flow example. The tagging module is instantiated in the top level of the IP RTL. Dummy files required by the various EDA tools in the design flow are created to ensure the tag information is propagate to and included in the final GDSII.

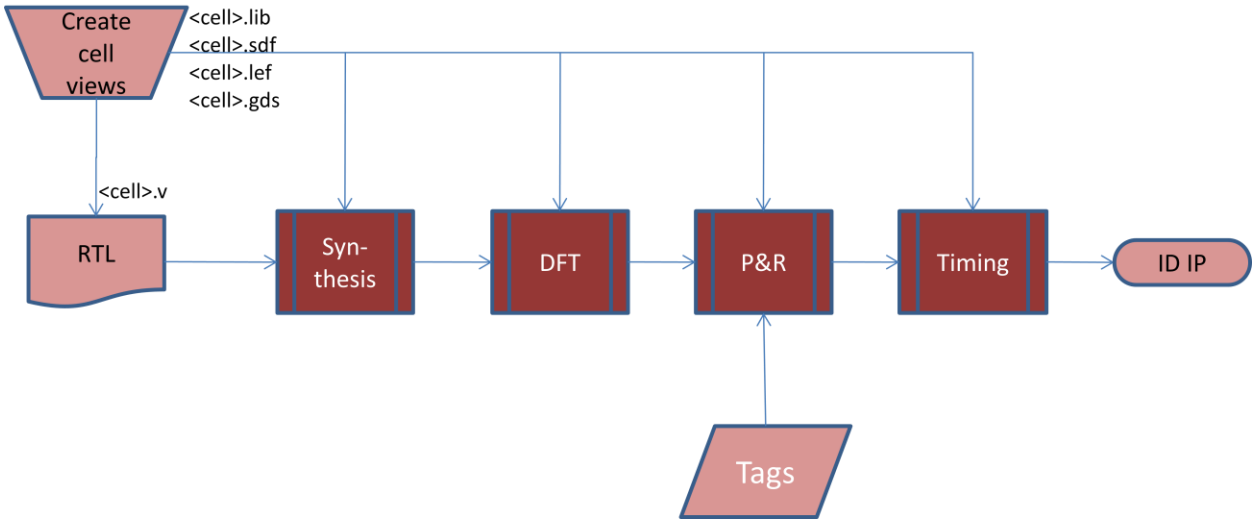

Figure 1: Design Flow incorporating Soft IP Tags

4.3.System Level Requirements

These requirements encompass the general structure of the Soft IP Tagging Specification.

Req 1.1. The Format and Structure of Soft IP Tags Is Similar to That of Hard IP Tags.

A Soft IP tag adds a set of tag information with content similar to that identified in the Hard IP tagging standard for GDS II. The format and structure is a reserved character followed by a keyword and keyword information, as shown in the following physical tagging example:

& Vendor CompanyX & Product FunctionX & Version 1.4 & Metric 111703

Per the Hard IP tagging standard, the recommended magnification should be 0 and the recommended xy coordinate should be 0, 0, however the user can choose any value for these parameters.

Req 1.2. The Soft IP Tags specific to technology should only be inserted when mapping the file to a technology.

The techno and area tags are not applicable to HDL or RTL level designs. These tags must not be inserted until the design is physically mapped.

Req 1.3. Every Soft IP Tag Must Be Delimited as a Separate Entity in the Output File

Each Soft IP tag must be considered a complete, individual entity. Soft IP tags cannot be combined and passed on as a new tag. The reader of the tags in the final GDS II output file is not able to determine the correctness or intent of combined tags.

Req 1.4. Specified Keywords Cannot Be Duplicated Within a Tag.

The following keywords cannot be duplicated within a single tag:

- Vendor
- Company
- Version
- Metric
- IP_Owner
- Techno
- Area
- Celltype
- Cell_Id
- Signature
- Tag_Spec
- Date_Time
- Process_Step
- ECCN

The reader of the tags in the final GDS II output file is not able to determine the correctness or intent of combined tags. These keywords are described in Section 2.4, “Tag Requirements.”

Req 1.5. Soft IP Tags Must Be Written to the Identified Layer in the GDS II Output File as Specified in the VSIA Physical Tagging Standard.

Tagging text lines may be added to any layer of the GDS II. The text information in this layer is the GDS II information, as required by that standard and additional Soft IP tag lines that are defined in this specification. Tag scanning programs should trigger on the appearance of & or % as the first character of the text field.

Req 1.6. Soft IP Tags Must Be Placed in the Top Most Hierarchical Level That Uses the IP.

The Soft IP tags should be written into the output file at the topmost hierarchical level that uses the IP or script. If multiple files are generated from the design step, the Soft IP tags should be placed only in the files that contain the topmost hierarchical level that uses the IP or script. Users must follow this rule when generating a Soft IP tag for a source RTL file. For example, a Soft IP tag placed in a source Verilog file must be in the top level Verilog module. It must be inside the module, not just in the file. This requirement removes any ambiguity as to the placement or duplication of the Soft IP tags.

4.4.Tag Requirements

The following requirements describe the actual format of the Soft IP tag, the different elements of the tag, and the constraints upon those elements.

Req 1.7. Soft IP Tag Format

The Soft IP tag has the following format:

{file format delimiter} {soft IP Tag delimiter} % {keyword} {data} % {keyword} {data} % …

This format follows the same general structure as the Hard IP tagging standard with the additional elements to allow identification of the Soft IP tag.

Req 1.8. The First Characters are File Format Delimiter Followed By a Space Character.

The first character of a Soft IP tag within a tool output must be a file format delimiter. This delimiter identifies the Soft IP tag as a repository of information about the design, but not a design element itself. Each tool specific format delimiter might be different. For example, comments in various programming languages might be delimited by any of the following characters: /*…..*/, #, or //. The delimiter must be followed by a space character to delineate the end of the delimiter and the beginning of the rest of the tag.

In order to make this tagging specification implementable in an expedient fashion, the tool-specific format delimiter should be an existing character set, if possible. For example, the tagging with a Verilog file uses the // comment characters as the Verilog file format delimiter. By using the comment delimiter, the Soft IP tag is seen as a non-design element. Use of these characters does not require any changes to the Verilog language to implement the Soft IP tag.

If all tools within the design development path support comments in their output formats, the delimiters should be the comment character or character sets.

Req 1.9. Verilog File Format Delimiter

The file format delimiter for Verilog is two front slash characters (//), sequentially with no space between them, as with the Verilog comment delimiters. The delimiter characters should be the first characters on the Soft IP tag line. These characters appear as:

//

Req 1.10. VHDL File Format Delimiter

The file format delimiter for VHDL is two dash characters (--), sequentially with no space between them, as with the VHDL comment delimiters. The delimiter characters should be the first characters on the Soft IP tag line. These characters appear as:

--

Req 1.11. Tcl File Format Delimiter

The file format delimiter for Tcl and UNIX/Linux shell scripts (csh, tcsh, bash, and so on) is a pound sign character (#), as with the Tcl and UNIX/Linux shell script (csh, tcsh, bash, and so on) comment delimiters. It should be the first character on the Soft IP tag line. This character appears as:

#

Req 1.12. The Soft IP Identifier is the First Element of the Tag.

The Soft IP tag must be identified as a special element of a design or be considered as a comment. The tool must be able to identify this set of characters to pass onto the next design tool. For this specification, the Soft IP tag identifier is the following string:

Soft_IP_Tag

An example of this is a Soft IP tag implemented in Verilog. The beginning of the tag appears as follows:

// Soft_IP_Tag %

The percent character (%) is the delimiter between elements of a tag.

Req 1.13. Space and Percent Characters Must Be Placed Between Each Element.

A space character followed by a percent character and another space character (%) must be placed between each element of a tag. An element is either the Soft IP tag identifier or a keyword and its data. For example, the percent characters and spaces in the following example are used correctly:

// Soft_IP_Tag % Vendor *ProviderC* % Product *XXY* % Version *1.1* % Celltype *IP* % Tag_Spec *1.0* % Date_Time *20120115_164211*

The items shown in italic are the data for each of the keywords.

Req 1.14. The First Keywords are Required Elements of Every Tag.

The following keywords are required for every tag for soft (RTL) IP. This aligns with the Hard IP tagging standard structure.

Keyword	Argument(s)	Definition	Example
Vendor	String	Complete legal name of IP Vendor	% Vendor CompanyD
Product	String	Complete name of product	% Product New_IP
Version	String	Version of product	% Version 1.3a
Celltype	LIB or IP obligatory LEAF optional Empty also allowed	LIB = Library cell or memory generator IP = reusable high level block or memory LEAF = leafcell. Smallest entity,	% Celltype IP

		could not be found alone, only in an IP or a LIB Note : for memory generator, both IP or LIB are authorized	
Tag_Spec	String	Corresponds to the Revision of Soft IP Tagging specification	% Tag_Spec 1.0
Date_Time	String	Tagging date YYYYMMDD[_HHMMSS]	% Date_Time 20111014_231542

Req 1.15. Optional Keyword

The Export Control Classification Number (ECCN) is an alpha-numeric code that describes the item and indicates licensing requirements to support the Wassenaar arrangement control list for dual use goods and technologies. Since ECCNs reflect national policies, a product or technology might be linked to the same or to different ECCNs in different countries. The following optional keyword is defined to automate the extraction of the ECCN:

ECCN *country_code_ECCN_value*

Keyword	Argument(s)	Definition	Examples
ECCN	String	*country_code* is a two-letter code, that specifies the code for a country, as defined in ISO 3166 standard, e.g. FR for France, CN for China, etc, or a macro-region, e.g. EU for European Union. *ECCN_value* is the ECCN code applied by the related country. It is an alphanumeric string.	*%* ECCN *US_5D002.C.2* % ECCN *FR_5D002.C.2* % ECCN *EU_5D002.C.2*

Req 1.16. Required Keywords for physically mapped IP

The following keywords are required for every tag for physically mapped (GDSII) IP. This supports the Hard IP tagging standard structure. These tags are specific to technology and must be included during the mapping to a specific technology. They are not required for structural, non-technology mapped IP.

Keyword	Argument(s)	Definition	Example
Techno	String	Defines the process technology used. Vendor technology name in upper case.	% Techno CMOS090GP
Area	Floating	Product or cell area in mm^2	% Area 10.514

Req 1.17. Character Constraints for Data Information

The data information after a keyword can contain only the following characters:

- Alphanumeric characters: a-z A-Z 0-9
- Space character:
- Punctuation characters: . , ? ! " ' ` : ;
- Special characters: ( ) { } [ ] \ / | _ - = + @ # $ ^ * < > ~

The ampersand (&) and percent (%) characters are forbidden in the data information after a keyword. These characters are not allowed because they mark the beginning of the next keyword for the Hard IP and Soft IP tags, respectively.

The total length of reserved character, keyword, and keyword data cannot exceed 512 characters.

Req 1.18. User-defined Keywords

User-defined keywords are allowed within the Soft IP tag. However, the keywords must be delineated by the percent (%) character, and the actual keyword must start with an underscore (_) character. The only characters allowed within the keyword are alphanumeric and underscore characters. Spaces are not allowed.

Allowed characters: a-z A-Z 0-9 _

For example:

> // Soft_IP Tag % Vendor Green,_Inc. % Product CoreZ % Version 1.4 % Celltype IP % Tag_Spec 1.0 % Date_Time 20100328_231542 % _USER_KEYWORD user1

Req 1.19. All Soft IP Tags Must End with Characters that Delimit the Tag as a Separate Line or Entry in the Output File.

The Soft IP tag must end with whatever characters are required to separate the Soft IP tag as an individual element of the output file. In many cases, this requirement reduces to the use of a carriage return or line feed and carriage return. The ending characters, like the file format delimiter described previously, are dependent on the format of the output file.

4.5.IP Requirements

The following requirements define the physical implementation of the soft IP tags and the propagation through to the final database.

Req 1.20. A Dummy Cell Must be Instantiated in the Top Level of the Soft IP

The dummy tagging cell will be instantiated in the top-most level of the soft IP. The cell instantiation will be preceded by comments that identify the tagging cell. The associated tag should

be included in the comment field. A Verilog comment example for the version 1.4 of CoreZ from the vendor Green could be:

```
// DO NOT REMOVE THE FOLLOWING INSTANCE!
// This is an instance of a technology-independent id cell which is used
// to track usage of this soft IP. You can find design views for this
// cell (Liberty, LEF, GDS etc.) in the same block directory as this RTL
// code. You must include this cell throughout your implementation flow
// to avoid the risk of incurring legal consequences.
// Soft_IP Tag % Vendor Green,_Inc. % Product CoreZ % Version 1.4 % Celltype IP % Tag_Spec
1.0 % Date_Time 20100328_231542
LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4 idcell_instance();
```

A VHDL comment example for the same IP could be:

```
-- DO NOT REMOVE THE FOLLOWING INSTANCE!
-- This is an instance of a technology-independent id cell which is used
-- to track usage of this soft IP. You can find design views for this
-- cell (Liberty, LEF, GDS etc.) in the same block directory as this RTL
-- code. You must include this cell throughout your implementation flow
-- to avoid the risk of incurring legal consequences.
-- Soft_IP Tag % Vendor Green,_Inc. % Product CoreZ % Version 1.4 % Celltype IP % Tag_Spec 1.0
% Date_Time 20100328_231542% Metric 02
entity LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4 is
end LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4
```

Note that the example instance name is fairly verbose. The phrase "LEGALLY_REQUIRED_DO_NOT_REMOVE" is intended to clearly communicate that the instance belongs in the IP even though it does not include any functionality.

Req 1.21. All Tool-Specific View Files Must be Supplied with the Soft IP

The .v, .lib, .lef, and .gds views must be provided that align with the dummy tagging cell in the RTL. The files are necessary to ensure that the tools correctly recognize and propagate the soft IP tagging information.

Req 1.22. The module Only Contains the Tag Comment Text

The module file is required to provide the linkage between the soft IP used in the SoC and the final GDSII. The dummy tagging module must only consist of the module/endmodule statements and a comment containing the tag string. For example:

```
module LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4 ();
// Soft_IP Tag % Vendor Green,_Inc. % Product CoreZ % Version 1.4 % Celltype IP % Tag_Spec
1.0 % Date_Time 20100328_231542
```

```
endmodule
```

or for VHDL:

```
architecture V1 of LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4 is
 begin
 -- Soft_IP Tag % Vendor Green,_Inc. % Product CoreZ % Version 1.4 % Celltype IP % Tag_Spec
1.0 % Date_Time 20100328_231542
 end
end V1
```

Req 1.23. The .lib File Must Include Pragmas

The .lib file view must only contain the don't_touch and don't_use pragmas to ensure that the dummy tagging cell is not optimized out or otherwise removed from the SoC by the design tools. For example:

```
library (LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4) {
 cell (LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4) {
  dont_use : true;
  dont_touch : true;
 }
}
```

Req 1.24. The .lef File Must Not Impact the SoC's Physical Implementation Size

The .lef file view ensures that a physical component is instantiated in the SoC that will carry the text tagging information for the soft IP. A 0 size cover macro should be used. As the soft IP may be flattened with other blocks in the SoC, an arbitrary origin close to 0,0 should be defined. For example:

```
MACRO LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4
 CLASS COVER ;
 ORIGIN 0.9 0.9 ;
 SIZE 0.00 BY 0.00 ;
END LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4
```

Req 1.25. The .gds File Must Include the Soft IP Tags

```
LIBRARY:        LEGALLY_REQUIRED_DO_NOT_REMOVE_Green_CoreZ_1_4;
STR:            VSIATAG
TEXT:   "% Vendor Green, Inc.", layer: 63, type: 63, font: 0, adjust: (2,0), path: 0, width: 0,
        Strans: At: (0.000,0.255), Not_reflected, mag = 0.047, angle = 0
        {};
TEXT:   "% Product CoreZ", layer: 63, type: 63, font: 0, adjust: (2,0), path: 0, width: 0,
```

Strans: At: (0.000,0.255), Not_reflected, mag = 0.047, angle = 0
{};
TEXT: "% Version 1.4“, layer: 63, type: 63, font: 0, adjust: (2,0), path: 0, width: 0,
Strans: At: (0.000,0.255), Not_reflected, mag = 0.047, angle = 0
{};
TEXT: “% Celltype IP”, layer: 63, type: 63, font: 0, adjust: (2,0), path: 0, width: 0,
Strans: At: (0.000,0.255), Not_reflected, mag = 0.047, angle = 0
{};
TEXT: “ % Tag_Spec 1.0”, layer: 63, type: 63, font: 0, adjust: (2,0), path: 0, width: 0,
Strans: At: (0.000,0.255), Not_reflected, mag = 0.047, angle = 0
{};
TEXT: “% Date_Time 20100328_231542”, layer: 63, type: 63, font: 0, adjust: (2,0), path: 0,
width: 0,Strans: At: (0.000,0.255), Not_reflected, mag = 0.047, angle = 0
{};
STREND;
LIBEND;

Req 1.26. Reporting Mechanism

A GDSII-Stream file may be populated with tracking information from a variety of IP vendors. Upon scanning that file it must be possible to readily generate a report of the fields specified in Table 1. It must be possible for the report to show this information for each and every IP on the chip.

The reporter should fail if a file does not meet the specification. Examples of failures are unrecognized keywords. At the option of the party producing the reports of tagged GDSII-Stream databases, a more detailed report (such as, listing each product instance and its version) may be generated.

A complete accounting of all IPs occurring in a GDSII-Stream file (a single chip design) is required. This accounting must be accurate and allow for any hierarchical use of IP blocks that could occur if IPs are nested, one within another, rather than just placed side-by-side. Side-by-side IPs are fully accounted for, just as are multiply nested virtual components in both the example Tag Reading Program provided (Section 3.3), and the associated working copy available from the VSIA website (See Section 3). Any user of this specification should ensure similar accounting is maintained.

The report generator must analyze the hierarchy of the GDSII-Stream file to determine the count of the instances on a chip. For example, structure A may have 2 references to structure B and structure B may have 2 references to structure C. Assuming structure C is tagged with an IP vendor's product then, in the computation, the count for C is 4 and not 1. The following example shows an alternate GDS II structure with a Hard IP Tag and two Soft IP tags; note the delimiter difference between the first tag for the hard IP and the following two soft IP tags:

text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 2,1 "& Vendor Company1"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 3,1 "& Product UART"

text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 4,1 "& Version 02.01"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 5,1 "& Metric 0"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 6,6 "% Vendor Company2"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 6,6 "% Product IPX"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 6,6 "% Version 01.01"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 6,6 "% “% Celltype IP"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 6,6 "% Tag_Spec 1.0"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 6,6 “% Date_Time 20111014"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 6,6 “% ECCN US_5D002.C.2"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 7,7 "% Vendor Company3"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 7,7 "% Product ZZZ"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 7,7 "% Version 03.03"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 7,7 "% “% Celltype IP"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 7,7 "% Tag_Spec 1.0"
text 63 TT 63 PR 0,2,0 MAG 0.001 R 90 XY 7,7 “% Date_Time 20100328_231542"