

次期LPBフォーマット拡張検討

JEITA-LPB相互設計SC 集中討議

2016.9.2

フォーマット改訂への道のり

■ IEEE/IECのUpdate・・・新構造・熱解析等との連携のための書式拡張

- ・ 全面改訂だと、新標準化と同じ手順 通常の標準は5年ぐらい間隔

2020年以降に次バージョン制定とし、前回ベースだと

2019年度IEEEのWGキックオフ

2018年度のWGでPAR提出、ドラフト提出

2016-17年度で仕様をFix

今年度は、改訂項目を集中討議・LPBフォーラムから抽出する

今回の集中討議が改訂のスタートです!!

本日の検討内容事

次期LPBフォーマット拡張検討

1. パワーデバイス対応

- 大電流・大電圧、3D形状、熱解析など

2. コンストレイント追加

- インピーダンスマッチング、クロストーク、接続順定義など

3. ユーザ定義の拡張

- 拡張方法、管理・公開・採用のルールなど

パワーデバイス対応

■ パワーデバイス対応（大電流・大電圧、3D形状、熱解析）

大電流・大電圧：パワー系のネットは特別に扱う必要がある

- A) 通常の信号・電源とは別物
- B) 特別なDRCがある（3D形状を考慮）
- C) 同一端子に複数のボンディングワイヤがある

熱解析：電気特性解析に、熱を考慮する必要がある

- D) 温度端子をどう扱うか
- E) 熱源、ヒートシンク、ファンなどの位置情報、境界条件
- F) 熱解析をするために必要な情報

パワーデバイス対応

■ パワーデバイス対応（大電流・大電圧）

制御系の電源と、パワー系の電源が別なため、端子属性も分けたい
また、温度端子(モデル上だけのもの)を追加したい

⇒ 現状、C-Formatのmoduleの端子属性は、
power、ground、signal、floating、through、dontcare

⇒ これにパワー系の電源、信号と温度端子を追加
p_power、p_ground、p_signal、thermal

上段：現状、下段：記述案

```
<module name="IGBT_PKG" type="PKG" shape_id="PKG_BODY" x="0" y="0" angle="0" thickness="560">
  <socket name="IGBT_PKG" >
    <default>
      <port_shape padstack_id="BGA BALL" />
    </default>
    <port id="A1" x="-12500" y="12500" angle="0" name="C1" type="power" />
    <port id="A2" x="-11500" y="12500" angle="0" name="E2" type="ground" />
    <port id="A3" x="-10500" y="12500" angle="0" name="C2E1" direction="output" type="signal" />
    <port id="A4" x="-9500" y="12500" angle="0" name="G1" direction="input" type="signal" />
    <port id="TP" x="-9500" y="12500" angle="0" name="TP" type="dontcare" />
```

```
<module name="IGBT_PKG" type="PKG" shape_id="PKG_BODY" x="0" y="0" angle="0" thickness="560">
  <socket name="IGBT_PKG" >
    <default>
      <port_shape padstack_id="BGA BALL" />
    </default>
    <port id="A1" x="-12500" y="12500" angle="0" name="C1" type="p_power" />
    <port id="A2" x="-11500" y="12500" angle="0" name="E2" type="p_ground" />
    <port id="A3" x="-10500" y="12500" angle="0" name="C2E1" direction="output" type="p_signal" />
    <port id="A4" x="-9500" y="12500" angle="0" name="G1" direction="input" type="signal" />
    <port id="TP" x="-9500" y="12500" angle="0" name="TP" type="thermal" />
```

温度端子は、外部に出ないことがあるので、socket定義にはそぐわないかも

パワーデバイス対応

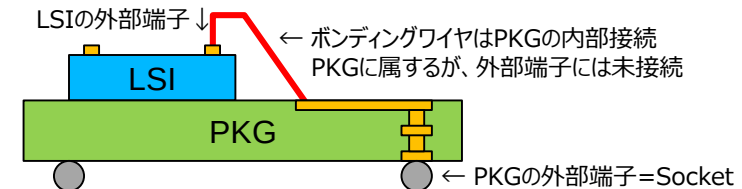
■ パワーデバイス対応（大電流・大電圧？）

大電流のためボンディングワイヤを同一端子間で複数使用、解析設定を自動化したい

⇒ 現状、C-Formatに、R-Formatのボンディングワイヤ定義を関連付けられない
R-FormatとG-Formatを読んだ電磁界解析ツールで指定する

⇒ ボンディングワイヤはC-Formatのmoduleの内部接続のため、socketに関連なし
例：PKGのボンディングワイヤは、PKGが内包するLSIと、PKG内部の導体を接続するものなのでPKG側で定義すべきだが、外部端子のsocketには未接続

⇒ moduleのsocketで定義ではなく、componentのplacementで定義が良いか
ボンディングワイヤのID、形状(R-Formatで定義)、
始点xy、始点の部品(nullだと自分自身)、
終点xy、終点の部品(")、
配置面



上段：現状、下段：記述案

```
<component>  
  <placement ref_module="IGBT_DIE" inst="IGBT_DIE" x="0.0" y="0.0" angle="0" mount="TOP" />  
</component>
```

```
<component>  
  <placement ref_module="IGBT_DIE" inst="IGBT_DIE" x="0.0" y="0.0" angle="0" mount="TOP" />  
  <placement bondingwire="WB1" bw_name="WIREBOND1" x1="1.0" y1="1.0" inst1="IGBT_DIE" x2="2.0" y2="1.0" />  
  <placement bondingwire="WB2" bw_name="WIREBOND1" x1="1.0" y1="2.0" inst1="IGBT_DIE" x2="2.0" y2="2.0" />  
</component>
```

Component自体を
moduleに入れる方
が良いかも
LSI側にPKG側が定
義する方法もあり

パワーデバイス対応

■ パワーデバイス対応（大電流・大電圧？）

パワーデバイス特有のDRCをコンストレイントとして入れる

⇒ 現状定義できない 製造上のルールではないのでC-Formatか

項目	問題
部品間の空間距離 制約	3D形状をどう扱うか未定
配線—部品間の 空間距離(Z方向)の制約	3D形状をどう扱うか未定
配線間の沿面距離の制約 (スリットを設ける)	C-Formatに誘電体に対する抜きパターンを定義できない
配線容量(幅,厚み等)の 制約	C-Formatはコンポーネントの端子のみ扱いのみで中間ネットをサポートしていない
Via容量(穴径、数等)の制約	C-Formatはコンポーネントの端子のみ扱いのみで中間ネットをサポートしていない
放熱パターンの定義	3D形状をどう扱うか未定
ジャンパー線の定義	形状はR-Formatで定義可能だが、配置を定義できない

パワーデバイス対応

■ パワーデバイス対応（3D、熱）

3D形状は、従来の考え方のキャラメルボックスで良いのでは
熱解析に必要な情報がない

⇒ 熱源、ヒートシンク、ファンなどの位置情報 現状定義できる場所がない
強制対流などの境界条件を定義する

【具体的なパラメーター候補】

1. 熱源（発熱量）
2. 熱抵抗
3. 熱伝達率（残銅率）
4. 熱伝導率
5. 筐体寸法
6. 表面温度
7. （半導体の）Junction温度

熱解析に必要な情報として
R-Formatの物性値に熱関係のパラメータを入れる
熱解析のために、C-Format
に熱モデルを入れる、そのI/F
をどうするか

⇒ 熱解析をどうするかの方針を決める必要がある
A)形状だけ与えて熱解析ツールで解析する
B)熱抵抗、熱容量などの熱回路網で表現

コンストレイント拡充

- コンストレイント拡充（インピーダンスマッチング、クロストーク、接続順定義）
従来のC-Formatは、外部に対する制約を記述しているもの
→ 自身の制約も記述したい

インピーダンスマッチングのためのL/S制約

- A) moduleの外に、配線層ごとにインピーダンスを満足するL/Sのテーブルを追加
- B) C-Formatにネットの概念を持ち込むか
- C) 構想設計に必要なものに留めるべき

クロストーク考慮のスペース

- D) ネットがないと難しい

部品の接続順を定義

- E) トポロジを制約として扱う必要はあるか
- F) トポロジを制約として扱うには記述は複雑になる

予定の配線層情報をC-Formatに持たせる案

現状 Cフォーマットの<impedance>が、
Rフォーマットのクロスセクション定義の<layer_def>要素と紐づいていない

```
<LPB_CFORMAT> <module> <socket>
<constraint>
  <impedance
    port_name="端子名" port_id="端子番号"
    group_name="端子グループ名" type="[single|differential|common]"
    min="最小値 (-公差)" typ="狙い値 (公称値)" max="最大値 (+公差)"
  />
```

自分(module)の端子の
狙い値を外に示しているのみ

```
<R-Format>
<Physicaldesign name="デザインルール名">
  <layer_def> ...スタック層数分
    <layer 層名、素材名、物性、厚み、L & Sルール 等の属性 />
  </layer_def>
/>
```

狙い値を満たすような層を
用意しているはず

TOP SR
L1
BU_1_2
L2
CORE
L3
BU_3_4
L4
BOTTOM SR

C-Formatに
予定の配線層を示したらどうか

予定の配線層情報をC-Formatに持たせる案

案

Cフォーマットの<impedance>と、
Rフォーマットのクロスセクション定義の<layer_def>要素と、層名で紐づける

```
<LPB_CFORMAT> <module> <socket>  
<constraint>  
  <impedance  
    port_name="端子名" port_id="端子番号"  
    wiring="層名"  
    group_name="端子グループ名" type="[single|differential|common]"  
    min="最小値 (-公差)" typ="狙い値 (公称値)" max="最大値 (+公差)"  
  />
```

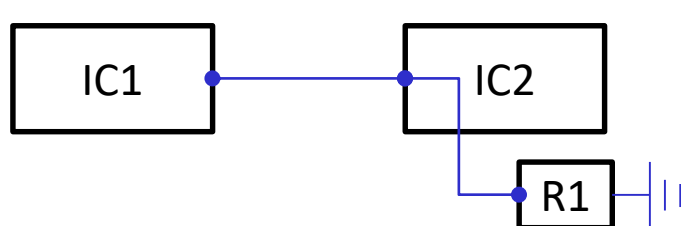
```
<R-Format>  
<Physicaldesign name="デザインルール名">  
  <layer_def> ...スタック層数分  
    <layer 層名 素材名、物性、厚み、L & Sルール 等の属性 />  
  </layer_def>  
</>
```

TOP SR
L1
BU_1_2
L2
CORE
L3
BU_3_4
L4
BOTTOM SR

接続順 その1

案 部品の接続順を指示する（終端抵抗なのか、ダンピング抵抗なのか、など。配線トポロジを明示）

<C-Format><component><placement>の拡張

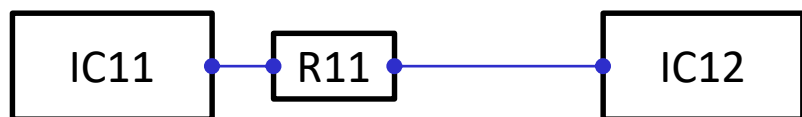


<component>

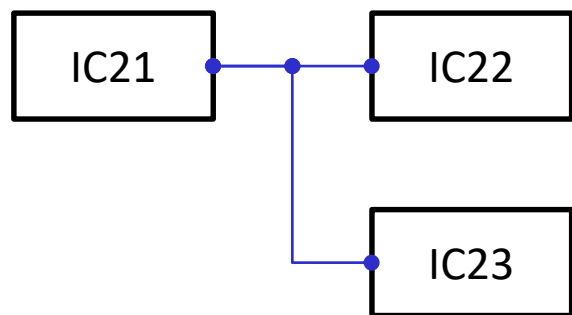
<placement inst="IC1" gid="1" seq="1" .../>
<placement inst="IC2" gid="1" seq="2" .../>
<placement inst="R1" gid="1" seq="3" .../>

グループID

接続順



<placement inst=" IC11" gid="2" seq="1" .../>
<placement inst=" R11" gid="2" seq="2" .../>
<placement inst=" IC12" gid="2" seq="3" .../>

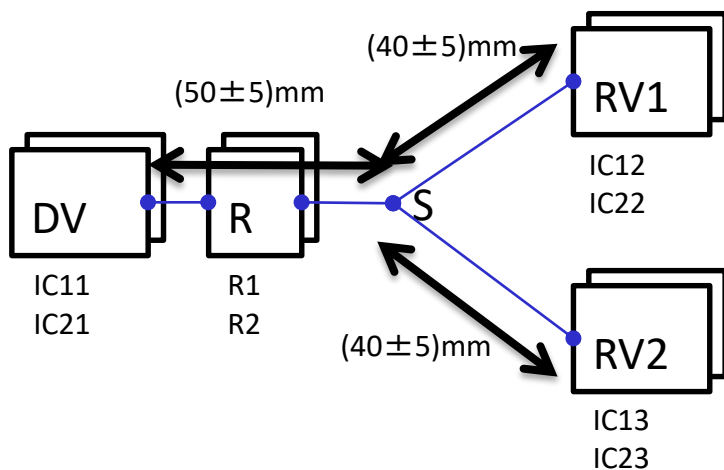


<placement inst=" IC21" gid="3" seq="1" .../>
<placement inst=" IC22" gid="3" seq="2" .../>
<placement inst=" IC23" gid="3" seq="2" .../>

接続順 その2

案

<C-Format> <component> <placement> のあとに <topology> を置いて、部品間（本当はピン間）配線制約を表現



<topology>

<pathgroup name="star1">

<path>

<inst name="IC11" seq="1" label="DV"/>

<inst name="R1" seq="2" label="R"/>

<fanout name="R1" id="1" seq="3" label="S"/>

<inst name="IC12" seq="4" label="RV1"/>

<inst name="IC13" seq="4" label="RV2"/>

</path>

<path>

<inst name="IC21" seq="1" label="DV"/>

<inst name="R2" seq="2" label="R"/>

<fanout name="R2" id="1" seq="3" label="S"/>

<inst name="IC22" seq="4" label="RV1"/>

<inst name="IC23" seq="4" label="RV2"/>

</path>

<rule p2p="DV:S" unit="mm" value="50" tolerance="5"/>

<rule p2p="RV1:S" unit="mm" value="40" tolerance="5"/>

<rule p2p="RV2:S" unit="mm" value="40" tolerance="5"/>

</group>

</topology>

instance名

トポロジ内の仮名

R1からの第1分岐

R2からの第1分岐

ユーザ定義の拡張

■ ユーザ定義の拡張 （拡張方法、管理・公開・採用のルール化）

C/R/M-Format

- A) ユーザ拡張に識別名を追加する
- B) ext_“識別名”で始まるエレメント、アトリビュートを許す
- C) 識別子は、LPBフォーラムが発行、管理
- D) フォーマット採用は、LPBフォーラムが判断、JEITAで申請

G-Format

- E) コメントに記述？

ユーザ拡張領域とは

ユーザ拡張領域とは

LPB-Formatにユーザ独自の仕様を追加する仕組み

将来的な仕様追加のプレビュー/テストの為の仕組み

ユーザ拡張領域の仕組み

C/R/M-Format

XML書式の範疇で、拡張領域を取り込む。

標準的なXMLパーサを使用していれば、特別な改良無しで互換性を保つことができる。

G-Format

G-Formatのコメントとして拡張領域を記述する。

特別な改良無しで互換性を保つことができる。

特定ユーザによる拡張対応案

R/C/Mフォーマットのパーザーは、先頭が“ext”で始まるエレメントとアトリビュートについて、エラーとして扱ってはならない。

先頭が“ext_xxx”で始まるエレメントやアトリビュートは、識別名xxxを持つユーザのみが定義できるものとする。

xxx社には、識別名の発行する際、拡張部分についての著作権を放棄してもらう。

拡張部分についての詳細仕様の公開は、義務ではないが、望ましい。

拡張部分についての詳細仕様が公開された場合、JEITAは当該拡張の内容を将来のバージョンで基本仕様に取り込むことを検討することが望ましい。

識別名 とは

ユーザによる仕様拡張の競合を防ぐ仕組み

複数のユーザが独自に仕様拡張を行うと、互いのキーワードが競合する恐れがある。

仕様を拡張するユーザが予めJEITAに申請し、個別の識別名の発行を受ける。

ユニークな識別名の下で拡張を行うことで、仕様の競合を防ぐ。

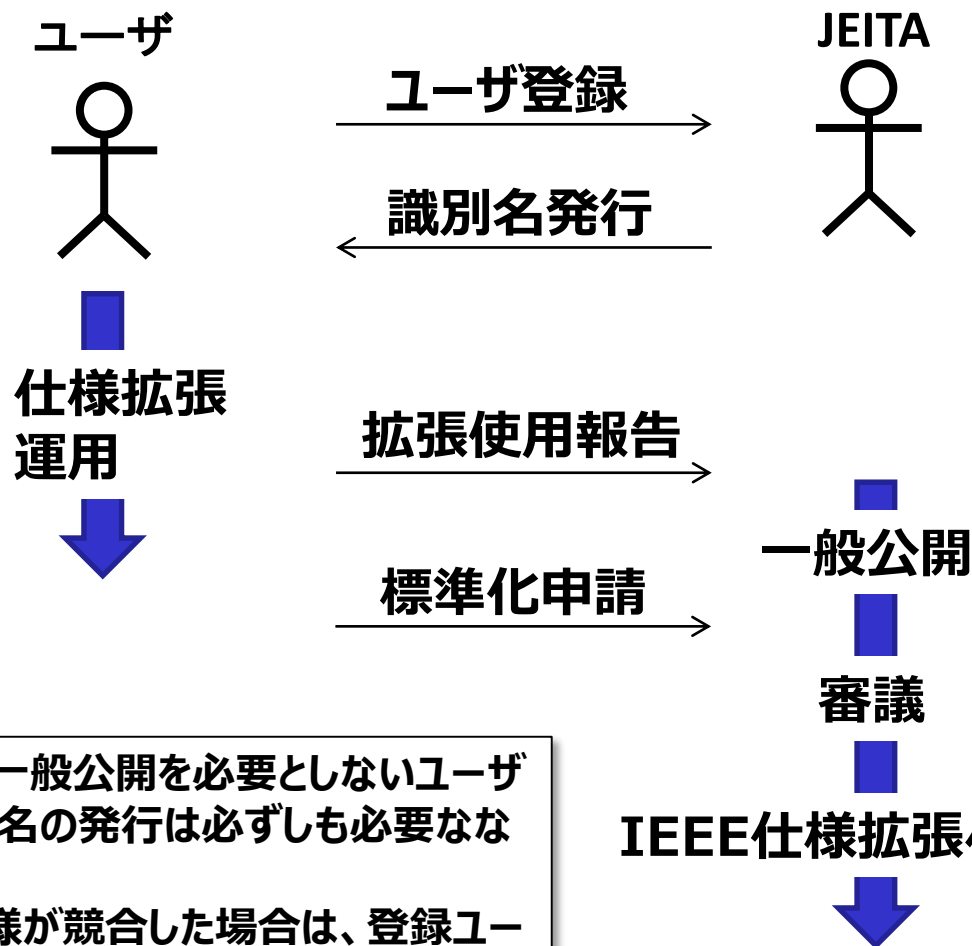
【例】

A社 ... 識別名 : A

B社 ... 識別名 : B

```
<ext_A>  
  ~A社による仕様拡張  
</ext_A>  
<ext_B>  
  ~B社による仕様拡張  
</ext_A>
```

ユーザ拡張領域の運用形態



社内運用など、一般公開を必要としないユーザの場合は、識別名の発行は必ずしも必要はない。
ただし、拡張仕様が競合した場合は、登録ユーザの仕様を優先する。
また標準化を申請することもできない。

審議の結果、IEEEへの取り込みを拒絶する場合もありうる。

C/R/M-Formatでの書式

ユーザ定義エレメント

ユーザ独自のエレメントを追加する

【記法】

<ext_識別名>

(※)識別名に関しては後術

～ この部分は、XMLの書式に従ってる限り、ユーザ独自で
定義した構文を記述してよい～

</ext_識別名>

ユーザ定義アトリビュート

既定義のエレメントに対しユーザ独自のアトリビュートを追加する

【記法】

ext_識別名=" ～この部分にユーザ独自のアトリビュートを追加する～”