LISAを利用した新機能による EXPERTのカスタマイズ

はじめに

LISA(Language for Interfacing Silvaco Applications) は、レイアウト・エディタである Expert から利用できる、シルバコ独自のスクリプト言語です。条件文やループ文などの主要なプログラム・フロー制御を全てサポートし、様々なデータ・タイプと演算子を使用できます。XI スクリプトは EXPERT に対するアプリケーション専用コマンドを持つ LISA の拡張機能です。EXPERT で使用するレイアウト・オブジェクトやほぼ全てのコマンドを完全にコントロールできます。

新規コマンドの概要

現在、以下の新しい機能が利用可能です。

●ガードリングの自動作成
●アレイ配置数の自動最大化
●バス・テキストの一括作成
●くし形抵抗への自動変換（PCELL を利用）

これらの新機能はレイアウト作業効率を向上するために作成されました。実行方法は二つあり、コマンド入力による方法と、GUI による入力方法があります。コマンド入力による実行の場合は、スイッチ・オプション”/h”によって、ヘルプ・ダイアログが表示されます。これらの新コマンドは、Expert 起動時にスクリプト自動実行の設定をする事で標準コマンドの様に利用できます。さらに Expert カスタム・メニューへ追加すれば、ショートカットキーを割り当てる事も可能となります。

ガードリングの自動作成

ガードリング自動作成は、選択しているオブジェクト (ボックス／ポリゴン／インスタンス／アレイ) 周囲を、アクティブ・レイヤを使ってワイヤ図形で作成します。オブジェクトを一つも選択していなければ、警告メッセージが表示され終了します。選択している複数のオブジェクトの間にリングが通らない場合は、同一リングで囲います。（図1参照）

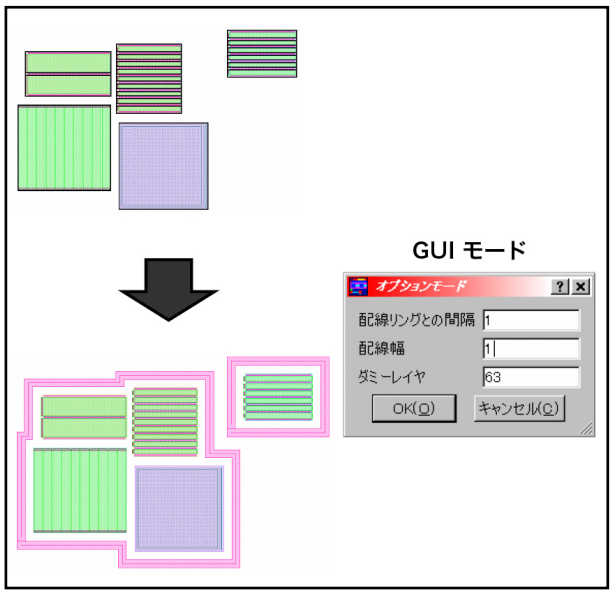

図１．自動生成されたガードリングの例

指定可能な変数値は、選択オブジェクトからの距離とリング幅、中間処理で使用するダミーレイヤ番号です。

実行コマンド構文：make_ring
XI > make_ring < 距離 > < 幅 > < レイヤ番号 >

距離か幅のどちらか一方でも省略された場合 (引数が 1 以下) は、GUI モードを表示します。
ダミーレイヤ番号が省略された場合、デフォルト値である”63”が使用されます。

アレイ配置数の自動最大化

アレイ配置数の自動最大化は、指定した領域内に配置するアレイ数を自動で最大化する事が可能です。領域の指定方法は二つあり、、任意の二点をマウスクリックにより指定する方法（デフォルト）と、選択しているオブジェクトの内側領域を指定する方法があります。選択しているオブジェクトの内側領域を指定する場合、オブジェクトを一つも選択していなければ、警告メッセージが表示され終了します。

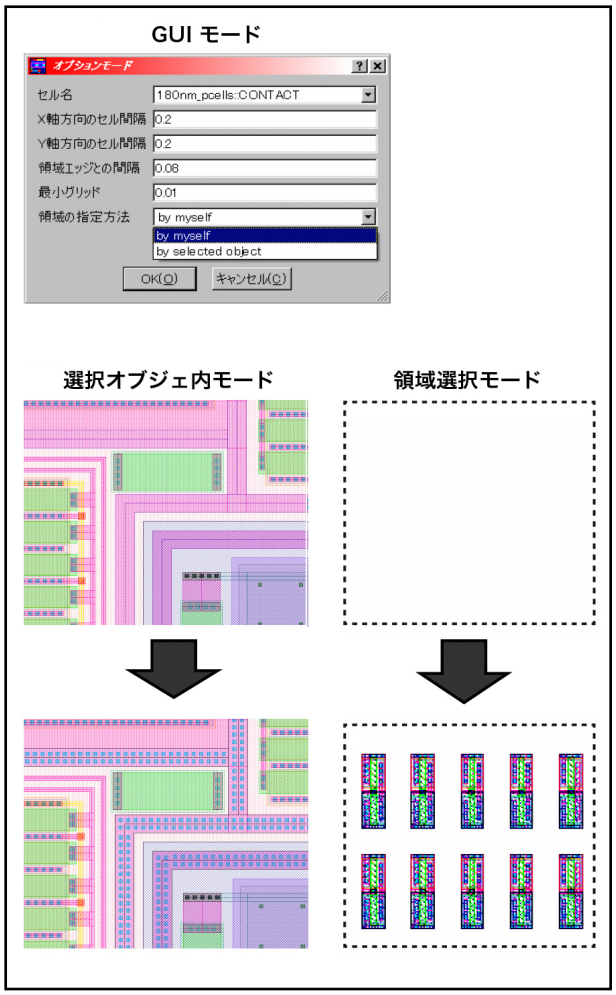

図２．配置アレイ数の最大化の例

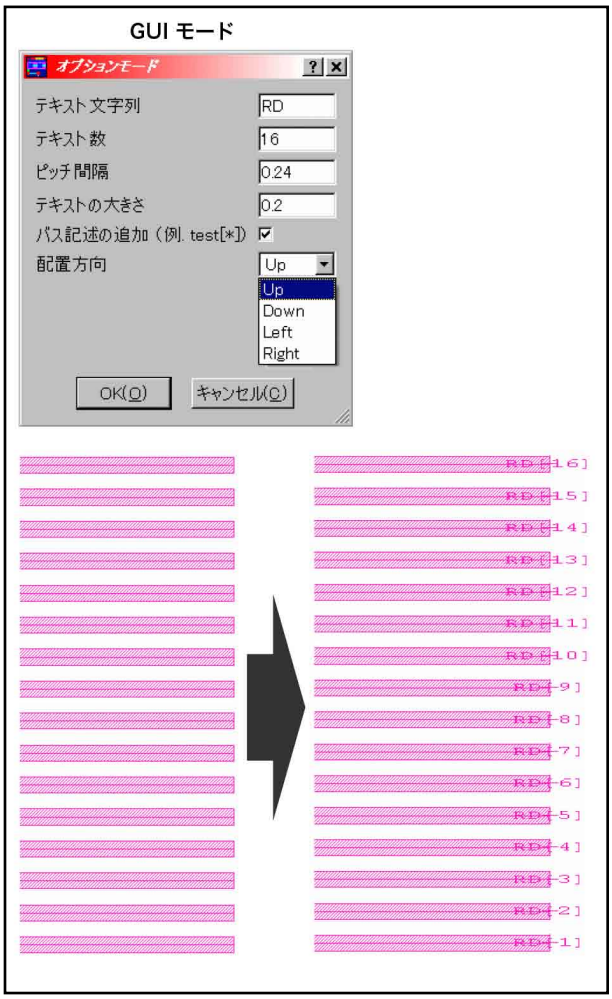

図３．バス記述によるテキスト作成の例

領域指定方法の他に、指定可能な変数値は、配置するセル名とアレイ間隔 (X 軸方向と Y 軸方向)、領域エッジとの間隔、最小グリッドです。

実行コマンド構文：max_array
XI > max_array < セル名 > <X 軸間隔 > <Y 軸間隔 >
< エッジ間隔 > < 最小グリッド >

引数はセル名から連続順にて割り当てられます。
スイッチ・オプションは”/d”がマウスによる領域指定 (デフォルト) で、”/s”が選択オブジェクト内を指定領域になります。

セル名を省略するか、不明なセル名を指定した場合は、GUI モードを表示します。

バス・テキストの一括作成

この機能はバス名称のテキストを作成するので、メモリなどのビット線に利用すると、最も効果があります。

バス記述は、“***[数]”の形式で作成され、バス記述にするかしないかは選択が可能です。デフォルトはバス記述となります。

バス記述の On/Off の他に、指定可能な変数値は、テキスト文字列と総数、ピッチ間隔、テキストの大きさ、及びテキストの作成方向です。

実行コマンド構文：text_array
XI > text_array < テキスト名 > < 総数 > < ピッチ >
< 大きさ >

引数はテキスト名から連続順にて割り当てられます。
スイッチ・オプションは、”/nb”でバス記述を OFF、“/b”または指定しなければ、バス記述を ON とします。

他のオプションとして、テキストの作成方向は必須入力で、“/u”が上方向、”/d”が下方向、”l”が左方向、”r”が右方向となります。テキストの作成方向をオプション指定しなかった場合は、GUI モードを表示します。

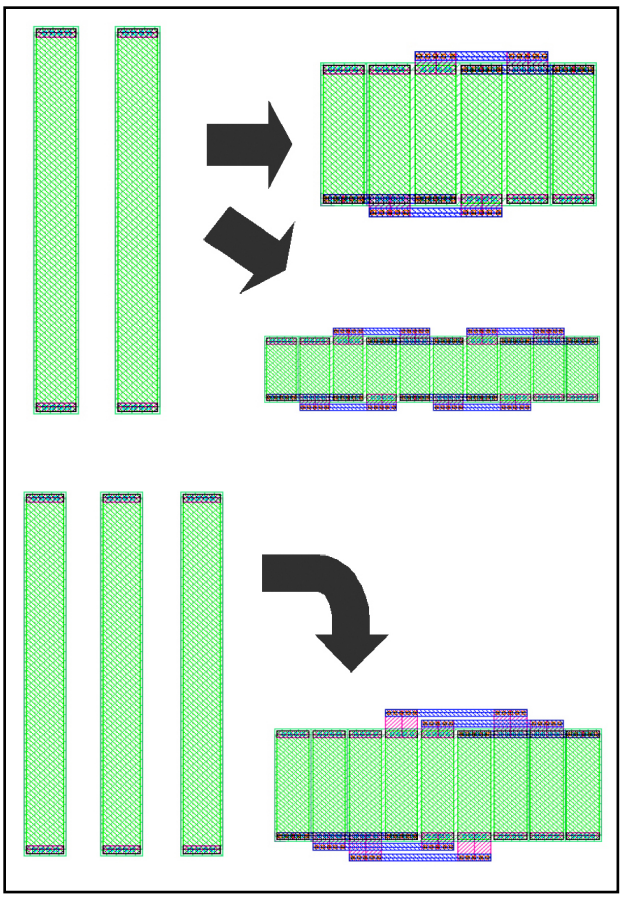

図４．くし型抵抗の自動変換の例

くし形抵抗へ自動変換（PCELL を利用）

アナログ集積回路は、回路の精度と性能を良く得るためには、デバイス・マッチングに依存します。
最低限のマッチングはあまり困難なく得る事が可能ですが、適度なマッチングを取るには、コモンセントロイド・レイアウトを生成するのに、相互嵌合を使用する事で確実に得る事が可能です。

この新機能では、明示されたパラメータを持った PCELL 抵抗を自動的に相互嵌合 (くし型) で生成します。この機能を利用するために、以下のパラメータのセッティングが、PCELL に要求されます。

・分割数
・くし形への変換の制御
・変更するセルの数
・くし型の位置による変形

実行手順は、対象となる２個以上の PCELL を選択して、コマンドを実行します。
ひとつの抵抗体からマッチした抵抗群を構成するので、同一の PCELL マスターのインスタンスしか選択する事が出来ません。そして、選択した PCELL の数や回転などを考慮して、PCELL が最適に変更されます。

実行コマンド構文：comb
Xl > comb < 分割数 >

引数は分割数のみで、スイッチ・オプションはありません。引数が省略された場合、デフォルト値である”3”が使われます。また、GUI モードに対応していません。

最後に

LISA によるカスタム機能は、単調で反復的なレイアウト作業を最小にするために、先端の自動設計を提供する事で、レイアウト設計のエントリー・タイムやデザイン・ルール違反を軽減します。PCELL は、抽象レベルをコンポーネント・レベルにあげて、オリジナルのセルを変更せずに、それぞれのセル・インスタンスのサイズや形状、コンテンツを変更する事が可能です。

このアプリケーション・ノートは、素子やセル、ブロック単位のカスタムのアナログ・ミックスド・シグナル設計の物理レイアウトを迅速にするために、カスタム・スクリプトの例を提供します。サンプル・スクリプトは付録A、B、C、D で提供されています。

付録A：ガードリングの自動作成のサンプル・スクリプト

```
define command/replace "make_ring ";
define action
parameter wire_space
parameter wire_width

parameter dummy_layer
parameter Usage
do begin

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Usage !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if(Usage EQL true) then begin
message_box("XI> make_ring <オプション> <間隔> <配線幅> <ダミー用レイヤのGDS II番号> \n"
&"\n"
&"   <オプション> \n"
&"/h  ・・・ ヘルプ \n"
&"\n"
&"   <間隔>     ・・・ 省略した場合は、GUIが立ち上がります。 \n"
&"   <配線幅>    ・・・ 省略した場合は、GUIが立ち上がります。 \n"
&"   <ダミー用レイヤのGDS II番号>  ・・・ 省略した場合は、デフォルト値の 63 が使われます。 \n"
,{});
end else begin

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! make "drop-down-list" of all cells. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!   (including library-cells, and except for the edited-cell. !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  all_cells = get_library_cell_list("");
  a=0;
  loop begin
    a=a+1;
    if(&all_cells[a] EQL &get_edited_cell_name()) then (leave loop);
  end;
  seq_remove_ith(all_cells, a);
  seq_sort(all_cells);
  cells_buf = buffer_create();
  buffer_insert_at_end(cells_buf, &all_cells);

  all_library = get_activated_library_list();
  b=1;
  loop beginåå
    b=b+1;
    if(b GTR all_library.size) then (leave loop);
    library_cells=get_library_cell_list(all_library[b]);
    seq_sort(library_cells);
    c=0;
    loop begin

      c=c+1;
      if(c GTR library_cells.size) then (leave loop);
      library_cells[c]=&all_library[b] &"::" &library_cells[c];
    end;
    buffer_insert_at_end(cells_buf, &library_cells);
  end;
  loop begin
    if(cells_buf.at_end) then (leave loop);
```

```
    buffer_scan(cells_buf, "\n");
    buffer_replace(cells_buf, cells_buf.index, cells_buf.index, "\t");
    buffer_advance(cells_buf, 1);
  end;
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! make "drop-down-list" of all layers. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!   (Except for the Derive Layer, Datatype=/0 & GDSnumber>0. !!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  layers = get_layer_list();
  c=1;
  loop begin
    if(c GTR layers.size) then (leave loop);
    if((get_gds_datatype(&layers[c]) NEQ 0) OR (get_gds_number(&layers[c]) LSS 1))
      then(seq_remove_ith(layers, c))
    else(c=c+1);
  end;
  layers_buf = buffer_create();
  buffer_insert_at_end(layers_buf, &layers);
  loop begin
    if(layers_buf.at_end) then (leave loop);
    buffer_scan(layers_buf, "\n");
    buffer_replace(layers_buf, layers_buf.index, layers_buf.index, "\t");
    buffer_advance(layers_buf, 1);
  end;
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Search Selected Instance !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  selectInst = (find objects (SEARCH_ANY_OBJECT) /selected /visible /seq_output);
  if(selectInst.size EQL 0) then (message_box(" 少なくとも１つ以上のオブジェクトを選択し
てください。",{}))
  else begin
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Create rule !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    if((wire_space*wire_width) EQL 0.0) then begin
      t1={ wire_space , " 配線リングとの間隔 "};
      t2={ wire_width , " 配線幅 "};
      t3={ dummy_layer , " ダミーレイヤ "};
      tt={t1,t2,t3};
      tt1={" オプションモード "};
      vars1 = (form create (tt) (tt1));

      if(&vars1.type NEQ "Null") then begin    !!! Form
        wire_space=vars1[1];
        wire_width=vars1[2];
        dummy_layer=vars1[3];
      end; !!!!!!! Form Cancel !!!!!!!
    end;
    if((wire_space*wire_width) GTR 0.0) then begin    !!! Execution
      Flag = &layers[dummy_layer];
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Make Shape !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      region mode /merge;
      i=0;
      loop begin
        i=i+1;
        if(i GTR selectInst.size) then (leave loop);
        inst = selectInst[i];
```

```
        inst_attr = string_to_lower(inst.shape);
        if((&inst_attr EQL "instance") OR (&inst_attr EQL "array")) then begin
          box (inst.bbox.xpos-(wire_space+wire_width/2)) (inst.bbox.ypos-(wire_space+wire_
width/2))  (inst.bbox.xsize+2*(wire_space+wire_width/2))  (inst.bbox.ysize+2*(wire_
space+wire_width/2)) /layer = (&Flag);
        end
    elseif((&inst_attr EQL "box") OR (&inst_attr EQL "polygon")) then begin
          select object (inst);
          copy layer (&Flag);
          resize selection (wire_space+wire_width/2) /over /grid;
          deselect all;
        end;
      end;
      deselect all;
      select layers (&Flag);
      merge selection;  polygonize;
      region mode /normal;
      SearchCriteria = {search_criterion_create(OA_LAYER, (&Flag), EQ)};
        select_shape = (find objects (SEARCH_ANY_SHAPE) /criteria = (SearchCriteria) /
selected /visible /seq_output);

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Action !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      j=0;
      loop begin
        j=j+1;
        if(j GTR select_shape.size) then (leave loop);
        vertex = select_shape[j].coords;
        seq_add_last(vertex, (select_shape[j].coords)[1]);

        if(vertex[1].x LSS vertex[vertex.size-1].x) then begin
          vertex[-1].x = vertex[-1].x+wire_width;
            wire (vertex) /width = (wire_width) /extend /extjoint /layer = (get_edited_
layer_name());
        end
    elseif(vertex[1].x GTR vertex[vertex.size-1].x) then begin
          vertex[-1].x = vertex[-1].x-wire_width;
            wire (vertex) /width = (wire_width) /extend /extjoint /layer = (get_edited_
layer_name());
        end
    elseif(vertex[1].y LSS vertex[vertex.size-1].y) then begin
          vertex[-1].y = vertex[-1].y+wire_width;
            wire (vertex) /width = (wire_width) /extend /extjoint /layer = (get_edited_
layer_name());
        end
    elseif(vertex[1].y GTR vertex[vertex.size-1].y) then begin
          vertex[-1].y = vertex[-1].y-wire_width;
            wire (vertex) /width = (wire_width) /extend /extjoint /layer = (get_edited_
layer_name());
        end;

      end;
      select layers (&Flag);
      delete selection;
    end; !!!!!!! Execution Cancel !!!!!!!
  end; !!!!!!! Search Cancel !!!!!!!
end; !!!!!!! Usage Cancel !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
end; !!!!!!! define end  !!!!!!!
define argument wire_space /positional /coerce_to=(Float) /default=0.0 /is_default;
define argument wire_width /positional /coerce_to=(Float) /default=0.0 /is_default;
define argument dummy_layer /positional /coerce_to=(Integer) /default=63
/is_default;
define argument h Usage /named /coerce_to=(Boolean) /value=true /is_default;
set parameter default Usage false;
complete command;
```

付録B：アレイ配置数の自動最大化のサンプル・スクリプト

```
define command/replace" max_array";
define action
parameter cell_name
parameter x_space
parameter y_space
parameter edge_space
parameter min_grid
parameter how_to
parameter Usage
do begin

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Usage !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if(Usage EQL true) then begin
message_box(" XI> max_array <オプション> <セル名> <X軸間隔> <Y軸間隔> <領域内エッジ間隔> <最小グリッド> \n"
&"\n"
&"   <オプション> 指定セルが存在しない場合は、GUI が立ち上がります。 \n"
&"/d ・・・ 配置する領域を、自分で指定します。 \n"
&"/s ・・・ 配置する領域を、選択しているオブジェクト内に指定します。\n"
&"/h ・・・ ヘルプ \n"
&"\n"
&"   <セル名>          ・・・ 省略した場合は、GUI が立ち上がります。 \n"
&"   <X軸間隔>         ・・・ 省略した場合は、デフォルト値の 0.2 が使われます。 \n"
&"   <Y軸間隔>         ・・・ 省略した場合は、デフォルト値の 0.2 が使われます。 \n"
&"   <領域内エッジ間隔> ・・・ 省略した場合は、デフォルト値の 0.08 が使われます。 \n"
&"   <最小グリッド>     ・・・ 省略した場合は、デフォルト値の 0.01 が使われます。 \n"
,{});
end else begin

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! make" drop-down-list" of all cells. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!   (including library-cells, and except for the edited-cell. !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  all_cells = get_library_cell_list(" ");
  a=0;
  loop begin
    a=a+1;
    if(&all_cells[a] EQL &get_edited_cell_name()) then (leave loop);
  end;
  seq_remove_ith(all_cells, a);
  seq_sort(all_cells);
  buf_string = buffer_create();
  buffer_insert_at_end(buf_string, &all_cells);

  all_library = get_activated_library_list();
  b=1;
  loop begin
    b=b+1;
    if(b GTR all_library.size) then (leave loop);
    library_cells=get_library_cell_list(all_library[b]);
    seq_sort(library_cells);
    c=0;
    loop begin
      c=c+1;
```

```
      if(c GTR library_cells.size) then (leave loop);
      library_cells[c]=&all_library[b] &"::" &library_cells[c];
    end;
    buffer_insert_at_end(buf_string, &library_cells);
  end;
  loop begin
    if(buf_string.at_end) then (leave loop);
    buffer_scan(buf_string,"\n");
    buffer_replace(buf_string, buf_string.index, buf_string.index,"\t");
    buffer_advance(buf_string, 1);
  end;
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Placement rule !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  if(cell_exists(cell_name) EQL false) then begin
    display(&cell_name &" is not exist.");
    t1={ &buf_string ,"セル名"};
    t2={ x_space ,"X軸方向のセル間隔"};
    t3={ y_space ,"Y軸方向のセル間隔"};
    t4={ edge_space ,"領域エッジとの間隔"};
    t5={ min_grid ,"最小グリッド"};
    t6={"by myself\t"
      &"by selected object" ,"領域の指定方法"}; !!! How to designate area
    tt={t1,t2,t3,t4,t5,t6};
    tt1={"オプションモード"};
    vars1 = (form create (tt) (tt1));

    if(&vars1.type NEQ"Null") then begin      !!! Form
      cell_name = vars1[1];
      x_space   = vars1[2];
      y_space   = vars1[3];
      edge_space= vars1[4];
      min_grid  = vars1[5];
      how_to    = vars1[6];
    end; !!!!!!! Form Cancel !!!!!!!
  end;

  if(cell_exists(cell_name) EQL true) then begin
    cell_bbox = get_cell_bbox(cell_name);
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! How to designate area !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    if(&how_to EQL"by selected object") then begin
      base_obj = (find objects (SEARCH_ANY_OBJECT) /åselecteåd /seq_output);
      if(base_obj.size EQL 0) then begin
        placement_area = seq_create_predicted(0);
        message_box"少なくとも1つ以上のオブジェクトを選択してください。",{});
      end else begin
!       duplicate selection 0 0;
        convert polygon(base_obj);
        tools cutbyvertex /regions /selected /regionvertices=4;

        base_obj = (find objects (SEARCH_ANY_OBJECT) /selected);
        placement_area = seq_create_predicted(base_obj.size);
        if(base_obj.size NEQ 0) then begin
          current_obj = base_obj.first;
          h=0;
          loop begin
            h=h+1;
```

```
            seq_add_last(placement_area , current_obj.bbox);
            if(h GEQ base_obj.size) then (leave loop);
            current_obj = base_obj.next;
          end;
        end;!!!!!!! Search End !!!!!!!
        undo;undo;   !!! convert & cutbyvertex
      end;!!!!!!! Selected instance Check
    end;!!!!!!! Mode-1 End !!!!!!!
    if(&how_to EQL" by myself") then begin
      placement_area = seq_create_predicted(1);
      designate_area = get_rect();
      if(&designate_area NEQ" nil") then (seq_add_last(placement_area , designate_area));
    end;!!!!!!! Mode-2 End !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Action !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    i=0;
    loop begin
      i=i+1;
      if(i GTR placement_area.size) then (leave loop);
      current_obj = placement_area[i];
      Nx=floor((current_obj.xsize+x_space-2*edge_space)/(cell_bbox.xsize+x_space));
      Ny=floor((current_obj.ysize+y_space-2*edge_space)/(cell_bbox.ysize+y_space));
      if(Nx*Ny NEQ 0) then begin
         Offset_x=round(((current_obj.xsize-(Nx*cell_bbox.xsize+(Nx-1)*x_space))/2), min_
grid); !!! Snap to center
         Offset_y=round(((current_obj.ysize-(Ny*cell_bbox.ysize+(Ny-1)*y_space))/2), min_
grid); !!! Snap to center
        xpos=current_obj.xpos+Offset_x-cell_bbox.xpos;
        ypos=current_obj.ypos+Offset_y-cell_bbox.ypos;
          array (xpos) (ypos) /cell = (&cell_name)   /row=(Ny) /col=(Nx) /xdeltarow=0 /
ydeltarow=(cell_bbox.ysize+y_space) /xdeltacol=(cell_bbox.xsize+x_space) /ydeltacol=0;
    end else (message_box" 警告：  指定領域が狭いので、セルを配置できません。",{}));
    end;
  end; !!!!!!! Form Cancel !!!!!!!
end; !!!!!!! Usage Cancel !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

end; !!!!!!! define end  !!!!!!!
define argument cell_name /positional /coerce_to=(String) /default="" /is_default;
define argument x_space /positional /coerce_to=(Float) /default=0.2 /is_default;
define argument y_space /positional /coerce_to=(Float) /default=0.2 /is_default;
define argument edge_space /positional /coerce_to=(Float) /default=0.08 /is_default;
define argument min_grid /positional /coerce_to=(Float) /default=0.01 /is_default;
define argument d how_to /named /coerce_to=(String) /value="by myself" /is_default;
define argument s how_to /named /coerce_to=(String) /value="by selected object" /is_de-
fault;
set parameter default how_to" by myself";
define argument h Usage /named /coerce_to=(Boolean) /value=true /is_default;
set parameter default Usage false;
complete command;
```

```
define command/replace "text_array";
define action
parameter Text_string
parameter Text_Number
parameter Pitch_Space
parameter Text_Size
parameter Bus_mode
parameter Direction_mode
parameter Usage
do begin

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Usage !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if(Usage EQL true) then begin
message_box("XI> text_array <オプション> <テキスト名> <数> <ピッチ> <大きさ> \n"
&"\n"
&"   <オプション> 方向が指定されていないと、GUI が立ち上がります。\n"
&"/b  ・・・ バス記述形式 （デフォルト） \n"
&"/nb ・・・ 全て同じ名前で記述 \n"
&"/u  ・・・ 上方向 \n"
&"/d  ・・・ 下方向 \n"
&"/l  ・・・ 左方向 \n"
&"/r  ・・・ 右方向 \n"
&"/h  ・・・ ヘルプ \n"
&"\n"
&"   <テキスト名> ・・・ 省略した場合は、デフォルト値の \"test\" が使われます。 \n"
&"   <数>       ・・・ 省略した場合は、デフォルト値の 1 が使われます。 \n"
&"   <ピッチ>    ・・・ 省略した場合は、デフォルト値の 1.0 が使われます。 \n"
&"   <大きさ>    ・・・ 省略した場合は、デフォルト値の 1.0 が使われます。 \n"
,{});
end else begin

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Attach the point you want to create !!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  message_box("Please click on the first point.",{});
  p1=get_point();
  if(&p1.type NEQ "Null") then begin    !!! Form-1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Setting Option !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    if(Direction_mode.size EQL 0) then begin
      s1={ &Text_string , "テキスト文字列"};
      s2={ Text_Number , "テキスト数"};
      s3={ Pitch_Space , "ピッチ間隔"};
      s4={ Text_Size , "テキストの大きさ"};
      s5={ Bus_mode , "バス記述の追加（例．test[\*]）"};
      s6={ "Up\t"
          &"Down\t"
          &"Left\t"
          &"Right" , "配置方向"};
      ss={s1,s2,s3,s4,s5,s6};
      ss1={" オプションモード "};
```

```
      vars1 = (form create (ss) (ss1));

      if(&vars1.type NEQ“ Null”) then begin    !!! Form-2
        Text_string=vars1[1];
        Text_Number=vars1[2];
        Pitch_Space=vars1[3];
        Text_Size=vars1[4];
        Bus_mode=vars1[5];
        Direction_mode=vars1[6];
      end; !!!!!!! Form-2 Cancel !!!!!!!
    end;
    if(Direction_mode.size GTR 0) then begin   !!! Execution

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Action !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      j=0;
      Text_Slope=0;

      PText_string=&Text_string;
      loop begin
        j=j+1;
        if(j GTR Text_Number) then (leave loop);
        if(&Bus_mode EQL“ True”) then(PText_string=(&Text_string&”[‟ &j&”]”));
        X=p1.x;
        Y=p1.y;
        if(&Direction_mode EQL“ up”) then begin
          Y=p1.y + (j-1)*Pitch_Space;
        end
    elseif(&Direction_mode EQL“ down”) then begin
          Y=p1.y - (j-1)*Pitch_Space;
        end
    elseif(&Direction_mode EQL“ left”) then begin
          X=p1.x - (j-1)*Pitch_Space;
          Text_Slope=90;
        end
    elseif(&Direction_mode EQL“ right”) then begin
          X=p1.x + (j-1)*Pitch_Space;
          Text_Slope=90;
        end;
           text (X) (Y) (&PText_string) /layer = (get_edited_layer_name()) /slope=(Text_
Slope) /MiddleCenter /height = (Text_Size) /width = (Text_Size);
      end;

    end; !!!!!!! Execution Cancel !!!!!!!
  end; !!!!!!! Form-1 Cancel !!!!!!!
end; !!!!!!! Usage Cancel !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

end; !!!!!!! define end  !!!!!!!
define argument Text_string /positional /coerce_to=(String) /default=”test” /is_default;
define argument Text_Number /positional /coerce_to=(Integer) /default=1 /is_default;
define argument Pitch_Space /positional /coerce_to=(Float) /default=1.0 /is_default;
define argument Text_Size /positional /coerce_to=(Float) /default=1.0 /is_default;
define argument nb Bus_mode /named /coerce_to=(Boolean) /value=false /is_default;
define argument b Bus_mode /named /coerce_to=(Boolean) /value=true /is_default;
set parameter default Bus_mode true;
```

```
define argument u Direction_mode /named /coerce_to=(String) /value=”up” /is_default;
define argument d Direction_mode /named /coerce_to=(String) /value=”down” /is_default;
define argument l Direction_mode /named /coerce_to=(String) /value=”left” /is_default;
define argument r Direction_mode /named /coerce_to=(String) /value=”right” /is_default;
set parameter default Direction_mode“ ”;
define argument h Usage /named /coerce_to=(Boolean) /value=true /is_default;
set parameter default Usage false;
complete command;
```

付録D：くし形抵抗への自動変換（PCELL を利用）のサンプル・スクリプト

```
define command/replace" comb";
define action
parameter split_n
do begin
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Set Parameters !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
select_cells = (find objects (SEARCH_INSTANCE) /selected /seq_output);
if(select_cells.size LEQ 1) then (message_box(" Please select two instances at least.",{}))
else begin
  X=select_cells[1].position.x;
  Y=select_cells[1].position.y;
  pitch=get_cell_bbox(select_cells[1].library&"::"&select_cells[1].cell).xsize;
!  pitch=select_cells[1].bbox.xsize;
  i=0;
  loop begin
    i=i+1;
    deselect all;
    if(i GTR select_cells.size) then (leave loop);
    select object (select_cells[i]);
    j=0;
    loop begin
      j=j+1;
      if(j GTR select_cells[i].params.size) then (leave loop);
      if((select_cells[i].paramnames)[j] EQL" N") then begin
         if((select_cells[i].rotation EQL 0) OR (select_cells[i].rotation EQL 180)) then
begin
          modify instance /selected /xpos = (X+pitch*(i-1)) /ypos = (Y)
                                   /params = ({split_n,TRUE,select_cells.size,i})
                                      /parnames = ({" N","CommonCentroid","Common_N","Shape_
Style"});
          leave loop;
        end
     elseif((select_cells[i].rotation EQL 90) OR (select_cells[i].rotation EQL 270)) then
begin
          modify instance /selected /xpos = (X) /ypos = (Y+pitch*(i-1))
                                   /params = ({split_n,TRUE,select_cells.size,i})
                                      /parnames = ({" N","CommonCentroid","Common_N","Shape_
Style"});
          leave loop;
        end;
      end;
    end;!!!!!!! Parameter Loop
  end;!!!!!!!  Select Loop
end;!!!!!!! Selected instance Check
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Set Placement !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

end; !!!!!!! define end  !!!!!!!
define argument split_n /positional /coerce_to=(Integer) /default=3 /is_default;
complete command;
```