

Expert におけるレイアウト操作 自動化のための LISA スクリプト作成

はじめに

Expert における数多くのレイアウト設計タスクは、LISA コマンド言語を使用して自動化できます。手動で繰り返し実行するタスクに対して LISA Expert Interface (XI) スクリプトを作成すると、プロジェクトに要する時間を短縮できます。また、手動によるレイアウト上のエラーを削減することでレイアウト品質を向上できます。LISA コマンドを使用して Expert のメニューコマンドをコピーできるので、XI スクリプトを作成してほとんどのレイアウトタスクを自動化できます。これらのスクリプトは、一度書き込むと、別の設計プロジェクト(異なるテクノロジ・プロセス・ノードを持つプロジェクトを含む)で使用するためにすばやく変更できます。

本稿では、2 つの XI スクリプト例を用いて、手動では非常に時間がかかるレイアウト操作の実行を LISA スクリプトでいかに短縮できるかを示します。これらの例には、設計プロジェクトのレイアウト段階で発生するさまざまなタスクに適用可能ないくつかの基本的な LISA コマンドが含まれています。

LISA XI スクリプトの作成

Expert には、XI スクリプト・エディタが内蔵されています。XI スクリプト・エディタは、メインメニューの[ツール]→[XI スクリプト]→[スクリプト・パネル]を選択して開きます(図 1 参照)。また、『Expert ユーザーズ・マニュアル』に、XI スクリプト、および LISA コマンドを使用して Expert 機能を実行するためのシンタックスについて記載しています。スクリプトを書き込むと、スクリプト・パネルのツールバーの[実行]ボタンをクリックするか、スクリプト・パネルのメインメニューの[スクリプト]→[実行]を選択してスクリプトを実行します。

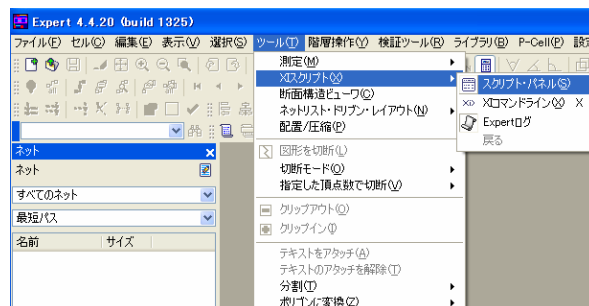


図 1. XI スクリプト・パネルの起動方法

メモリアレイにテキスト・ラベルを配置する スクリプト例

1 つ目の XI スクリプト例(図 2)は、短い LISA スクリプトです。この例では、メモリエルのアレイを含んだユーザ指定のレイアウトセルを開き、メモリ・ビットラインとワードラインの信号パスにインクリメントされたテキスト・ラベルを配置します。信号ノードにテキスト・ラベルを使用すると、LVS デバッグを大幅に向上できますが、メモリ回路の大規模なアレイでこれらのラベルを手動で配置すると非常に時間がかかります。

スクリプトの冒頭で、後でコードの中で使用する変数を定義します。変数定義ステートメントの後に、スクリプトは cell open コマンドを使用して編集対象のメモリアレイ・レイアウトを開きます。このスクリプトでは、プロジェクト・データベースである.eld ファイルを読み込まないので、ユーザはスクリプトを実行する前にデータベースを読み込む必要があります。次のコマンド・シンタックスを用いて、.eld ファイルを XI スクリプトに読み込みます。

```
PROJECT LOAD proj_name [tech_name]
[/outputname=output_name] [/readonly]
```

```
! This script labels row and column bit line
! and word line with text labels
bl_pitch= 7.4;
wl_pitch= 7.8;
Xorg=-1.2;
Yorg=0;
bl_num=0;
wl_num=0;
cols=64;
rows=64;
cell_name="array_top";
cell open (cell_name);
LOOP BEGIN      !bit line loop
  IF (bl_num EQL cols) THEN (leave loop);
  text (Xorg+bl_num*bl_pitch) (Yorg-3) ("bl_"&bl_num)
  /height = 2 /width = 0.8 /bottomcenter /layer = "metall1";
  bl_num=bl_num+1;
END;
LOOP BEGIN      !word line loop
  IF (wl_num EQL cols) THEN (leave loop);
  text (Xorg) (Yorg+wl_num*wl_pitch) ("wl_"&wl_num)
  /height = 2 /width = 0.8 /bottomright /layer = "poly";
  wl_num=wl_num+1;
END;
```

図 2. テキスト・ラベル配置を実行する XI スクリプト

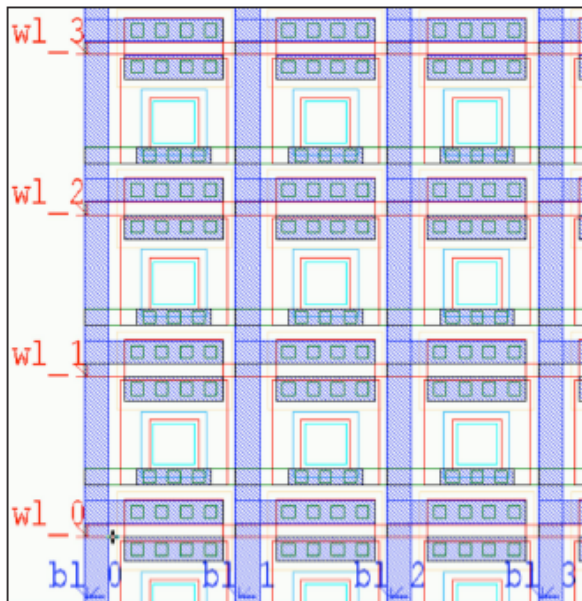


図 3. ラベルを付けたアレイのレイアウト表示

次の行は LOOP BEGIN ステートメントです。このステートメントは各ビットライン列をループし、metal1 縦ビットラインにテキスト・ラベルを配置します。テキストは bl_0 から始まり、ループごとにインクリメントされます。同様の手順を用いて、ポリワードライン行にテキスト・ラベルを配置します。図 3 は、レイアウト結果の拡大表示です。

コンタクト検索および置換のスクリプト例

2 つ目の XI スクリプト例では、レイアウト・データベースでコンタクト形状のサイズ変更を実行します。この例では、ある特定サイズのコンタクト形状を検索し、その形状を新しいコンタクトを含むレイアウト・セルのインスタントに置換します。この例の記述を図 4 に示します。

現在読み込まれているレイアウト・データベースにおけるすべてのセルのリストは、get_library_cell_list コマンドを用いてアレイ変数 cells に保存されます。次にループを生成します。アレイ変数 cells の各エントリで反復し、特定の検索条件を指定した find objects コマンドを用いてサイズが 0.6um であるすべてのコンタクトを検索します。find objects コマンドによる検索結果が 0 よりサイズが大きい場合、つまり、セルに 0.6x0.6um のコンタクトが検出されたことが示された場合、2 回目のネストループが開始されます。ループ内では、Select Object コマンドにより、検出された各形状が選択され、削除されます。検索によって戻された X 値および Y 値を使用して、正しいサイズ 0.4um のコンタクト・インスタンスが削除された形状の原点に配置されます。

```
! Script to replace a flat contact shape with
! an instance of a smaller contact

cells = get_library_cell_list(""); ! store all cells in array "cells"
i=1;
LOOP BEGIN !loop through all cells
  cell open (cells[i]);
  DISPLAY(cells[i]);
  cont_Op6 = (find objects (SEARCH_BOX)
  /criteria = ((search_criterion_create(OA_LAYER, "contact", EQ),
  search_criterion_create(OA_SIZEX, 0.6, EQ),
  search_criterion_create(OA_SIZEY, 0.6, EQ))) /seq_output
  );
  num_cont=cont_Op6.size;
  DISPLAY (num_cont);
  IF ((cont_Op6.size) NEQ 0) THEN BEGIN
    j=1;
    LOOP BEGIN
      Select Object (cont_Op6[j]);
      pt=cont_Op6[j].position;
      DELETE SELECTION;
      instance (pt.x) (pt.y) /cell="cont_Op4";
      j = j+1;
      IF (j EQL (num_cont+1)) THEN (leave loop);
    END; !end of loop to delete 0.6um contact and replace
    !with "cont_Op4" instance
  END; !end of condition where num_cont > 0 in a cell
  cell close;
  i = i + 1;
  IF (i EQL (cells.size)) THEN (leave loop);
END;
```

図 4. コンタクト置換を実行する XI スクリプト

図 4 で使用した find objects コマンド例を変更することで、特定のレイアウト・オブジェクトを検出するために大規模なレイアウト・セルまたはレイアウト・データベース全体で検索を必要とする、さまざまなタスクに適用できます。特定のレイアウト・オブジェクトには、ユーザ定義の一連の検索条件にマッチする形状、ワイヤ、テキスト、インスタンスが含まれます。

まとめ

LISAコードを使用して XI スクリプトを作成すると、Expert でシステマティックかつ反復的なレイアウト機能を効率的に実行できます。2 つの例を用いて、手動で実行した場合に非常に時間がかかる設計タスクをいくつかの基本的な LISA コマンドを使用して自動化する方法を示しました。プロジェクトを通じて再活用可能、かつ今後のプロジェクトに移行可能な XI スクリプトを少し時間をかけて作成することで、設計のレイアウト段階におけるサイクルタイムを大幅に短縮できます。