

Creating LISA Scripts to Automate Layout Operations in Expert

Introduction

Many layout design tasks in Expert can be automated using the LISA command language. Creating LISA Expert Interface (XI) scripts for repetitive manual tasks can reduce project time and increase layout quality by reducing the chance of manual layout errors. Since Expert menu commands can be replicated using LISA commands, an XI script can be created to automate almost any layout task. Once written, these scripts can be quickly modified for use in different design projects including ones with different technology process nodes.

Two XI script examples are provided to show how short LISA scripts can perform layout operations that would be very time consuming if done manually. These examples contain several basic LISA commands that can be applied to various tasks that may arise during the layout phase of a design project.

Creating a LISA XI Script

Expert has a built in XI script editor that can be opened using the following menu selection: Tools >> XI Script >> Script Panel, shown in Figure 1. The Expert User Manual contains a chapter devoted to XI scripts and provides the syntax to access Expert functions using LISA commands. Once a script is written it can be executed by clicking the "Run" icon on the Script Panel tool bar, or by selecting Script >> Run in the menu.

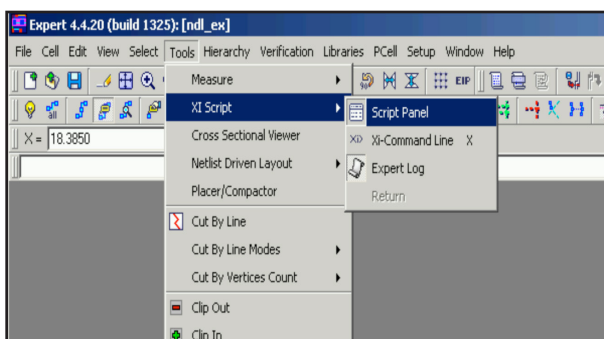


Figure 1. Accessing the XI Script Panel.

Memory Array Text Labeling Example Script

The first XI script example shown in Figure 2 is a short LISA script that opens a user specified layout cell containing an array of memory cells and places incremented text labels on the memory bit line and word line signal paths. The use of text labels on signal nodes can greatly improve LVS debugging, but the placing of these labels on large arrays found in memory designs would take a prohibitive amount of time if done manually.

The beginning of the script defines the variables that will be used later in the code. After the variable definition statements, the script uses the "cell open" command to open the memory array layout that will be edited. This script does not load the project database .eld file, and so it is left to the user to load the database before executing the script. The .eld file can be loaded in the XI script using the following command syntax:

```
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 = "metal1";
  bl_num=bl_num+1;
END;                !end of bit line loop

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;                !end of word line loop
```

Figure 2. Text Labeling XI Script.

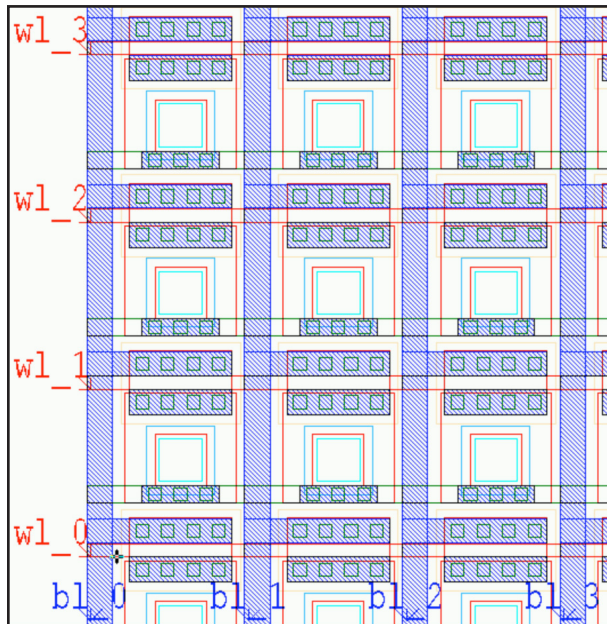


Figure 3. Layout View of Labeled Array.

Next is the “LOOP BEGIN” statement that will loop through each of the bit line columns and place a text label on the metal1 vertical bit line. The text begins as “bl_0” and is incremented each time through the loop. The same procedure is used to place text labels on the rows of poly word lines. A close-up view of the resultant layout is shown in Figure 3.

Contact Search and Replace Example Script

The second example XI script performs the task of resizing contact shapes in a layout database by searching for any contact shape of a certain size and replacing that shape with an instance of a layout cell containing the new contact. The code for this example is found in Figure 4.

A list of all cells in the currently loaded layout database is stored in the array variable “cells” by using the “get_library_cell_list” command. Then a loop is created that iterates through each of the entries in the “cells” array variable and performs a search for all drawn contacts of 0.6um size by using the “find objects” command with specific search criteria. A second nested loop begins if the “find objects” search results has a size greater than zero, indicating that a 0.6x0.6um contact was found in the cell. Inside the loop, the “Select Object” command selects each shape that was found and deletes it. By using the stored X and Y position values returned by the search, an instance of a contact that is the correct size of 0.4um is placed at the origin of the deleted shape.

```

! 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=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;

```

Figure 4. Contact Replacement XI Script.

The example of the “find objects” command used in Figure 4 can be modified to apply to any number of tasks that require searching through a large layout cell or an entire layout database to find specific layout objects including shapes, wires, text, and instances that match a series of user defined search criterion.

Conclusion

Creating XI scripts with LISA code is an economical way to perform systematic and repetitive layout functions in Expert. The two examples shown above illustrate how several basic LISA commands are used to create procedures that automate design tasks that would be very time consuming if done manually. Investing a small amount of time to create an XI script that can be reused throughout a project and even migrated to future projects can have a substantial impact in reducing the cycle time of the layout phase of a design.