Simulation Standard

Connecting TCAD To Tapeout

A Journal for CAD/CAE Engineers

Hierarchical Layout versus Schematic

1. Introduction

A new Hierarchical Layout versus Schematic (HLVS) system that provides significant improvement in verification of huge circuits is described. Other of LVS verification tools perform a netlist flattening and comparison in transistor level [1, 2]. These tools are based on standard graph isomorphism algorithms and are sufficiently efficient in practice. Their running time is almost linear in the size of compared devices: O (N_m), where N is the number of compared devices and m is value a little larger than 1. These tools are not effective enough to verify circuits containing tens of millions of transistors. There is a need for hierarchical comparison of two netlists. This proposed hierarchical approach does not flatten the netlist. The comparison is carried out for separate cells. Once a cell is checked, its contents are not used later in comparison, only the verification of connectivity to ports of this cell is performed. This approach enables a large decrease in the number of compared elements and, consequently, the run time of HLVS verification.

2. Some Features of *Guardian HLVS*

Compared netlists should satisfy to some conditions to make hierarchical verification more effective:

- Both compared netlists should have hierarchical structure. Otherwise, if at least one netlist is flat, you will not obtain the advantages of hierarchical approach of LVS verification. Notice, exactly the same hierarchical structure of netlists is not required. The netlists can have different number of cells and levels of hierarchy
- ***Guardian HLVS*** can handle different number of ports of equivalent cells from opposite netlists. Some cells can have feed ports. The feed port is a port unconnected to devices of the cell. ***Guardian HLVS*** is ignored feed ports during LVS verification. But the equivalent cells from opposite netlists should have the same number of "real" ports (ports connecting to at least one device in cell by a net)
- ***Guardian HVLS*** can merge and reduce the devices connected in parallel or in series inside each cell, if corresponding options are selected. If the cells from compared netlists are recognized as hierarchically equivalent, the content of these cells is ignored and the reduction operations for their elements are inadmissible
- HLVS can handle the primitive port swappability. The logical equivalence of gates of complex logical elements obtained in reduction stage can be carried up from transistor level to respective ports of cell. Also special cases of port swappability can be recognized (for details, see section 5)

3. Hcells

Abbreviation **Hcell** is used for hierarchically compared cells. You can specify the pairs of hierarchically compared cells in initial correspondence file by statement

```
.HCELL <SchSubckt>=< LaySubckt >,
```

where SchSubckt is the name of subcircuit in schematic netlist, and LaySubckt is name of subcircuit in layout netlist. It's expected that the Hcells exist in both netlists and the cells of each pair are equivalent. A pair is ignored if a subcircuit of this pair does not exist in a netlist. HLVS verification is performed for each pair, and the hierarchically equivalent cells can be used later in verification as

Continued on page 2 ...

INSIDE

SILVACO INTERNATIONAL

whole blocks without considering their contents. Also you can specify the top cells in initial correspondence file by the statement

```
.TOPCELL <SchSubckt>=< LaySubckt >,
```

where SchSubckt and LaySubckt are the subcircuits of schematic and layout netlists, correspondingly, which interpreted as top cells for hierarchical LVS verification. If this statement is not in the initial correspondence file, the top level subcircuits of netlists are considered top cells.

Often the schematic and layout can contain thousands of cells and the preparation of whole list of Hcells is difficult. In this case you can use the option "Hierarchical mode" in "HLVS Setting Panel". If this option is switched on, ***Guardian HLVS*** tries to determinate all subcircuit pairs that are possibly equivalent. Then LVS verification is performed for each pair from prepared list. All these pairs are listed in Log report and you can use this list for preparation of Hcell pairs for initial correspondence file.

The option "Hierarchical mode" can be used along with the option "Match same name subcircuit" in "General Setting Panel". If both options are switched on, all pairs of subcircuits with the same names will be interpreted as Hcells. You can control case sensitivity of subcircuit names by using the option "Case-sensitive matching of names" in "General Setting Panel". When the initial correspondence file contains a list of Hcells and the option "Hierarchical mode" is switched on, only the Hcells from initial correspondence file take part in HLVS verification.

Schematic or layout can include several equivalent cells, in other words, one cell of schematic can correspondence to many cells in layout (one-to-many correspondence), and otherwise, one cells from layout can correspondence to many cells in schematic (many-to-one correspondence). You may specify such correspondences in initial correspondence file. But many-to-many correspondences are inadmissible. The following is the one-to-many correspondence example of Hcells, where one schematic subcircuit NAND2X will be compared with each of three subcircuits of layout netlist:

```
.HCELL NAND2X = NAND2XT

.HCELL NAND2X = NAND2XW

.HCELL NAND2X = NAND2XZ
```

Now let the initial correspondence file contains the following statements:

```
.HCELL NAND2X = NAND2XT

.HCELL NAND2XW = NAND2XZ

.HCELL NAND2XW = NAND2XT
```

If ***Guardian HLVS*** detects that the cells of first and second statements are equivalent, then the third one is the example of many-to-many correspondence. Such statement is ignored and corresponding message appears in Message report: "Many-to-many subcircuit correspondence NAND2XW = NAND2XT is ignored". If the subcircuits of first or second statements are not equivalent, then third pair is processed for equivalence.

Notice some cells can be unspecified as hierarchically equivalent in HCELL statements. Such cells will be expanded down to the next levels of hierarchy or device level.

4. Hierarchical Verification Algorithm

The operation flow of this algorithm used for hierarchical verification of netlists is depicted in Figure 1. Hierarchical netlists are read in the first stage "**Netlist reading**" and the data structures storing the information about these netlists are created. Filtering and merging operations for devices of the same types are performed in the second stage "**Inside Cell Preprocessing**".

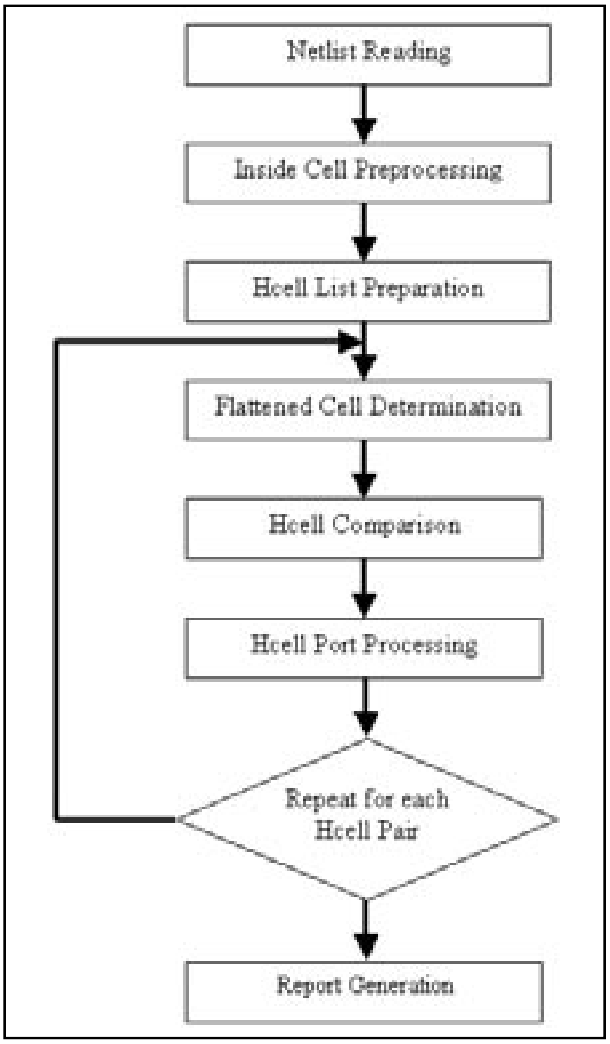


Figure 1. Operation Flow for Hierarchical Verification.

Then a list of Hcell pairs is formed in "**Hcell List Preparation**" stage based on Initial Correspondence File information or options in Setting Panels (see previous section for details). All incorrect Hcell statements, for example, statements containing non-existent cells, are ignored, and corresponding messages are put in Message report. The list of Hcell pairs is sorted such that a cell of list can not contain the instances of cells placed after the cell, in other words, at first the cells defined then can be instanced. All following stages are performed for each pair of Hcells from prepared list.

"**Flattened Cell Determination**" stage determines the subcircuit instances that must be flattened to bring in correspondence of the hierarchies of compared Hcells. All subcircuits belonging to each pair of compared Hcells are determined. The subcircuits that do not belong to Hcell list are always marked for flattening. The number of instances of each Hcell contained in compared Hcells is calculated. The Hcells with different number of instantiation are also marked for flattening. The rest of subcircuits are marked as non-flattened.

"**Hcell Comparison**" is the main stage of hierarchical verification. The flattening is performed for instances of subcircuits marked for flattening in previous stage. Here it is necessary to note that the flattening is carried out down to the nearest level of cells that were earlier detected as hierarchically equivalent. This allows ***Guardian HLVS*** to reduce the number of compared elements and to decrease running time of comparing algorithm. Then the reduction operations are performed for devices if they exist in compared cells. The c omparing process is done using refinement techniques for graph isomorphism problem [1, 4, 5]. During comparison ***Guardian HLVS*** collects the information about swappability of Hcell ports. The reports are generated for each compared Hcell pair. You can see in "Comparison Summary" section of Log report for which primitive elements (devices and hierarchically equivalent cells) the comparison has been done. The Unmatched report lists all nets and instances of devices and cells that have not been matched by HLVS. If unmatched elements have been detected, compared Hcells can't be used as hierarchically equivalent in comparison another Hcells. Notice you can interrupt the verification process when first pair of non-equivalent Hcells was detected by using the corresponding option in "HLVS Setting Panel".

If unmatched elements have not been detected, the "**Hcell Port Processing**" stage is performed. At this stage Guardian HLVS determines the swappability (logical equivalence) of Hcell ports based on information prepared on the previous stage. The next section describes the swappability types detected by ***Guardian HLVS***. After port swappability detection the correspondence of equivalent ports of schematic and layout Hcells are assigned. This port correspondence is used to compare Hcells at the next levels of the hierarchy. This stage is not performed for top cells.

Consider a simple example that explains this algorithm. Let the schematic and layout netlists have the following hierarchies (see Figure 2).

Usually the schematic has more complex hierarchy than layout. So, the schematic in this example has an additional level of hierarchy and additional cells Cell_4, Cell_5. Let the cells with corresponding names Cell_1, Cell_2, Cell_3, and Top in schematic and layout netlists be equivalent. Cell_3 in schematic consists of an instance of cell Cell_5 and a set of transistors. Cell_3 in layout

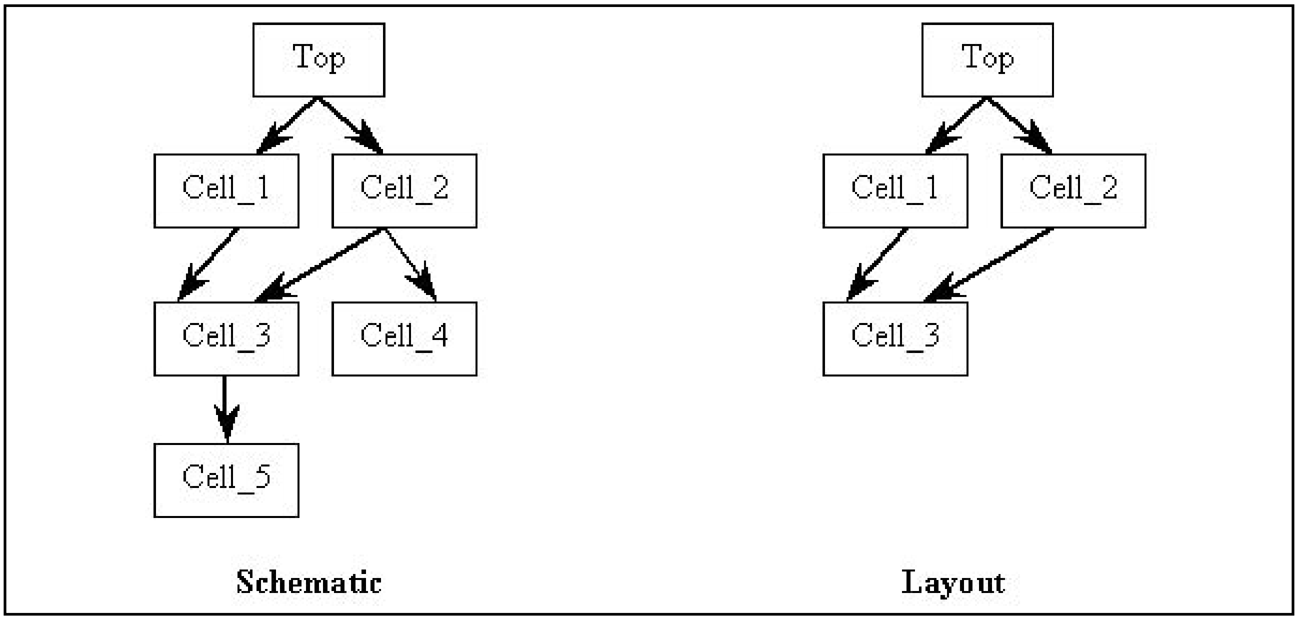


Figure 2. The hierarchies of schematic and layout

consists of a set of transistors only. Cell_2 in schematic includes the instances of cells Cell_3 and Cell_4. Cell_2 in layout consist of the instance of cell Cell_3 and a set of transistors. Cell_1 in schematic and layout includes only the instances of cell Cell_3. Top cell of schematic and layout consist of the instances of cells Cell_1 and Cell_2.

Guardian HLVS forms the following list of Hcells after the stage "Hcell List Preparation": Cell_3 and Cell_3, Cell_2 and Cell_2, Cell_1 and Cell_1. Cell_5 will be flattened, when Hcells Cell_3 are compared. Port correspondence for these Hcells is stored. When the Hcells Cell_2 are compared, Cell_4 is flattened, but Cell_3 is not flattened and comparison is performed for instances of Hcells Cell_3 using information about port correspondence obtained before. During the comparison of Hcells Cell_1 only the instances of cells Cell_3 are considered. Comparison of top cells is performed for instances of cells Cell_1 and Cell_2 only.

5. Hcell Ports

In general case, there is no swappability for ports of hierarchical cells. ***Guardian HLVS*** determines logical equivalence of Hcell ports in some special cases that are listed below.

- **Transistor level port swappability**

  When the parallel and serial reduction is performed in preprocessing stage, the complex logical configurations with swappable gates (base) terminals can be obtained [6]. Consider the Figure 3 for example.

  The logical configuration Cl is created as result of parallel reduction of the transistors M1, M3 and the transistors M2, M4, and then serial reduction for obtained elements. For this example, it's possible to interchange the gates G1 and G3, G2 and G4, and the pairs of gates G1-G3 and G2-G4 at transistor level. Port swappability at Hcell level is permitted only for first level. So, the ports G1 and G3, G2 and G4 of Hcell A are swappable, but the port pairs G1-G3 and G2-G4 are not swappable.

  Ports of Hcells that are connected to swappable terminals of device (for example, swappable source and drain of MOSFET transistor) are also swappable, if they are not connected to another devices of Hcell.

- **RAM cells recognition**

  Guardian HLVS detects the six-transistor SRAM cells (see Figure 4).

  The nets B and B_ must only be connected to ports of Hcell containing the SRAM configuration and not connected to another devices in Hcell. The net W can be or not be a port of Hcell and can be connected to other devices. The nets VDD and VSS are connected to power and ground nets directly or through the ports of Hcell and can have the connections with other devices in cell. The following pairs of transistors M1 and M6, M2 and M4, M3 and M5 must have the same device type and the same bulk terminals. Internal nets adjacent to groups of transistors M1, M2, M3 and M4, M5, M6 must not be connected to another devices. For such SRAM configuration ***Guardian HLVS*** detects that ports B and B_ of Hcell are logical equivalent and can be swappable.

  Also HLVS recognizes the eight-transistor SRAM cells (see Figure 5).

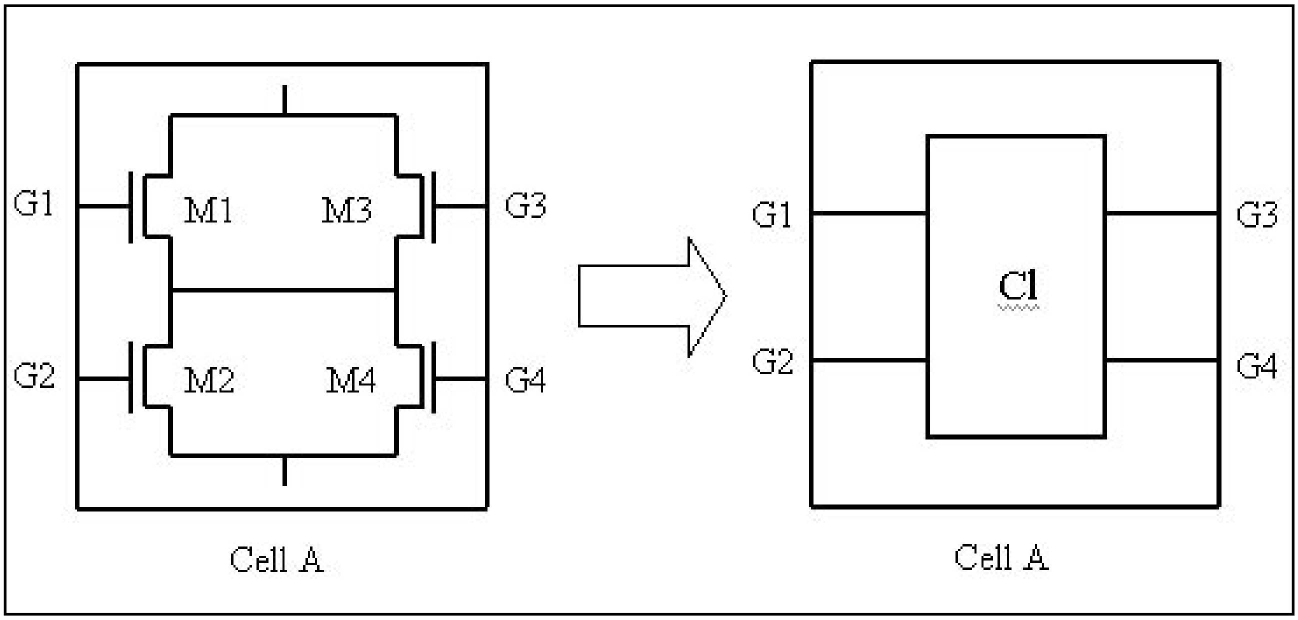


Figure 3. Port swappability for logical configurations.

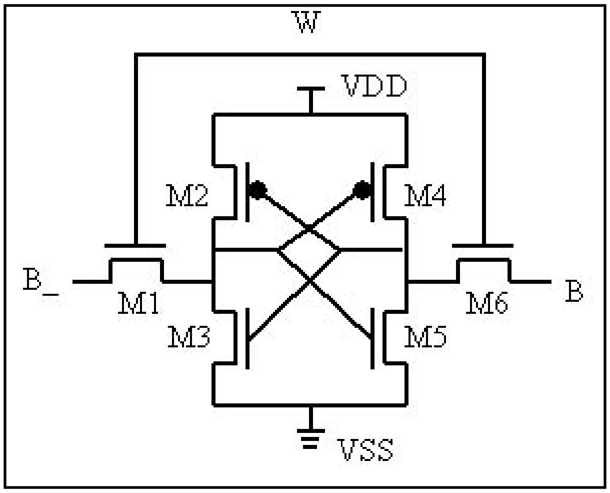


Figure 4. Six-transistor SRAM.

Guardian HLVS detects that port A is logical equivalent to port B, A_ to B_.

- **Hcell level port swappability**

Guardian HLVS tries to extend the information about port swappability from transistor level up as far as possible. The ports of Hcell are recognized as logically equivalent, if they are connected to the pins of instances of internal cell by the nets that do not have another connections and these instance pins correspond to swappable ports of internal cell. For example, see Figure 6.

Cell B contains two instances X1 and X2 of cell A. The ports P1, P2, and P3 of the cell A are logical equivalent. Then Guardian HLVS will detect the swappability of ports X1/P1, X1/P2, X1/P3 and the ports X2/P1, X2/P2, X2/P3, if they have only connections with ports P1, P2, P3 of cell A.

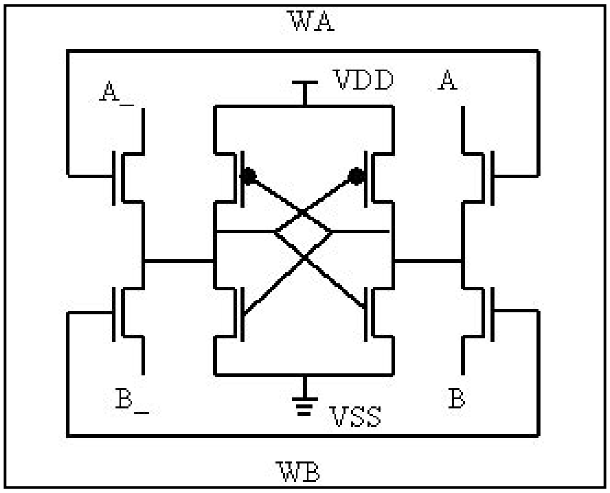


Figure 5. Eight-transistor SRAM.

Conclusion

A hierarchical approach for Layout versus Schematic verification has been proposed and realized in ***Guardian HLVS***. This approach allows you successfully to verify LVS for large SOC chips.

References

1] C.Ebeling, “GeminiII: A Second Generation Layout Validation Program”, IEEE International Conference on Computer-Aided Design, 1988, pp. 322-325.

2] L.Williams, “Automatic VLSI Layout Verification”, Proceedings of the 18th ACM/IEEE Design Automation Conference, 1981, pp. 726-732.

3] P.Batra, D.Cookie, “Hcompare: A hierarchical netlist comparison program”, Proceedings of the 29th ACM/IEEE Design Automation Conference, 1992, pp. 299-304.

4] D.G.Corneil, C.G.Gotlieb, “An algorithm for graph isomorphism,” Journal of the ACM, N 17, 1970, pp.51-64.

5] D.G.Corneil, D.G.Kirkpatrick, “A theoretical analysis of various heuristic for the graph isomorphism problem,” SIAM Journal of Computing, N 9, 1980, pp.281-297.

6] “Complex parallel-series reduction”, Simulation Standard, V. 13, N 3, 2003, pp.4-6.

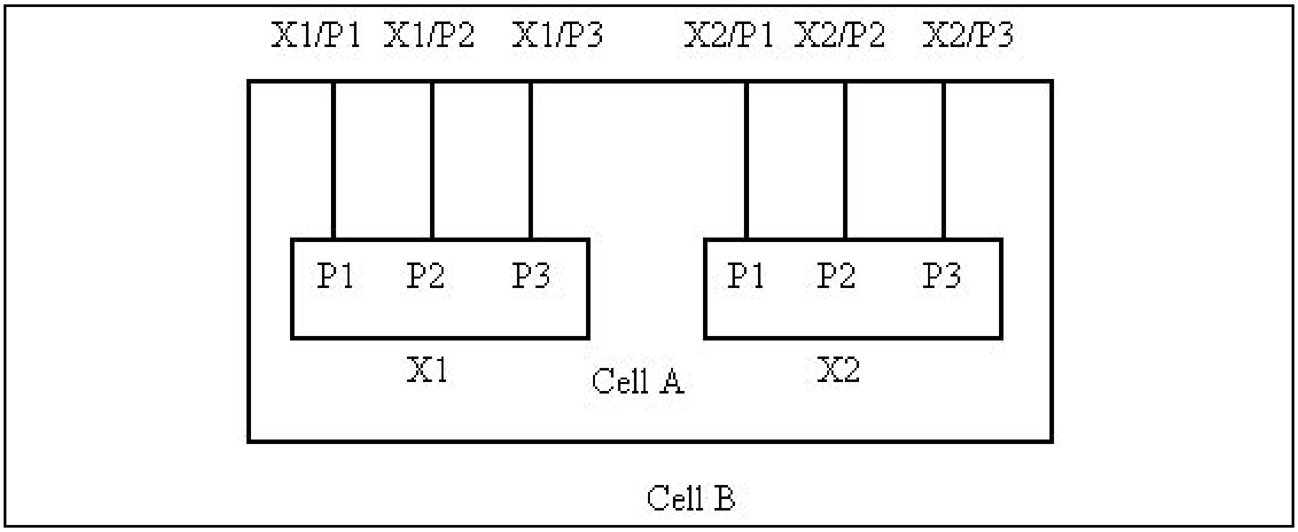


Figure 6. Hcell level port swappability.

Design of Optimal Spiral Inductors in *Expert*

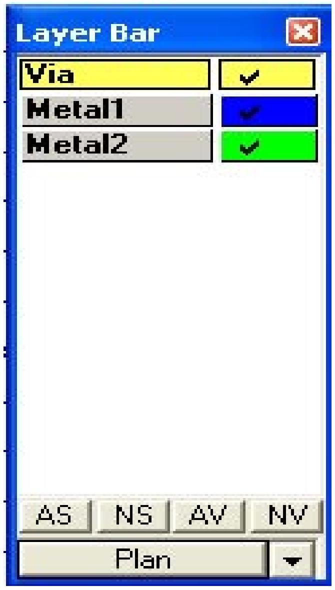
Goran Stojanovic, University of Novi Sad, Serbia

Introduction

One of the key components in many RF ICs applications is the inductor. It is very important that the inductor has optimal design, meaning optimal geometry with the best possible characteristics [1]. Our computer-aided optimization technique using geometric programming (GP) has been used to find the optimum design for spiral inductors with different layout [2-4]. The goal is selection of the best geometry of inductors to find optimal values of inductor parameters (the number of turns and layout dimensions) and then drawing layouts of optimal inductors. Silvaco ***Expert*** is a powerful tool for drawing spiral inductors with different mask geometry (square, octagonal or circular). In this paper we presents a simple way to draw layouts of optimal inductors in ***Expert***, without using parameterized cells.

Layout of Square Spiral Inductors

The purpose of ***Expert*** Wiring tool is to draw a wire in the active layer. In case of inductors we use Metal2 layer to draw all spiral conductor segments, Metal1 to draw underpass, and layer Via to link them. The current layer name is displayed in the Wire panel and all wires are built in this layer until altered. To run a wire, select **Tools > Wiring**… menu command. This launches the Wire panel, which can be seen in Figure 1. In this panel we enter input data for Wire width and Wire spacing, which are output results of our geometric programming optimization algorithm. When the wiring parameters are adapted, then drawing of inductor layout can start.



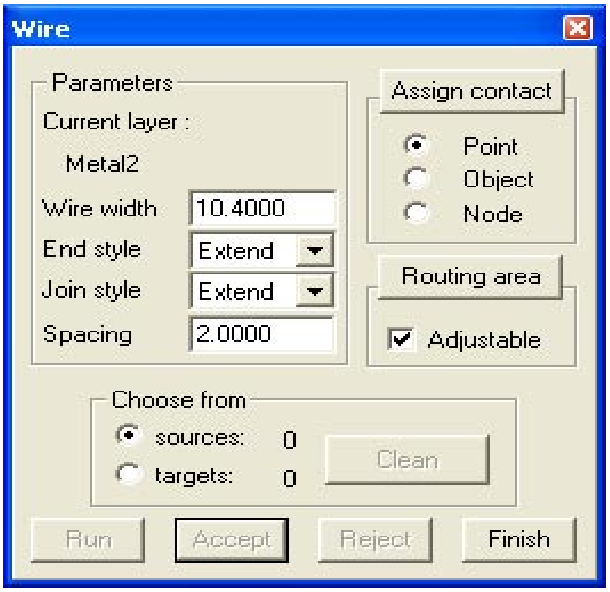
Layer Bar

Via ✓

Metal1 ✓

Metal2 ✓

Figure 2. The view of Layer bar.



Wire

Parameters

Current layer :

Metal2

Wire width: 10.4000

End style: Extend

Join style: Extend

Spacing: 2.0000

Assign contact

● Point

○ Object

○ Node

Routing area

☑ Adjustable

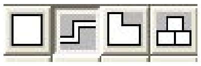
Choose from

● sources: 0

○ targets: 0

Figure 1. Adjustment of wire width and spacing.

To start drawing inductor select **Edit > Create object > Wire** (or second icon)



When drawing the angle of 90º should be adjusted, what is obtained by selecting **Edit > Angle Mod > 90 deg** (or by selecting the third icon from the picture,

the first icon designates that we have selected any angle, the second one means we have selected 45º angle and the third one designates that we have selected 90º angle). In this way we acctually adjust the angle used to start our wire shaped structure. If we select E**dit > Angle Mod > All** and select **Edit > Numeric input** (icon)

we can choose arbitrary angle by which our structure will move, and at the same time we can change the angle and width of the metal structure while we are drawing. In the course of drawing once we achive the desired length by the click of the mouse we designate the end of drawing because each and every part of the inductor has to be drawn with given precision in order to get suitable distance between the segments, in our cases 2 μm or 4 μm spacing.

It is clear that there are two metal layers: metal2 and metal1, as well as the layer for via. All these layers differ in color (Figure 2). The metal2 is green, metal1 is blue and via is yellow. In the metal2 the spiral itself is real-

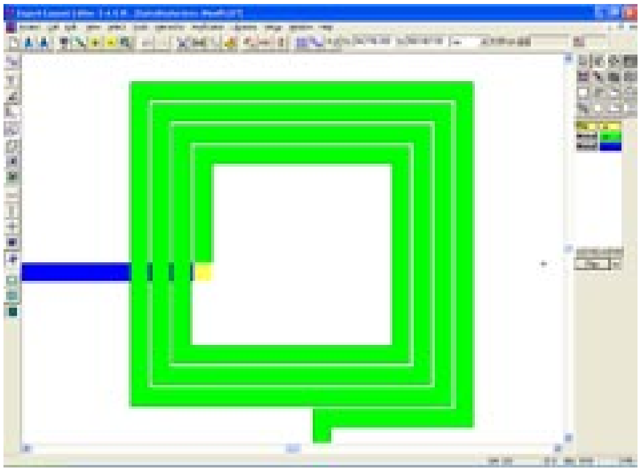


Figure 3. The square spiral inductor in ***Expert*** (zooming).

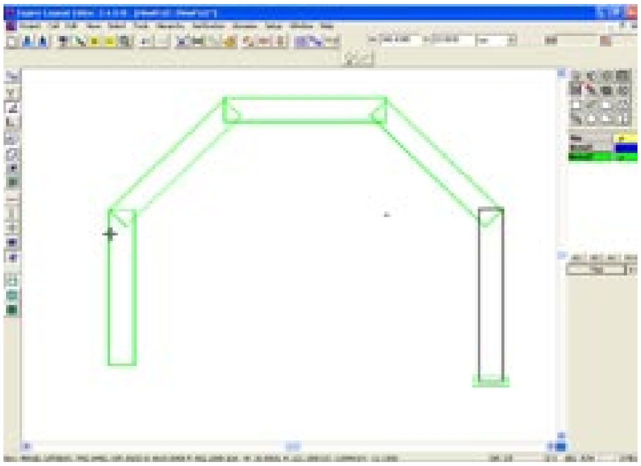


Figure 4. Drawing process of an octagonal inductor in ***Expert***.

ized, while metal1 serves for creation of the underpass conductor for linking of the spiral's center to the second external port.

It also should be taken into account that in the course of drawing the parts of wire should be prolonged in order to prevent the cut interruption of the structure.

Lengths of the first four segments of the structure from the Figure 3 are:

$$l_1 = d_{IN} - s - \frac{w}{2} \tag{1}$$

$$l_2 = d_{IN} + w \tag{2}$$

$$l_3 = d_{IN} + w \tag{3}$$

$$l_4 = (d_{IN} + w) + (s + w) \tag{4}$$

where d_{IN}, *s*, w are layout parameters. The iner diameter is d_{IN}, the width of conductor segment is *w*, the spacing between adjacent conductor segment is *s*. All of these parameters are output result of GP optimization algorithm. In such a case the first part of the equation (4) represents the previous length of the segment, while the second part of the equation is a step of enlargement of the part of the inductor. We must take care that we always have the two equal segments, except at the start, and after enlargement of segments. This means that the segment l_4 is equal to l_5 , ($l_4 = l_5$), while segments l_6 and l_7 , ($l_6 = l_7$) are enlarged for *w*+*s*.

Layout of Octagonal Spiral Inductors

Let us fit the wire's width to suit the width of our structure, (for example *w*=10.4 μm). The angle that the wire can move should be adjusted to be 45º. In the course of drawing when we achieve a desired length of the segment by click of the mouse we pass to drawing of the next segment. Here also should be taken care of prolonging of the segments in order to get corresponding distance between segments of the inductor.

Prolongation of segments of the wire is being done on the basis of the following formula:

$$l_1 = \frac{d_{IN}}{1+\sqrt{2}} \tag{5}$$

$$l_2 = \frac{d_{IN} + (w+s) + w}{1+\sqrt{2}} \tag{6}$$

$$Step = l_2 - l_1 \tag{7}$$

In all that l_1 and l_2 form the overall circle of the octagon, i.e. the first four segments of length l_1 and the following four ones are of length l_2. The equation 7 designates the step between the two lengths, so that l_3, and by that the four following segments have length $l_3 = l_2 + Step$, and so on, as can be seen in Figure 4. Layouts of all optimally designed spiral inductors are depicted in Figure 5.

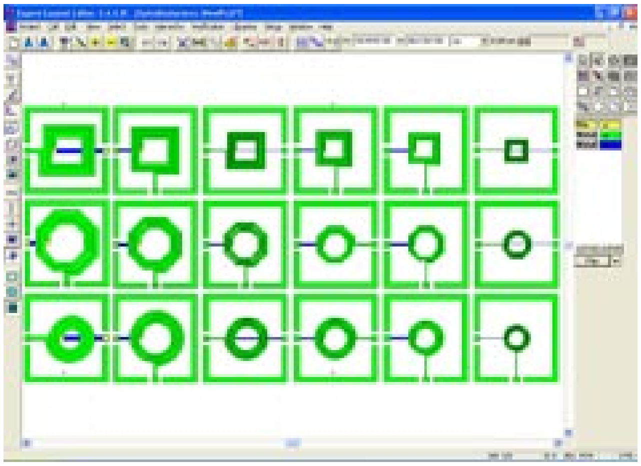


Figure 5. Layout of optimal spiral inductors in ***Expert*** Layout Editor.

Layout of Circular Spiral Inductors

The layout of the circular inductor is drawn slightly different from drawing of the octagonal and square inductors. The principle we use is to draw first the circumference of corresponding radius and to cut off that half that we do not need. This way we enlarge diameter to each semi-circle. The equations used to draw the structure are:

$$r_1 = \frac{d_{IN}}{2} \qquad (8)$$

$$r_2 = r_1 + w \qquad (9)$$

$$Step = \frac{w + s}{2} \qquad (10)$$

We select **Edit > Set Origin**, i.e. we place the coordinate system there where we draw the structure. Select **Edit > Create Object > Circle** (icon)

○

and draw first the circle with greater radius. Then, select **View > Modify object** and **Edit > Numeric input** and the window will open data for X and Y axis, X=Y=0 and data for radius, value for r_2 is entered. After that select **Edit > Region Mode > Hole** (third icon)

first designates the normal mod of drawing, second icon designates the merge mod, third icon designates hole mod, while the fourth icon also designates hole selected mod). Then, select **Edit > Create Object > Circle** and draw the circle of smaller radius. For values of coordinates X and Y we write down X=Y=0, while for value of radius r_1. Here we actually deducted from the greater circle the smaller one and we obtained as the difference the surface between these two circles needed for drawing of the circular inductor. By selecting **Edit > Create Object > Box** we access to removing of the negative part (lower half) of our circle in order to get the upper part, which we need, for drawing of the structure.

The equation (10) represents a step between radiuses of different segments needed for drawing of the circular structure. When drawing this structure, user needs to pay attention as to which part of the circle is to be removed, especially when linking segments. The segment that is drawn should be moved slightly vertically or horizontally by selecting **Edit > Move**. For fine displacement select **Edit > Move** (icon)

and **Edit > Numeric input** on the basis of which we can input arbitrary values for displacement of the structure along X and Y axis, so that we move the segment to the needed place with suitable accuracy. Layouts of designed optimal circular inductors are depicted in Figure 5.

Conclusion

In this paper we present design of different layouts (square, octagonal or circular) using ***Expert*** tool. All inductors are optimized to have maximum quality fator (Q-factor) with constrain for minimal spacing between adjacent conductor segments and for fixed inductance at operating frequency (for our examples inductances are 4nH or 10nH at operating frequency 2.5GHz). ***Expert*** can create all angle polygons for RF and microwave circuit elements, including different geometry of spiral inductors in many process technologies.

Reference:

[1] M. Hershenson, S. S. Mohan, S. P. Boyd, and T. H. Lee: "Optimization of inductor circuits via geometric programming", in Proc. Design Automation Conf., session 54.3, June 1999, pp. 994-998.

[2] Goran Stojanovic, Lj. D. Zivanov, M. Damnjanovic: "Improved method for optimisation of inductors using geometric programming", Electronics, Banjaluka, vol. 7, no. 1, september 2003. godine, pp. 31-34.

[3] G. M. Stojanovic, Lj. D. Zivanov: "Comparison of optimal design of different spiral inductors", 24th International conference on microelectronics (MIEL 2004), 17-19. May 2004, Niš, Serbia and Montenegro.

[4] Goran Stojanovic, Ljiljana Zivanov and Mirjana Damnjanovic: "Optimal design of octagonal inductors", Buletinul Stiintific al Universitatii "Politehnica" din Timisoara, Romania, Seria automatica si calculatoare, periodica politechnica, Transactions on automatic control and computer science, Vol. 49 (63), pp. 219-222, 2004, ISSN 1224-600X.

Resistance Calculation Approach in Hipex-NET

Hipex-NET uses two techniques to calculate resistor values. First method of resistance extraction, which is usual for full chip netlist extractors and uses heuristic algorithms to recognize common shapes for which were obtained empirical formulas depending on geometry of resistor body and resistor terminals. Unfortunately, this method doesn't cover the wide range of resistors and can handle only rectangular resistors, L-bends, and T-shaped resistive fragments. It also can be used to calculate resistance of snake and dog-bone shapes. We should also note that comparing to ***Maverick***, ***Hipex-NET*** recognizes some new shapes as routine shapes for which resistance value can be calculated by the well-known formulas [1].

If this method cannot be applied to a specific resistor then the tool uses the general technique.

The general technique is capable of finding the resistance between any set of arbitrarily shaped boundaries through any shape resistive region, see figure 1. This well-known technique solves Laplace's equation,

$$\nabla^2 \varphi = 0,$$

over the resistive region Ω. The potential φ is a function of (x, y). Boundary Γ consists of two kinds of boundaries, Γ_φ and Γ_q. Γ_φ is the forced boundary and potential on this boundary is constant and the current flows across the boundary. This condition can be written as

$$\varphi = \varphi \text{ on } \Gamma_\varphi$$

The other Γ_q is free boundary, , and the current along the perpendicular direction to this boundary is zero. This condition can be written as

$$q = \frac{\partial_\varphi}{\partial_n} = q_0 \text{ on } \Gamma_q$$

Hipex-NET can extract a full equivalent resistance network from the shape of which the terminals are connected to P different electrical nodes. Usually the typical value of P is two, but the algorithm implemented in ***Hipex*** implies no restrictions on P. Resistance between any ports and can be easily found if bias voltages are chosen as follows:

$$V_j = 1 \text{ and } V_j = 0,\ k = 1,\dots,\ j - 1,\ j + 1,\dots,P.$$

Then,

$$R_{kj} = \frac{\Delta V_{kj}}{I_k} = \frac{1}{I_k}$$

and the current can be evaluated by the line integral

$$I_k = \int_{\Gamma\varphi} \frac{\partial_\varphi}{\partial_n} d\Gamma.$$

In ***Hipex-NET***, the boundary element method (BEM) is employed to solve the above Laplace's equation. The BEM transfers partial differential equation in a domain into a set of integral equations on the boundary. The discretized integral equations are the only equations solved. This method yields a much smaller linear algebraic equation systems comparing to finite difference method (FDM) and finite element method (FEM). In addition, mesh generation in the BEM is not that expensive. We should note that the similar approach was implemented in the old Silvaco's tool ***Maverick*** but the algorithms used in ***Hipex-NET*** to generate boundary meshes and auxiliary matrices are faster and more accurate. They also reduce the memory space drastically.

References

[1] Kuang-Wei Chiang, "Resistance Extraction and Resistance Calculation in GOALIE2", Proceeding 26-th Design Automation Conference, pp. 682–685, June 1989.

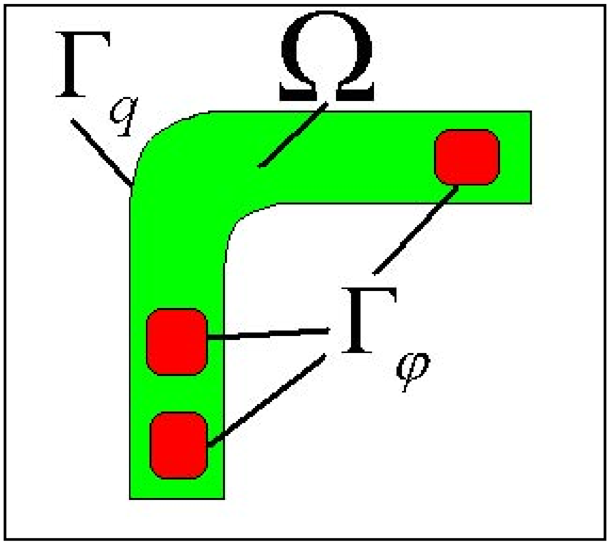


Figure 1. Example of resistor.

Calendar of Events

March

1 DVCON - San Jose, CA
2 DVCON - San Jose, CA
3 DVCON - San Jose, CA
4 DVCON - San Jose, CA
5
6
7
8
9
10
11
12
13
14
15 GOMACTech - Monterey, CA
16 GOMACTech - Monterey, CA
17 GOMACTech - Monterey, CA
18 GOMACTech - Monterey, CA
19
20
21
22
23
24
25
26
27
28
29
30
31

April

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Bulletin Board

Verilog-A Active Device Models

Silvaco released seven key Verilog-A Active Device Models are available on its website for free download under open source distribution. Complete source code of BSIM3, BSIM4, EKV, and LEVEL3 for CMOS, Gummel-Poon, Mextram, and VBIC for bipolar, RPI-TFT for TFT, and Diode are compliant to Verilog-AMS 2.1 from Accellera, derived from IEEE 1364-2001 Verilog HDL specification.

FSA PDK Checklist

Seeking to make life easier for analog, mixed-signal and RF designers, the Fabless Semiconductor Association (FSA) has approved a standardized checklist that describes the content of process design kits (PDKs). Ken Brock, PDK working group chair and vice president of marketing at Silvaco, described the checklist as "a combination of an ingredients list and a nutrition facts panel." Companies represented in the Committee's working group include 1st Silicon, Agilent Technologies, austriamicrosystems, Cadence Design Systems, HPL, Jazz Semiconductor, Mindspeed Technologies, OK Initiative, PolarFab, Silvaco and TSMC

If you would like more information or to register for one of our our workshops, please check our web site at ***http://www.silvaco.com***

The Simulation Standard, circulation 18,000 Vol. 14, No. 4, March 2004 is copyrighted by Silvaco International. If you, or someone you know wants a subscription to this free publication, please call (408) 567-1000 (USA), (44) (1483) 401-800 (UK), (81)(45) 820-3000 (Japan), or your nearest Silvaco distributor.

Hints, Tips and Solutions

Galina Makovsky, Applications and Support Engineer

Q: When I do an extract, the generated layers are present in the layout. If I need to do layout modifications I find it very cumbersome to deal with the generated layers. Is there a way to turn-off all of the generated layers so that layout modifications can be done without them cluttering things?

A: Generated layers that appear in layout after extraction are either Derived or Scratch layers. Tools >> Derived Layers >> Clean Derived Layers in Whole Project or in Cell Hierarchy deletes all objects from derived layers. If some modifications will be done to layout, derived layers should be rebuild anyway. If extraction technology uses Scratch layers, use Tools >>Scratch Layers >> Clean Scratch Layers to delete objects from generated layers. New layer plan, including drawing layers only, can be created to hide all generated layers at once.

Layer plans are used to set the following properties for each layer:

- Individual filling settings
- Visible/Invisible in the layout
- Selectable/Nonselectable
- Included/Not included into the layer list of the Layer bar

If you click on the 'Plan' button in the Layer bar, the Layer Plan panel appears. Layer Plan dialog can also be opened from View >> Layer View >> Layer Plan >> Setup. Each particular combination of visibility/selectability/inclusion may be assigned a name and added to a drop-down list of Layer Plans. This makes it easy to switch between different layer plans.

The "Include" column contain check marks indicating that the corresponding layer will be included in the particular layer plan. Layers may be checked on/off in three ways:

- You may click on a button with a plus (+) or a minus (-) sign to check On/Off all layers simultaneously;
- You may click on the button under the required column to check On/Off the highlighted layer;
- You may click the right mouse button directly on the layer name in the layer list column to set On/Off all properties of any layer.

The Layer Plan panel contains a Plan List option. You may add, delete, or rename plans in it. You may select any plan from the list to be used in the Layer Bar. This list is available as a drop down list at the Layer Bar.

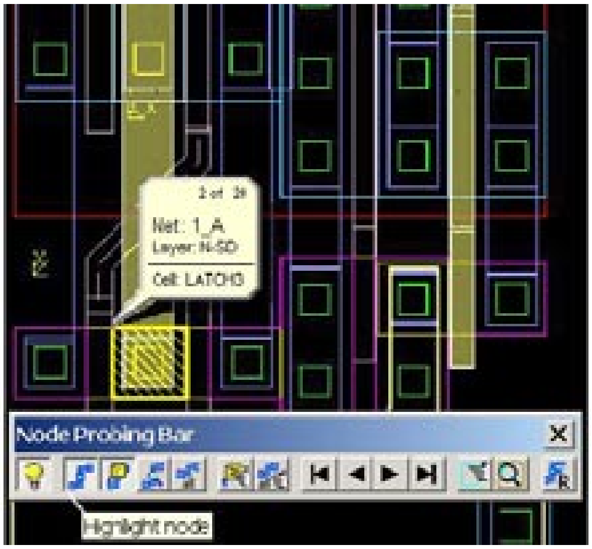


Figure 1. Highlighting Net in Expert .

If new plan, including drawing layers only is added, you can switch between layer plans 'FULL', 'DATA' and 'Drawing_Only' easily.

Layer plans are stored in ***Expert*** <*.eld> project files. If you want to reuse layer plans in other projects, you may save them into <*.elp> files and later load them using 'Save' and 'Load' buttons in the Layer Plan panel.

Q: Is there a way to turn-off markers in the layout resulting from LVS Navigator and/or node probing? I really like these features, but I find them cumbersome to work around after I find the issues and wish to address them!

A: The button "Highlight Node" next to "Probe Node" button in the Node Probing Bar switches highlight for active node objects On/Off (Figure. 1). Another way to turn-off node probing markers is to click in empty area of the layout when in a node probing mode (light bulb cursor) to clear active node data.

Call for Questions

If you have hints, tips, solutions or questions to contribute,
please contact our Applications and Support Department
Phone: (408) 567-1000 Fax: (408) 496-6080
e-mail: support@silvaco.com

Hints, Tips and Solutions Archive

Check our our Web Page to see more details of this example
plus an archive of previous Hints, Tips, and Solutions
www.silvaco.com

Your Investment is Safe

20 Years and Growing

Financially Rock-Solid

Fiercely Independent

We are NOT For Sale

SILVACO
INTERNATIONAL

USA Headquarters:

Silvaco International
4701 Patrick Henry Drive, Bldg. 2
Santa Clara, CA 95054   USA

Phone: 408-567-1000
Fax: 408-496-6080

sales@silvaco.com
www.silvaco.com

Contacts:

Silvaco Japan
jpsales@silvaco.com

Silvaco Korea
krsales@silvaco.com

Silvaco Taiwan
twsales@silvaco.com

Silvaco Singapore
sgsales@silvaco.com

Silvaco UK
uksales@silvaco.com

Silvaco France
frsales@silvaco.com

Silvaco Germany
desales@silvaco.com

E*ECAD
PARTNER

*Products Licensed through Silvaco or e*ECAD*