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

Volume 14, Number 3, March 2004

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**".

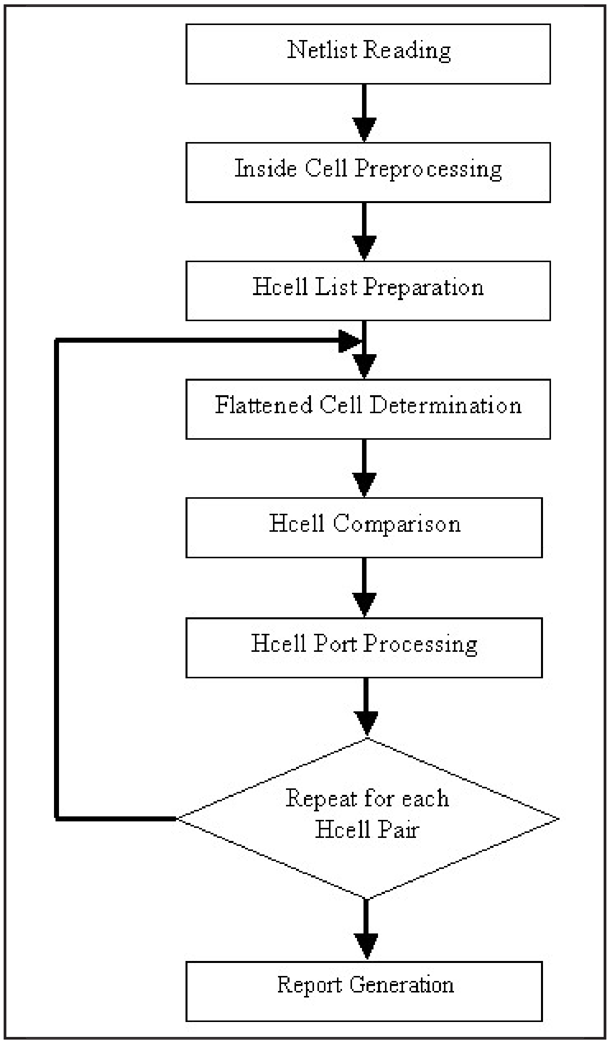

Figure 1. Operation Flow for Hierarchical Verification.

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

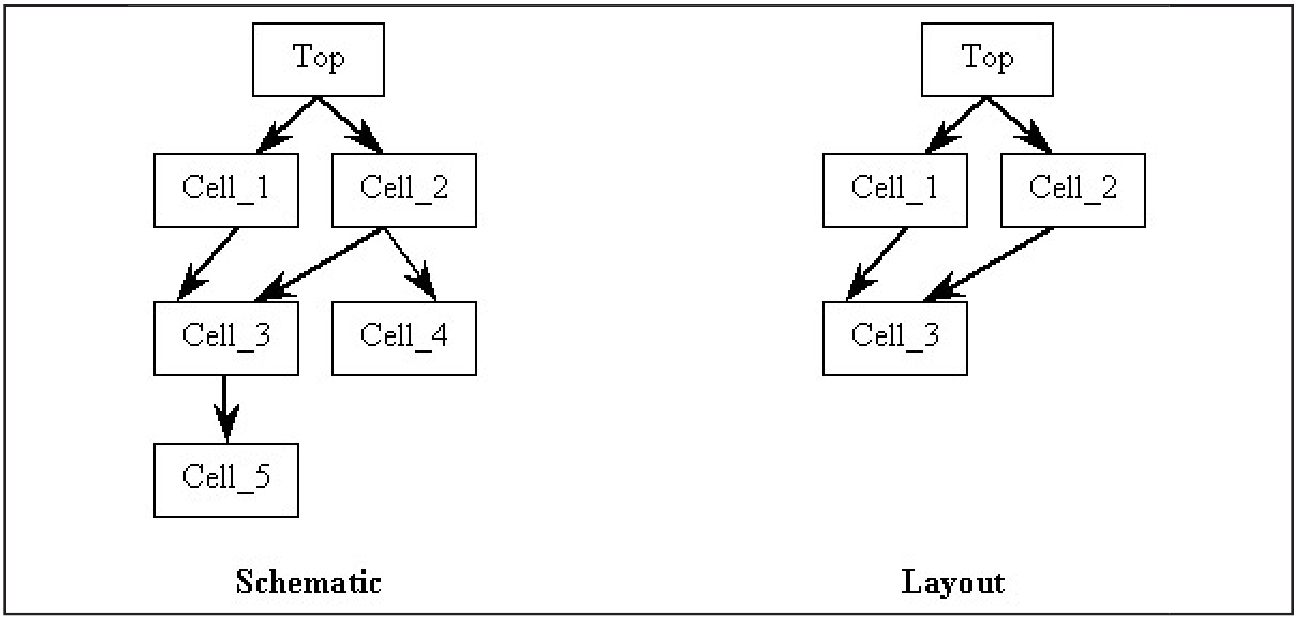

Figure 2. The hierarchies of schematic and layout

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).

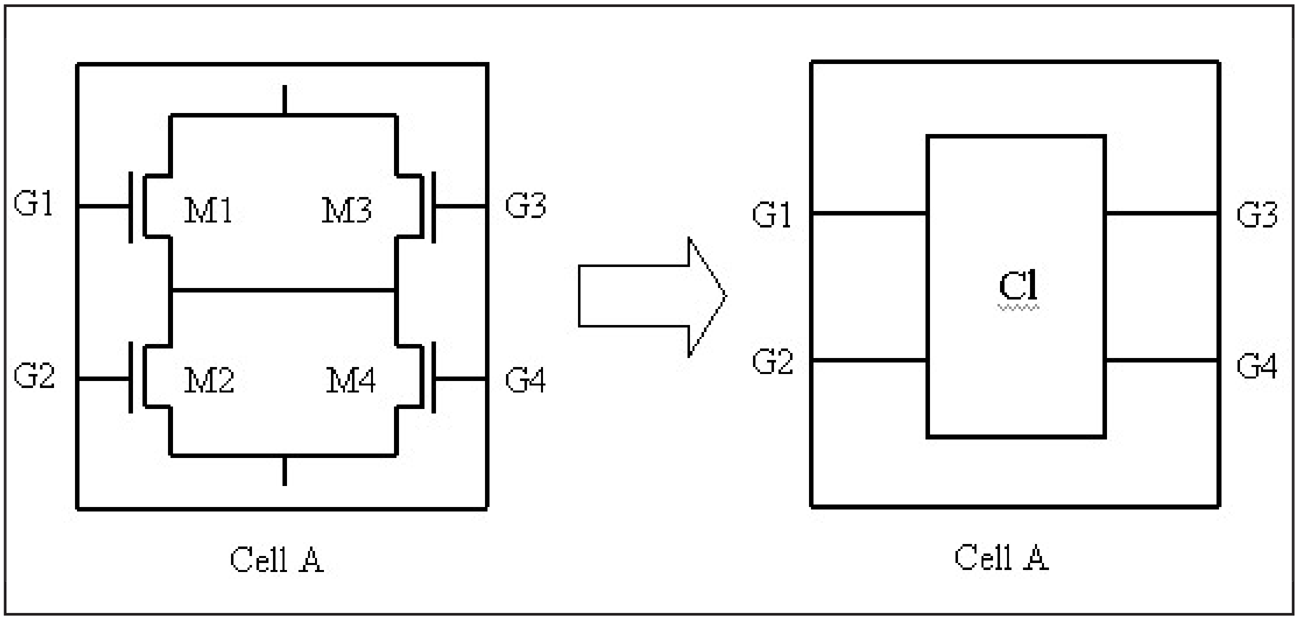

Figure 3. Port swappability for logical configurations.

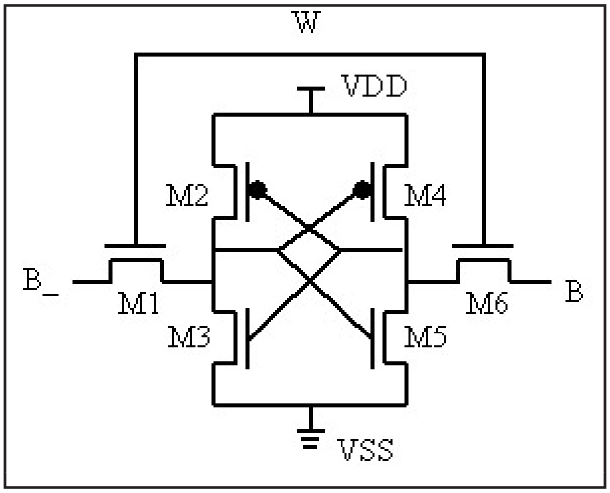

Figure 4. Six-transistor SRAM.

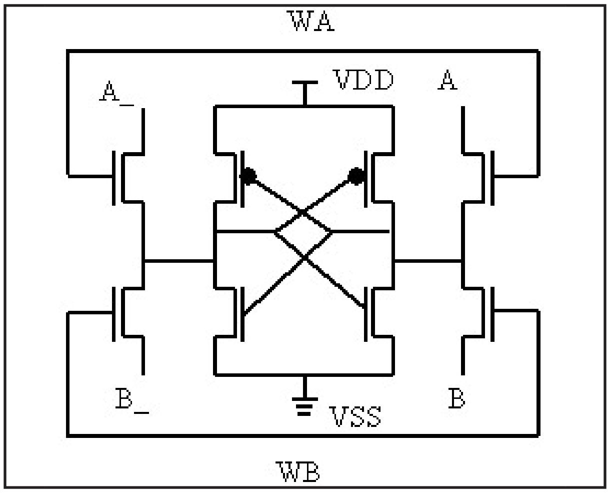

Figure 5. Eight-transistor SRAM.

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.

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.

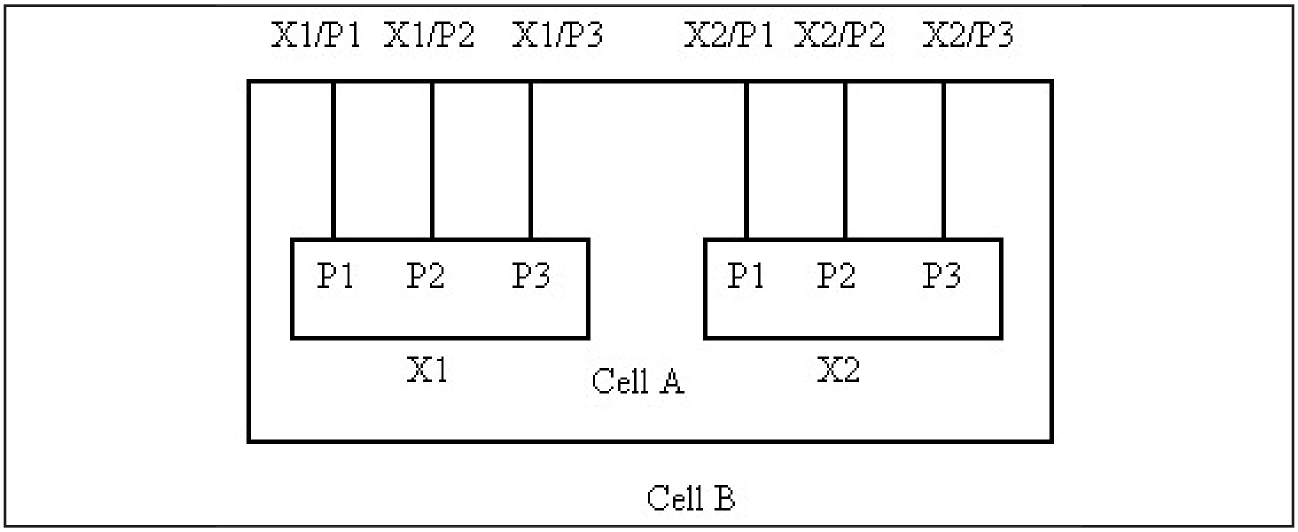

Figure 6. Hcell level port swappability.