

# Guardian LVS: New Platform-Independent GUI

## Introduction

We introduce platform-independent version of *Guardian LVS*. At present, very important feature of any tool is its capability to operate on various platforms and to support the internationalization. Now *Guardian LVS* is a multiplatform application operating on the following platforms: Windows, Linux, Solaris (32- and 64-bit versions), HP-UX. *Guardian* User Interface is realized on the base of Trolltech's Qt that is a C++ toolkit used by numerous companies and organizations for development of multiplatform GUI and applications.

## Guardian User Interface Using Qt

*Guardian LVS* is equipped with Silvaco Text Editor which is also based on Trolltech's Qt toolkit. This Text Editor has advanced memory management that allows you to operate with files up to 2Gb. The system optimally uses RAM by "dynamic loading-unloading" the contents of file. The Text Editor has all the basic features, such as:

- File features for creating, opening, and saving text files.
- Printing features with possibilities of choosing printer, specification of size, source, orientation of the paper, choosing color or grayscale mode for printing, setup of page margins, print preview.
- Editing features with multi-level undo and redo; cut, copy, paste, delete and select all support.
- Search features that allows to find, find next, find previous, replace a given text string, quickly go to a

given line number, to set the bookmarks and travel from a bookmark to another one in the forward and backward directions.

Moreover, *Guardian* User Interface has the features for management of LVS verification process:

- Project file features for loading and saving LVS project files.
- Action features for performing LVS verification, flattening hierarchical netlists, viewing hierarchical structure of netlist, loading "LVS Navigator" tool after the verification is done.
- Window management features for arrangement of netlists and reports. *Guardian LVS* allows you to view simultaneously all LVS reports as well as to link these reports with the schematic and layout netlists for easy inspection of LVS results.
- Setup features for setting options of LVS verification, some options of user interface, font setup.
- *Guardian LVS* on-line help system that uses Qt text browser for hypertext navigation.

*Guardian LVS* also performs the spice syntax highlighting for schematic and layout netlists. It can recognize names of devices and subcircuit instances, spice commands, parameter keywords, and comments (see Figure 1).

Frequently used commands are assigned to buttons of four toolbars: main, action, window and edit. You can move these toolbars and show or hide any one from them.

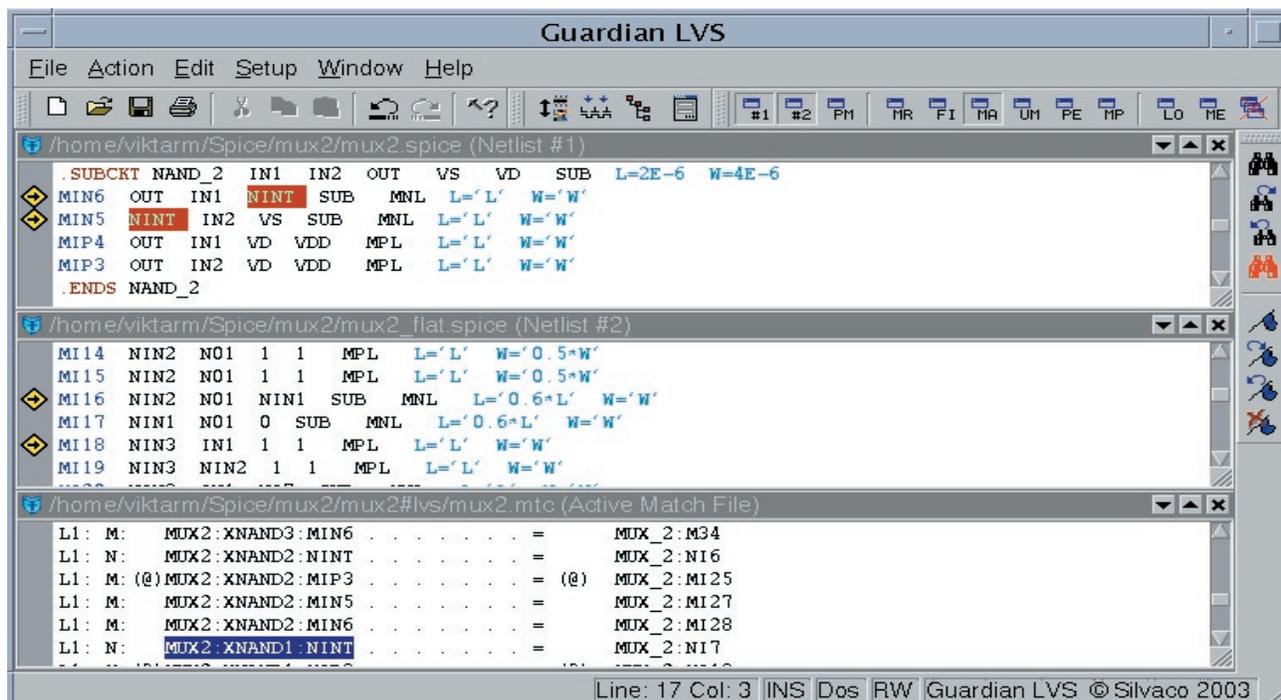


Figure 1. Spice syntax highlighting in platform-independent Guardian LVS.

## Inspection tools in platform-independent *Guardian LVS*

After a LVS run has been performed, *Guardian LVS* creates the reports that are used for inspection of LVS results. Platform-independent *Guardian LVS* is equipped with a set of tools that link the report files with the schematic and layout netlists and make the inspection of results maximum easy. *Guardian LVS* has the following inspection tools:

- Netlist Hierarchy tool activates the Netlist Rover that contains the full information about netlist hierarchy represented in the form of the tree (see Figure 2). You can see in the tree all instances belonging to a subcircuit as well as the names of subcircuits corresponding to these instances. You can highlight the instance or subcircuit in netlist by double click on its name in the tree of Netlist Rover.
- Connectivity Traversing tool shows all nodes to which selected node is connected. Double click on any of the neighboring nodes opens the same panel with connections for new node.
- Netlist File Highlighting tool allows you to find a node in netlist after double click on its name in a report. The corresponding spice file will be open and all appearances of this node name will be highlighted at the top level or in a subcircuit (see Figure 1).
- LVS Navigator tool searches and inspects matched and unmatched nodes, discrepancies and parameter errors. These nodes will be highlighted in reports and both netlists (see Figure 2).
- Node Walker tool shows you the neighborhood list of a node and match or unmatched information for these neighborhoods in the Node Walker panel. Double

click on a node name in this panel activates neighborhood list of this node (see Figure 2).

- Merge Hierarchy tool displays the information about merge and reduction for a selected hierarchical name in the merge, match, unmatched, parameter error, and parameter match reports (see Figure 2).
- Instance-Subcircuit Info tool shows the hierarchy of a selected node name in filter, match, unmatched, parameter error or parameter match reports. You can highlight in netlist any instance and the corresponding subcircuit which contains a selected node by double click in Instance-Subcircuit panel.

*Guardian LVS* can process huge netlists containing millions of devices. These netlists have the nets connecting to hundreds of thousands neighbors. Some from above-listed tools, for example Connectivity Traversing, Node Walker, must contain and represent the information about all connections of these nets. Moreover, the tree in Netlist Rover panel must keep all names of instances, subcircuits of these instances, and devices of the netlists. It's necessary to note that the representation of information in dialog panels in *Guardian LVS* on the base Trolltech's Qt takes much less time than *Guardian LVS* on the base MFC. For example, representation of the tree of Netlist Rover for netlist with 120000 devices in *Guardian LVS* with Qt is performed about 50 times faster than in *Guardian LVS* with MFC.

## Conclusion

At present *Guardian LVS* is a platform-independent application. It operates on Windows, Linux, Solaris, HP-UX and can be built for many other Unix-variant platforms.

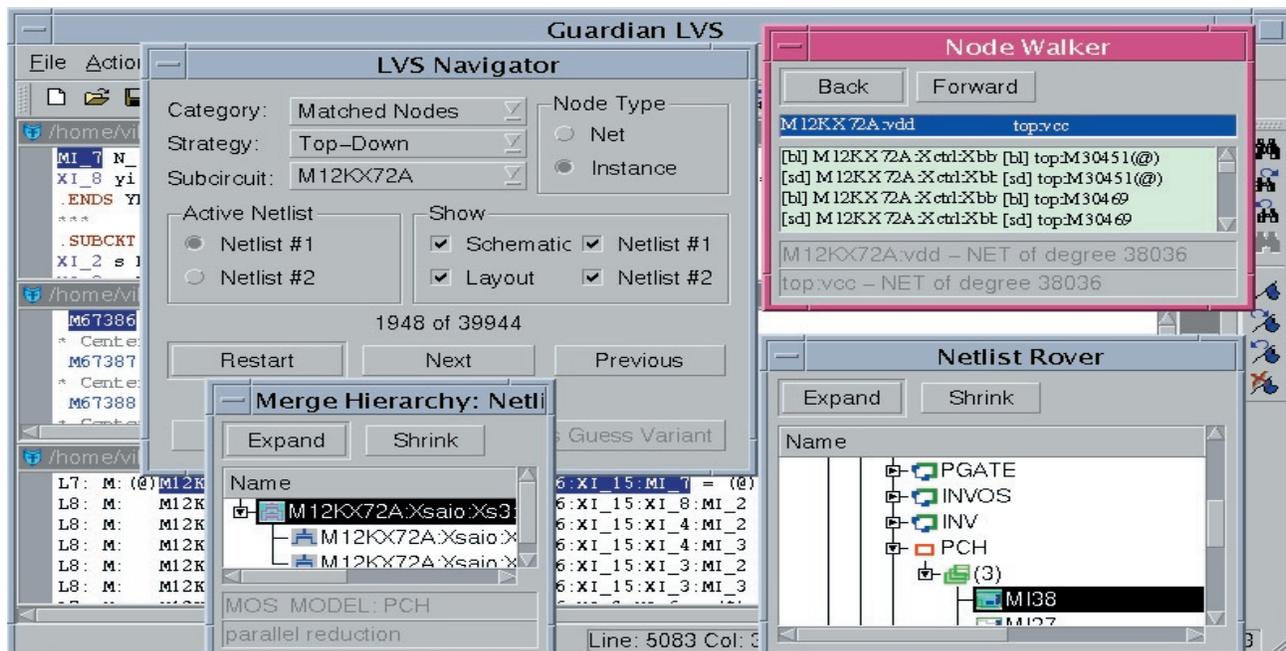


Figure 2. Inspection tools in Guardian LVS (Linux example).