Simulation Standard

TCAD Driven CAD

A Journal for CAD/CAE Engineers

An Intuitive Front-End to Effective and Efficient Schematic Capture Design

Introduction

In our previous publication ("***Scholar***: An Enhanced Multi-Platform Schematic Capture", *Simulation Standard*, Vol.10, Number 9, September 1999) we presented the main features of a new-generation schematic capture tool – Silvaco's ***Scholar***. In this article we are going to describe and analyze the graphical user interface (GUI) of ***Scholar***.

The GUI is a very important part of any software tool, especially if this tool is for making some interactive manipulations. That is the case for any schematic editor. ***Scholar*** provides the user with an intuitive, highly effective GUI that makes the schematic design process a very easy and comfortable task (Figure 1 demonstrates a typical screen while using ***Scholar***).

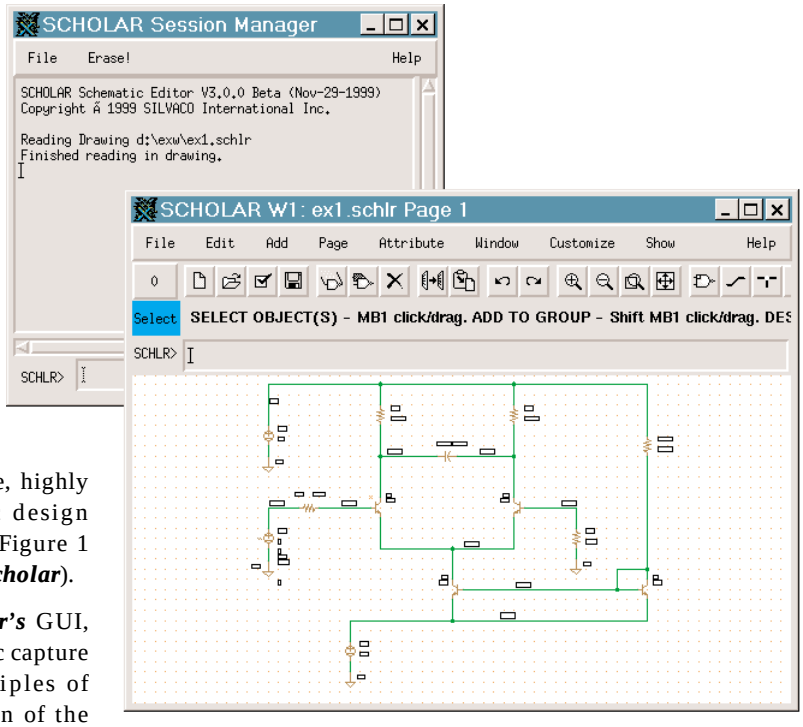

Figure 1. Typical screen of ***Scholar*** Schematic Capture Editor.

Before dipping into the details of ***Scholar's*** GUI, some basic concepts of ***Scholar*** as a schematic capture tool will be presented. Then, the principles of ***Scholar's*** GUI, followed by the description of the main elements of the GUI are described. The user interface of operations for both creating and editing the schematic elements is analyzed in two separate sections. Finally, the aspects of Scholar's GUI customization are considered.

Concepts of Scholar Schematic Capture

Scholar is a multi-platform schematic capture editor. A single schematic entity in ***Scholar*** is called a drawing. ***Scholar*** supports the creating of a hierarchical schematic with unlimited depth. The drawing that contains schematics of the given hierarchical level, is called a schematic drawing. The drawing which contains a graphical picture that is used for instantiating of the given design on the upper levels of hierarchy, is called a symbol drawing.

Schematic drawings contain wires, macroboxes, arcs, junction dots and symbol instances. Each element of schematics may be assigned attributes. Attributes are some kind of properties: the attribute has a name and a value. ***Scholar*** provides for flexible and complete functionality on managing attributes.

Continued on page 2....

INSIDE

SILVACO INTERNATIONAL

The large and complex schematic drawings may be organized as a set of pages. Each page contains a part of the whole schematics.

Upon completion of the schematic design, ***Scholar*** generates a SPICE netlist that is used for simulation purposes. There is a comprehensive set of electrical rules that a drawing is checked against before the final netlist is written out.

Scholar Windows

Scholar is a multi-window application: the information about schematics is presented in several logically related windows. These are ***Scholar*** Session Manager window, drawing (or graphic) windows, magnifying glass window and dialog boxes.

Session Manager Window

The Session Manager window is the very first window that is presented to the user upon ***Scholar's*** start up. (see Figure 1 for the example of the ***Scholar*** Session Manager window).

The session manager window consists of three parts: the menu bar, the messages area and the command line field. The message area displays all the output from ***Scholar*** that is a result of commands' execution. Although the capacity of the messages area is not limited, the Erase! item on the menu bar provides for an immediate clearing of the messages area. This command is useful if the user wants to get a clear picture of the messages as a result of some complex operation (for example, checking a schematic drawing against the electrical rules).

The command line field allows the user to enter ***Scholar*** commands using a keyboard. ***Scholar***, like some other tools from Silvaco, supports the command language ***LISA***(Language for Interfacing Silvaco Applications). Most of ***Scholar's*** functionality on working with schematics that available through GUI is also available in the form of ***LISA*** commands. Being a sophisticated and powerful command language, ***LISA*** creates and executes complex command scripts, thus providing the user with ability of writing their own design procedures.

The Session Manager window is on the screen during the ***Scholar*** editing session. Closing the Session Manager window exits application.

Drawing (Graphic) Windows

Schematics of the drawings are displayed on the screen in the drawing (graphic) windows. The drawing windows are those with which the user spends the most of the session time. A typical drawing window is presented on Figure 1.

It is possible to have one drawing displayed in many windows, as well as several drawings (each in several windows) opened at a time. Different windows of the drawing may display the same or different pages of the drawing.

The graphic area of the drawing window is the place where the schematics are displayed. User interaction with ***Scholar*** is primarily done through the graphic area of the drawing windows.

The upper part of the drawing window is occupied by three bars: the function bar (which can also be called the toolbar), the hint line bar and the command line bar.

The function bar contains the selected objects button, the row of command buttons and the group of controls for navigating the drawing pages. The text on the selected objects buttons represents the number of currently selected objects. Clicking on the button results in a summary about the selected objects to be printed in the message area of the Session Manager window. If there are some objects selected, the background color of the button changes to reflect the selection.

The row of buttons to the right provides quick and easy access to the most frequently used commands of ***Scholar***. These are the commands for opening and saving drawings, creating different types of objects, editing operations (move, copy, delete) and so on.

The hint line bar contains the command name indicator and the hint line itself. The command name indicator always displays the name of the currently active operation (command). For example, if the user selects the Move command, the command name indicator displays "Move" as its text, while the hint line displays mouse button operations. If there are several windows that display the same drawing page and the user activates a command in one of them, that command becomes active in all drawing windows for that page and the contents of the hint line bar changes accordingly in all of those windows.

The command line bar contains the command line field, which is used for the same purpose as in the Session Manager window. However, there are some ***LISA*** commands that may be executed only in the context of the given drawing window. Such commands may be issued only from the command line field of that drawing window.

If for some reason the user does not need all of the bars to be present on the screen, they may be hidden in any combination (see below the section about ***Scholar's*** GUI customization).

Magnifying Glass Window

The magnifying glass window is a special pop up window that displays a magnified portion of the schematics currently under the cursor in the drawing window. Figure 2 demonstrates the drawing window before and after the magnifying glass window is displayed.

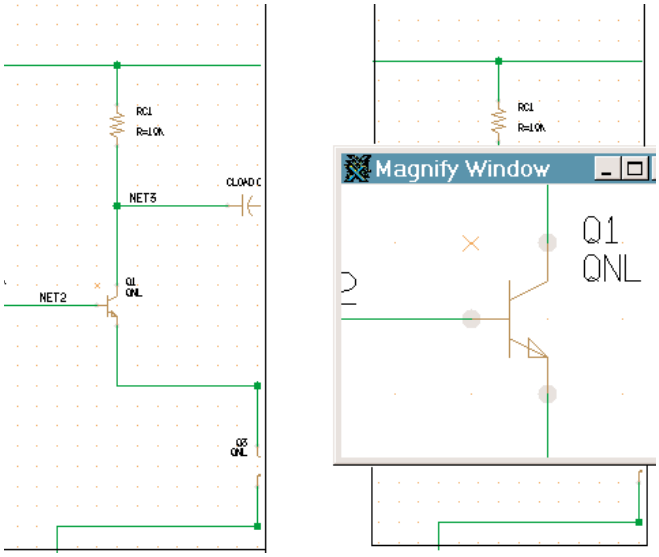

Figure 2. The example of magnifying glass window.

The magnifying glass window may be easily displayed as well as easily made to disappear. For both commands the user needs to press the middle button of the mouse while holding the Control key on the keyboard. The first button pressed pops the window up, while the next one pops it down. This functionality is available in all of ***Scholar's*** modes and allows avoiding the need in frequent zoom in and zoom out operations.

Dialog Boxes

The dialog boxes accept different parameters for ***Scholar*** commands and modes. Most of ***Scholar's*** dialog boxes are modeless which means that the user need not close the dialog box when he wants to switch back to the drawing window. The dialog box is simply displayed on the background thus providing for easy access and permanent display of its parameters (see Figure 3 with example of the grid customization dialog box on the background of the drawing window).

Another feature of ***Scholar's*** dialog boxes is that many of them are "direct action" dialog boxes. Direct action dialog boxes contain controls (lists, push-, checkbuttons, and so on) that immediately result in changes in the schematics. It means that if the user changed some controls to modify the data, the changes are applied to data immediately, without the need for the user to click OK or the Apply button.

Basics of the Scholar's GUI

Additionally, there are three main tasks that the user performs when working in any schematic editor: displaying schematics in various scales and views, creating new schematic elements like wires and symbol instances and modifying the schematics by moving or deleting elements and changing their attributes.

Before considering how each of these tasks can be done in ***Scholar***, let's see at some basic aspects and features of ***Scholar's*** GUI that need to be mentioned before we step further in to the details of the GUI. These aspects and features are common for the whole GUI of ***Scholar*** and provide for consistency of its interface. Similar actions are performed in a similar manner for all commands and modes. Such a consistent interface greatly simplifies both the learning and the usage of the tool.

Menus

There are two types of menus in ***Scholar***: pull down menus (also known as menu bars) and pop up menus (context specific menus).

The pull down menus of the drawing window menu bar provide for "tear-off" feature. The tear-off menu item is present in any pull down menu. When the user clicks on that item, the menu is "teared off" from the menu bar (Figure 4 demonstrates a teared-off menu), thus allowing it to be placed in any position around the drawing window. As a result, the items in the teared off menu are always visible and easily accessible for selection.

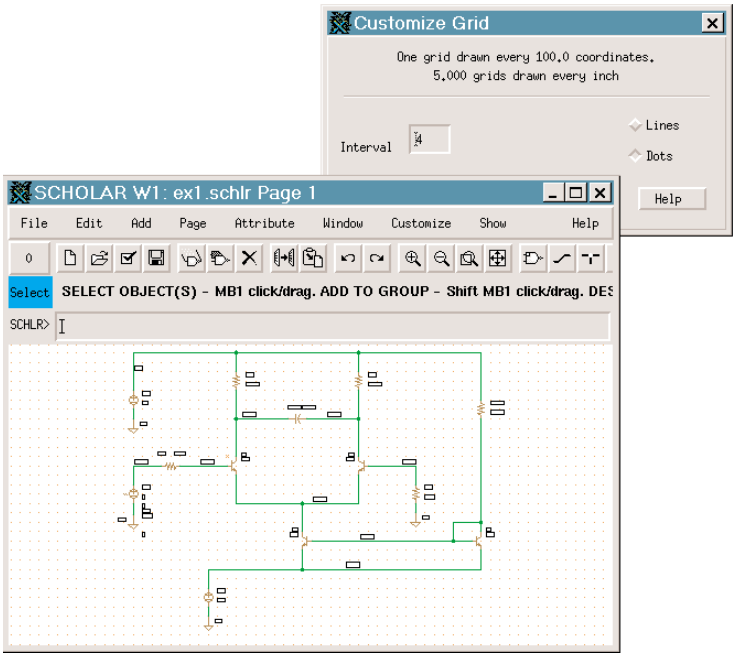

Figure 3. The example of modeless dialog box in Scholar.

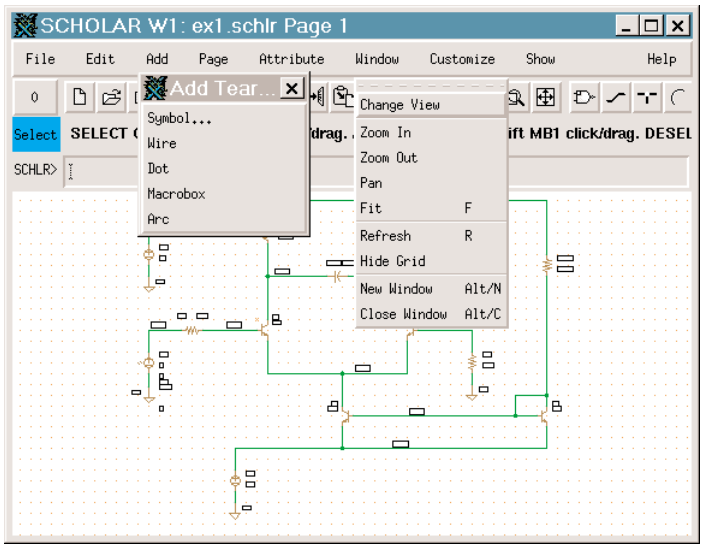

Figure 4. The example of teared off menu in ***Scholar***.

The pop up menus are displayed when the user clicks the right mouse button. The common pop up menu (system menu) is displayed regardless of the type of the object which the cursor is now over (Figure 5 shows the common pop up menu).

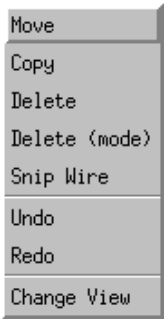

Figure 5. The common pop up menu.

The common pop up menu contains the commands that may be applied to any type of objects: Move, Copy, Delete, Undo, Redo. If, while clicking the right mouse button, the user holds down the Shift key, ***Scholar*** displays the object specific pop up menu. The contents of the menu depend on the type of the object for which the menu was invoked (Figure 6 shows the object specific menu for the wire object).

Figure 6. The wire specific pop up menu.

If there is no object under the cursor, but the Shift key is held down, ***Scholar*** displays a page specific menu which provides for placing some attributes, symbol instances and the creating of objects (see Figure 7).

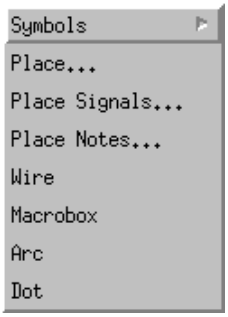

Figure 7. The page specific pop up menu.

The pop up menus in ***Scholar*** are customizable (see the section Customization of the Scholar's GUI).

Toolbar

The toolbar (the row of buttons in the function bar of the drawing window) contains the most frequently used commands. Having been placed in the toolbar, those commands are easily accessible all the time. When the cursor is over the button in the toolbar, ***Scholar*** displays a short description of the button's function. Figure 8 shows the ***Scholar's*** toolbar.

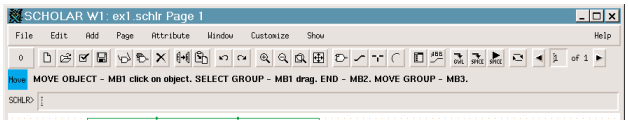

Figure 8. The ***Scholar's*** toolbar.

The behavior, look and feel of the ***Scholar's*** toolbar is similar to other modern applications.

Mouse Actions

The mouse is intensively used in the ***Scholar's*** GUI as a recognized input and pointing device for manipulating graphic objects. ***Scholar*** requires a three-button mouse for its normal operation. Pressing, dragging or clicking different mouse buttons results in different actions to be performed by ***Scholar***. The assignment of the actions by mouse buttons depends on the type of command, which is currently in effect. That assignment is displayed in the hint line. To make this descriptive text shorter, mouse buttons are denoted as MB1 for the left button, MB2 for the middle button and MB3 for the right button. If some modifier key (Control, Shift or Alt) should be held down while clicking on the mouse button, the modifier key name is placed before the button name, (for example: Shift MB1).

Although the assignment of the buttons' operations may differ from command to command, clicking on MB2 in any situation results in cancel of the previous operation or the mode as a whole. Clicking the MB3 or Shift MB3 usually displays the ***Scholar's*** pop up menus. Such a consistency in GUI greatly helps in working with the schematics in ***Scholar***.

Keyboard Shortcuts

The mouse is very important in doing the GUI related tasks that need precise pointing. There are also some actions that the user might want to activate quickly, but would find uncomfortable if they needed to move the cursor back and forth from the point of the interest (the "point of interest" is in our case the schematic picture in the drawing window). ***Scholar*** GUI provides for keyboard shortcuts in this case.

Almost any command that is activated from the menu or toolbar may be assigned to some combination of key and modifier (Control, Shift or Alt). This assignment is completely user-definable which means that every user may set up these shortcuts to reflect his own habits and needs.

Command Interface (LISA)

While creating schematics is mostly interactive work, involving point-and-click mouse actions, it may be necessary to automate the creation or modifying of some schematic elements. For example, one might want to set up the attribute values of some objects based on some parameterized arithmetic expression. ***Scholar*** provides capabilities for acting that way.

Almost all functionality of ***Scholar*** that is available through the menus and shortcuts is also accessible as commands of the ***LISA*** command language. Being a sophisticated and powerful command language, ***LISA*** allows to create and execute quite complex command scripts, thus providing the user with ability of writing his own design procedures.

Undo/Redo and Recovery Functionality

The important aspect of GUI is its ability to help the user in correcting their own mistakes, as well as in recovering from the situations that may be a result of hardware failure or third-party software bug. ***Scholar*** supports unlimited undo capabilities, which means that the user is able to roll back all the modifications he has made to the design from any point and up to the very beginning of the session. While undoing the design, the user may redo a just cancelled command from any point and up to the last change that he has made. These unlimited undo/redo capabilities work regardless of where the commands were invoked from – toolbar, menu, shortcut or ***LISA*** language script.

All the changes that ***Scholar*** applies to the data because of a user's actions are recorded in a special file, called a journal file. If some undesirable interruption of the working session occurs – hardware (power) failure, for example – the information stored in the journal files recovers all the results of the user's work up to the point of the interruption.

Manipulation of the Display in *Scholar*

Probably one of the most important aspects of any GUI is how easily a user can view the data that he is intending to edit. Before doing any editing operation, such as creating new elements or moving/copying the existent one, the user should have a clear picture on the screen of what they have in their data, how those different objects are arranged and so on. ***Scholar's*** GUI provides for comprehensive functionality on manipulation of schematic data display.

The commands responsible for manipulation of the data display are placed in the Window pull down menu of the drawing window menu bar. Figure 4 shows the contents of the Window pull down menu.

Zoom In and Zoom Out commands allow the scale of the displayed schematics to be changed. By dragging a mouse, the user may specify the rectangular area that will be used as a new "view" frame for the schematics. In the case of the Zoom In command, the specified rectangular area will be mapped to fill the window. The Zoom Out command results in the contrary operation – the current window "view" will be mapped to fill the specified rectangular area.

The Pan command changes the display of the schematics to shift the specified point to the center of the window.

The Fit command changes the display to make the whole schematic visible in the window.

The Refresh command allows the schematics to be redrawn. The Hide Grid command removes the visual grid from the screen. On the next invocation of the Window menu this command will read as "Show Grid".

The commands New Window and Close Window provide for the capability to have multiple windows opened for the same schematics. It may be very useful if the schematics are large and complex which is usually the case in modern designs.

The commands Zoom In, Zoom Out, Pan, Fit (as well as other commands in ***Scholar***) may be assigned to short-cuts. What is important here is that it is possible to specify the parameters for those assignments. For example, the shortcuts Alt+ and Alt- invoke the commands Zoom In and Zoom Out accordingly with parameters that result in changing the scale of the

Figure 9. The assignment of the mouse buttons in the Change View command.

schematics display by 25% up or down. Shortcuts assigned to arrow keys produce a pan operation for the window size in the correspondent direction.

The commands mentioned above are so called "one-time" commands. For example, if a user is entering a wire and needs to change the scale of the display or shift the picture, he may simply hit Alt+ (for Zoom In) or left arrow (for shift to _ of the window to the left) and then continue to enter the wire points. However, there may be some tasks that need browsing through the schematics in various directions. ***Scholar*** provides a very useful command for doing that – the Change View command.

The Change View command sets up the mouse button assignments to the actions that make browsing through the schematics easy. Figure 9 shows the assignment of mouse buttons in the Change View command.

Using these assignments, the user is able to jump quickly and easily to any portion of the schematics and then return back to the whole picture or just a previous view. The user need not select some items from the menu or toolbar, or move his eyes from the screen to hit some shortcuts; they are just doing work using ***Scholar*** through its intuitive and transparent GUI.

Functionality for Creating Schematic Elements

It was already mentioned above that schematic drawings in ***Scholar*** contain wires, macroboxes, arcs, junction dots and symbol instances. The commands responsible for the creation of these elements are placed in the Add pull down menu of the drawing window menu bar. Figure 4 shows the contents of the Add pull down menu.

Adding Symbols

Usually, the placement of the symbols is the initial stage of working with a new schematic drawing. Later on the user connects pins of the symbols with wires to create the connectivity. The Add Symbol... command brings up the dialog box that lists all available symbol libraries and their symbols (see Figure 10).

The upper list in the Add Symbol dialog box is the list of available symbol libraries. The other one is for symbols from the library, which is currently selected. Clicking on the desired symbol makes its image to appear on the screen at the cursor point. Now moving the cursor will move the symbol wherever the user wants. Clicking on the left mouse button places the symbol at the given point, while the dragged image does not disappear – the user may place as many symbol instances as he wants.

Figure 10. The Add Symbol dialog box.

If it is necessary to place another symbol, the user just selects it from the list and makes the placement again. The Add Symbol dialog box is modeless – it is always visible and does not require that the user close it before placing the symbol in the schematics.

If the user needs to change the orientation of the symbol being placed – the mouse button actions assignments are there to help. Shift MB2 will mirror the symbol, while MB3 will rotate it.

Adding Wires

The next step in creating new schematics – after some or all the symbols are placed – is to add some wires and connect the symbols' pins. The Add Wire command establishes new mouse button assignments that make entering wires a very intuitive process.

A simple click with MB1 begins new wire or marks the wire's second point. Shift MB1 begins the wire at the nearest suitable object. This feature is especially useful because the user need not point to the exact location; ***Scholar*** itself will choose the nearest pin or another wire and start the new wire at that point.

There are several modes for entering the new wire points. By default, Scholar creates two "L"-shaped wire segments for each one mouse click. Other modes for entering a wire are: non-orthogonal wire; only vertical segments; only horizontal segments; horizontal, then

vertical segment ("L"-A shaped); vertical, then horizontal segment ("L"-B shaped); "L"-A and "L"-B shaped segments that alternate; only horizontal or vertical segments alternate. The user can switch between these modes by clicking MB3, while Shift MB3 displays the pop up menu that presents all the choices at once (see Figure 11).

Figure 11. The wire types pop up menu.

Adding Dots

The two wires, that form a "T"-shaped intersection, are considered connected in ***Scholar***. ***Scholar*** can be customized to automatically place a dot at the point of "T"-shaped intersection (or can be set up not to do that – it is the user's choice and depends on his design style).

The two wires, that form a "+"-shaped intersection, are considered not connected in ***Scholar***. For these wires to be connected, the dot must be placed at the intersection point. Such dots can be added with the Add Dot command from the Add pull down menu. Each subsequent click on MB1 places a dot at that point.

Figure 12 shows the examples of "T"- and "+"-shaped intersections.

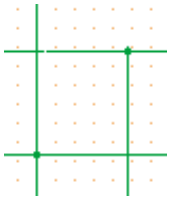

Figure 12. The examples of "T"- and "+"-shaped intersections.

Adding Arcs

Arcs are not used for connectivity purposes in ***Scholar***. Instead, they provide the ability to create some decorative elements in the drawing. That is especially useful when creating the schematic symbols. Arcs are added with the Add Arc command from the Add pull down menu.

There are different modes for entering arcs available in ***Scholar***. The user can choose the most suitable mode with the pop up menu. Figure 13 shows the arc modes pop up menu.

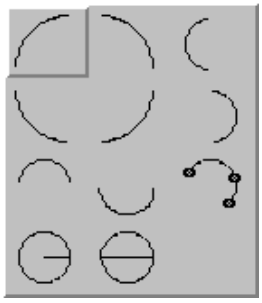

Figure 13. The arc modes pop up menu.

There are four modes that allow to define: the _ circle with two points, four modes that define _ circle with two points, two modes for entering the whole circle by specifying the radius or diameter, and the last mode provides for entering any kind of circle by defining its three points.

Functionality for Editing Schematic Elements

Features that allow the existent design to change and easily make any modifications are very important for schematic editors. Statistics shows that users spend only 20% of their time in creating a new schematics, while 80% of their time they change and modify their drawings. Let's consider how effectively main editing tasks are implemented in ***Scholar's*** GUI (most of them are activated from the Edit pull down menu).

Select Elements

Usually, some elements need to be selected before applying editing commands. Select mode is set up as a default mode in ***Scholar***. Clicking MB1 on an element selects that element while all the other ones get deselected (the selected elements are highlighted with special color). Clicking MB1 on an empty space of the drawing effectively deselects all the elements in that drawing.

Clicking Shift MB1 adds the object to those already selected. Control MB1 deselects the given object and does not affect all the other ones, while Control Shift MB1 toggles the select state of the object.

If the mouse is dragged, the select operation (select, deselect or toggle select) is applied to all the objects that are within or intersect with the drag area. Such a set of operations allows for quick and easy selection of any combination of schematic objects.

Editing Attributes

Attributes in ***Scholar*** are represented as text strings that are displayed near the elements they are attached to. Wire signal names, pin names, and symbol names are just some examples of these attributes.

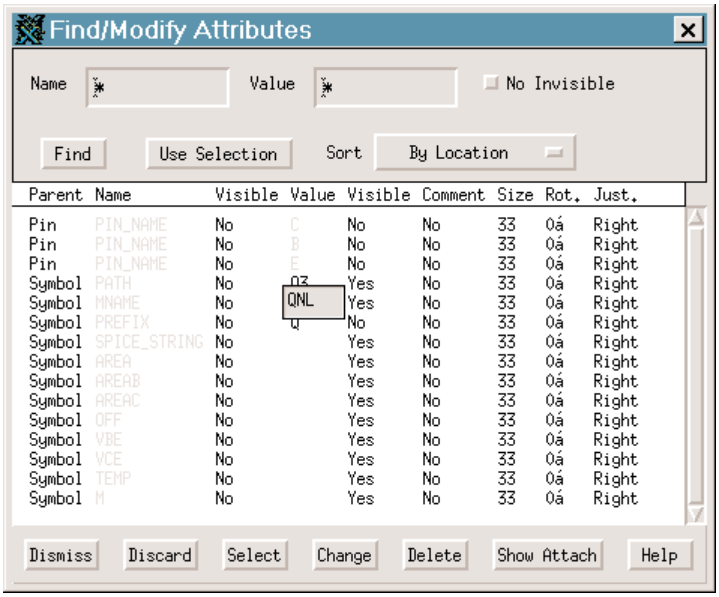

Figure 14. The Find/Modify Attributes dialog box.

Double-clicking on the element brings up the Find/Modify Attributes dialog box with all the attributes of the given element (see Figure 14).

Clicking on the value field of the attribute allows for direct editing of that value – the user can simply retype or change it. Double-clicking on the attribute value in the drawing brings up the Find/Modify Attributes dialog box just for that single attribute.

Such a direct and easy access to the attributes of elements creates a feeling of complete transparency of ***Scholar's*** GUI.

Move Elements

Probably the next most frequently used operation (after changing the attributes) is a move operation. The Move command from the Edit pull down menu activates the move mode in ***Scholar***. While in the move mode, MB1 on element initiates the moving of that element. The image of the element is moved as the user moves the cursor, so he can always see where the new location of the element may be. Clicking on MB3 will rotate the element, Shift MB2 will mirror it.

It is possible to move a collection of elements. The user selects elements as usual in select mode, or drags MB1 in move mode to select the elements to move, then clicks on MB3 to move the whole collection.

It is important to emphasize that ***Scholar*** moves elements while preserving their connectivity. Figure 15 shows the initial placement of the elements and how it looks while some of the elements are being moved around – the wires are modified to keep the connections intact.

However, it is also possible to move elements while breaking the connectivity (see the section on customization of the ***Scholar's*** GUI) – such a move simply "extracts" a part of schematics and repositions it to another location.

Copy Elements

The Copy command from the Edit pull down menu is just for duplicating the schematic elements. The mouse button assignments work similarly to those in the Move command: MB1 on an element initiates the copying of that element. The image of the elements is moved as the user moves the cursor, so the user can always see where the new copy of the element may be. It is possible to copy a collection of elements. The user selects elements as usual in select mode, or drags MB1 in copy mode, to select the elements for duplication, then clicks on MB3 to copy the whole collection.

Delete Elements

The Delete (mode) command from the Edit pull down menu switches ***Scholar*** into the mode for deleting the elements. The mouse button assignments work similar to those in the Move or Copy commands. MB1 on an element deletes that element, while MB3 deletes the currently selected group of elements. The latter assignment is also available as a separate command: the command Delete from the Edit pull down menu (as well as the Delete button in the toolbar) results in deleting of the currently selected elements.

Continued on page 11....

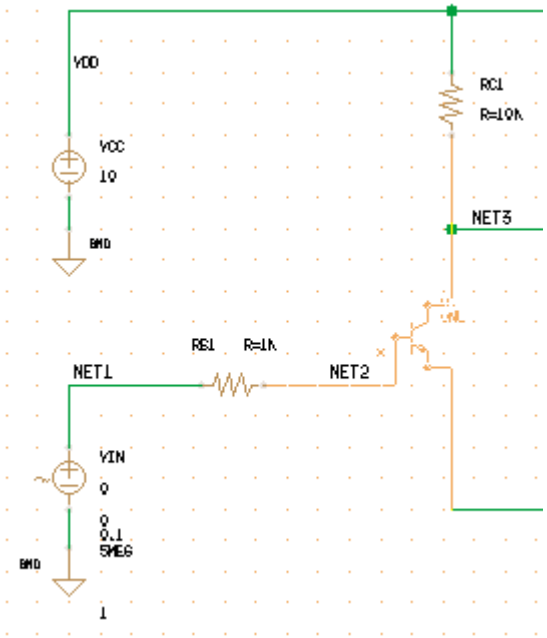

Figure 15. The example of move operation.

... continued from page 8

Copy/Paste Elements

While the Copy command allows for duplicating elements within a single drawing window, it may be necessary for the user to be able to copy some schematics between different windows of the same or even different drawings. The pair of commands “Copy to Buffer” and “Paste from Buffer”, located in the Edit pull down menu, simplify this task.

When the user selects the “Copy to Buffer” command, the currently selected elements are placed in ***Scholar's*** internal buffer for later use. Now the user may switch to another window, then select the “Paste from Buffer” command. This mode is quite similar to that of the Copy command. The image of the element to be copied is moved on the screen as the cursor moves. MB1 places a copy of the element at the given point, MB3 rotates the elements while Shift MB2 mirrors them.

Traversing a Schematic Hierarchy

In the bottom-up design flow, the user begins with creation of the schematics for the lower level blocks, then uses their symbol definitions to place the instances of those blocks in the upper level schematics. However, the design process is iterative, so the user may need to frequently go back to the lower level schematics or symbol definitions and change them.

In ***Scholar***, this traversing of hierarchy is just one click away. Shift MB3 on the symbol instance displays a pop up menu for that instance. Two of the commands in it, Edit Drawing and Edit Symbol, allow jumping to the schematics or symbol definition for that instance.

The Edit Drawing command opens a new window with the schematic drawing for the instance, while the Edit Symbol command opens a new window for the instances symbol definition. No need in browsing through the libraries and directories in search for schematics or symbol files – the ***Scholar's*** GUI helps the user get to the point with just one click!

Customization of the Scholar's GUI

Silvaco's ***Scholar*** was designed from the user's point of view. Its hallmark is it's customization based on different tasks, and varied levels of experience. Thus, ***Scholar*** is straightforward enough for the new user, yet has the power a veteran needs. The user will be able to customize the tool so that it interacts with the their level of experience, and the task at hand.

- colors for schematics elements: wires, dots, symbol instances, pins, arcs and macroboxes, as well as colors for highlighted and selected objects
- assignment of commands to the keyboard shortcuts
- size, type (dots or lines) and visibility of the grid
- presence of the drawing window's function bar, hint line bar and command line bar
- list of symbols that can be instantiated through the pop up menu
- list of attributes that can be attached to the specific objects through the pop up menu
- place dots automatically at the point of "T"-shape intersections or not
- preserve or break the connectivity while moving elements

Summary

The main features of the Silvaco ***Scholar*** Schematic Capture GUI are described. The implementation of the GUI aspects discussed in the given paper provides the users with effective and efficient user interface for the purposes of schematic design. The ***Scholar's*** GUI has proven to be one of the best-in-class, state-of-the-art front end to the modern schematic design tool.