



Lund University

Department of
Electroscience

CADENCE Condensed

Stefan Molund

September 2001

Abstract

The intention of this manual is to convey some basic instructions on how to use the various CAD tools needed to complete the laboratory courses in both analog and digital IC-design given at the department.

More actual instructions on how to set up the environment and start the various tools, than those in this manual, are given in the laboratory manuals for the different courses. Some of the courses might need some special setup.

The following tools; *Virtuoso Schematic Composer*, *Affirma Spectre Circuit Simulator*, *Virtuoso Layout Editor*, *Assura Diva Verification*, and *Envisia Silicon Ensemble*; presented in this manual are trademarks of **Cadence Design systems, Inc.**

The design kit used in this manual is from **Austria Mikro Systeme International AG**.

Based on previous work by **Pietro Andreani** and **Rifat Zejnic** at the **Department of Electrosience** (formerly known as Applied Electronics) at **Lund University**.

Stefan Molund

Contents

1	CADENCE design tools	1
1.1	Introduction	1
1.2	Cadence User Interface	2
1.3	The Design Process	2
1.3.1	Libraries and Views	2
1.3.2	Instances and Hierarchy	3
1.3.3	The Technology File	3
1.3.4	The SKILL Programming Language	4
1.3.5	The Design Flow	4
1.4	Schematic and Symbol tools	5
1.5	Simulation	5
1.6	Layout Tool	6
1.7	Place and Route	6
2	User Interface	7
2.1	Windows	7
2.1.1	Window Title	7
2.1.2	Status Banner	7
2.1.3	Menu Banner	8
2.1.4	Mouse Settings Line	8
2.1.5	Prompt Line	9
2.1.6	Pop-Up Menues	10

2.1.7	The Cursor	10
2.1.8	The Mouse	10
2.2	Forms	11
2.3	Toolboxes	12
2.4	Dialog Boxes	12
2.5	List Boxes	13
3	Starting Cadence and Library Management	14
3.1	Initializing and Starting	14
3.1.1	Syntax Conventions	15
3.1.2	Shutting Down a Session	16
3.1.3	Working Strategies	16
3.2	Menu Commands	16
3.3	Object Selection	17
3.3.1	Selecting One Object	17
3.3.2	Selection Sets	17
3.4	Libraries	17
3.4.1	The Library Manager	18
3.4.2	Creating a Library	18
3.4.3	Creating a Cellview	19
3.4.4	Copying and Deleting	19
3.4.5	Opening an Existing Cell	20
4	Schematics and Symbols	21
4.1	Creating a Schematic Drawing	21
4.1.1	Opening an Existing Schematic	21
4.1.2	Adding Instances	21
4.1.3	Modifying Instances	23
4.1.4	Pin Connections	24
4.1.5	Wires	24

4.1.6	Verification and Saving	25
4.1.7	Creating Symbols	25
4.2	Plotting a Schematic	26
5	Simulation of a Schematic Design	27
5.1	Preparation of the Schematic	27
5.2	Starting the Simulation Environment	28
5.2.1	Choosing Analysis Type	30
5.2.2	Node Selection and Management	30
5.2.3	Starting a Simulation	31
5.2.4	The Waveform Window	31
5.2.5	DC Operating Points	32
5.2.6	Using Variables	33
5.2.7	Parametric Analysis	33
5.2.8	Shutting Down the Simulation Environment	33
5.3	The Waveform Calculator	34
5.3.1	Entering Data	34
5.3.2	Waveform Analysis	35
6	Building with Layout	37
6.1	Creating Layout Views	37
6.1.1	Edit an Existing Layout	37
6.1.2	Layer and Selection Window	37
6.1.3	Components of a Layout	39
6.1.4	Instance and Object Management	41
6.2	Physical Verification of a Layout	41
6.2.1	DRC	42
6.2.2	Extraction of a Layout	42
6.2.3	LVS	43
6.3	Functional Verification	44

6.3.1	Parasitic Extraction	44
6.3.2	The Configuration File	45
6.3.3	Simulation from the Configuration File	47

Chapter 1

CADENCE design tools

1.1 Introduction

The intentions for this manual is to serve as an introduction to the Cadence design environment and describe the methodology used when designing integrated circuits.

The department is not giving courses in Cadence but in integrated circuit design so only the minimum knowledge, needed to run the laboratories, of Cadence can be gained from this manual. Also this manual describes the environment currently at the department which is Cadence version 4.45 in conjunction with a *Design Kit* from AMS (*Austria Mikro Systeme International AG*) which contains a set of rules and designs for a 0.35 μm CMOS process.

For a more thorough understanding of Cadence the extensive on line manual set is recommended. These are accessed from any of the tools by pressing the **help** button.

More information about the topics in the first two chapters can be found in the manuals *Design Framework II Help* and *Cadence Application Infrastructure User Guide*.

The Cadence tool kit consist of several programs for different applications such as schematic drawing, layout, verification, and simulation. These applications can be used on various computer platforms. The open architecture also allows for integration of tools from other vendors or of own design. The integration of all this tools is done by a program called *Design Framework II (DFW)*.

The DFW-application is the cornerstone in the Cadence environment. It provides a common user interface and a common data base to the tools used. This makes it possible to switch between different applications without having to convert the data base.

This chapter will give an overview of the user interface supplied by DFW and present some of the Cadence tools that will be used.

1.2 Cadence User Interface

In Cadence the user interface is graphic and based on windows, forms, and menus. The main windows of DFW are:

- *Command Interpreter Window* (CIW) is controlling the environment. Other tools can be started from here and it also serves a log window for many applications.
- *Library Manager* gives a view of the design libraries and the different constructions that exists therein.
- *Design Window* (DW) shows the current design. It is possible to have several DW opened at the same time with different, or the same, tools.
- *Text Window* (TW) show text. It can be a log or report that was asked for, or an editor.

The menus in Cadence are mostly pull-downs, i.e. the menu will appear when the title are clicked with the left button on the mouse. There are also pop-up menus that appear in the background of the design window on a middle button press.

The forms are used for entering some specific information that is needed by the function called, the size of a transistor for instance.

1.3 The Design Process

The design tools have a common structure of the designs. It is hierarchical and consists of libraries, views, and instances.

1.3.1 Libraries and Views

All desig data in Cadence are organized in libraries. There are Reference Libraries which contains basic building blocks usable in the construction and Design Libraries which embodies the current design.

Every library consists of cells and their different views, as in figure 1.1. A *cell* is a database object which forms a building block, an inverter for instance. A *view* represent some level of abstraction of the cell. It can be a schematic drawing, layout, or maybe some functional description.

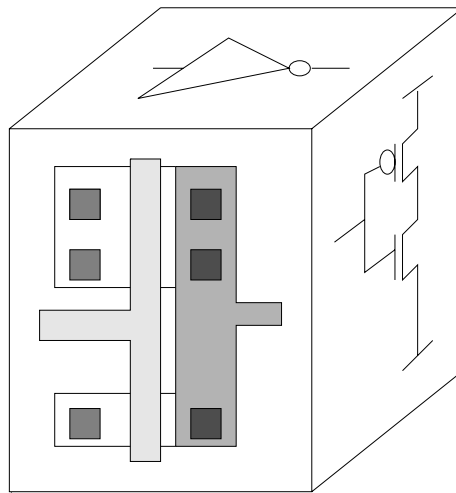


Figure 1.1: An inverter cell with three views: layout, schematic, and symbol.

1.3.2 Instances and Hierarchy

The main reasons for using hierarchical designs is to save design time and minimize the size of the data base. Say that a design would need 500 gates of the same type. Then instead of building it 500 times, it is designed once and then used where it is needed. In this way one cell can be used (**not copied**) several times and each such use is called an *instance* of the cell. In order to be instantiated every cell needs a *symbol* view which acts as a handle to the cell it represents. Only the symbol is shown when a cell is instantiated.

Thus by creating more complex structures by instantiating simple instances a hierarchical design is formed. It is possible to move up and down and work on a selected level in the hierarchy. When a design is opened, the highest level is the default one.

1.3.3 The Technology File

Since there are different semiconductor processes (with different set of rules and properties), Cadence has to know the specifications for the one that is to be used. This information is stored in a set of files called *Technology Files* which exists on different locations on the system. When a library is created it is therefore connected to a specific technology.

The technology files contains information about:

- Layer definitions: Conductors, contacts, transistors ...

- Design rules: minimum size, distance to objects ...
- Display: Colours and patterns to use on the screen.
- Electrical properties: resistance, capacitance ...

The technology files are usually supplied by the silicon vendor, that is to fabricate the design, along with some libraries of standard cells and IO pads that can be used by the designer. Such a collection is called a **Design Kit**.

1.3.4 The SKILL Programming Language

When a command is performed, from a form or a menu, the system is executing functions written in the SKILL language. SKILL is developed by Cadence and is based on Lisp. The Cadence tools are using SKILL for internal communication and for the tool-design communication.

SKILL is also accessible for the designers. Commands can be written in the CIW-window or placed in command files for execution. It can be used for simple tasks like executing a command or building more complex functions to perform various tasks.

1.3.5 The Design Flow

The abbreviated flow in figure 1.2 shows some of the steps in designing integrated circuits in the Cadence environment.

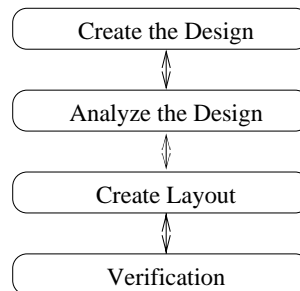


Figure 1.2: The design flow.

The step *Create the Design* consists of drawing schematic views of all cells and blocks. The schematic view contains transistor symbols, and maybe other components such as resistors and capacitances, and wires connecting them. From the schematic view the symbol view is created (almost automatically) so that the cell can be used on a higher level in the hierarchy.

The step *Analyze the design* includes functional verification (simulation) of the design on a schematic level.

The third step, *Create Layout*, is done in a *Layout Editor*. Here the final semiconductor layers are represented by different colours. All the cells and blocks used have the size they will have on the final chip.

The last step is *Verification* of the design. The layout is examined for violations against the geometric or electrical rules, and to verify the function of the physical implementation.

1.4 Schematic and Symbol tools

To create the schematic the tool *Virtuoso Schematic Composer* is used. This editor is an interactive system for building schematics by instantiating some basic components (transistors, capacitances, etc.) and to connect them to each other. The values (*properties*) of the components can be edited to suit the specifications. text and comments can also be included.

The editor will also create symbols of the cells so that they can be used in other parts of the construction.

1.5 Simulation

The simulation tool is started directly from the schematic editor and all the necessary netlists describing the design will be created. A simulation is usually performed in a *test bench*, which is also a schematic, with the actual design included as an instance. The test bench also includes signal sources and power supply. By using parameters for the properties of the components used it is possible to quickly analyze the design for a wide range of variables.

The simulator is run from within *Affirma Analog Circuit Design Environment* which is a tool that handles the interface between the user and the simulator. The current version of Cadence used at the department (4.45) uses the *Affirma Spectre Circuit Simulator*. The simulator offers a wide range of analyses (DC, frequency sweep, transient, noise, etc.) and the results can be presented graphically and be saved.

The results (voltage levels, currents, noise, etc.) can be fed into a calculator which can present various parameters of the analyzed circuit - delay time, rise time, slew rate, phase margin, and many other interesting properties. It is also possible to set up algebraic expressions of in or output signal which can be plotted as a function of some other variable.

1.6 Layout Tool

The *Virtuoso Layout Editor* is used for drawing the layout. A layout consists of geometrical figures in different colours. From the size and colour of these figures it is later possible to generate the final mask layers which are used in the fabrication of the design. It is possible to include other cells by instantiating their layout views.

To verify that the layout fulfills all electrical and geometric rules a *Design Rule Check (DRC)* program is used. This manual will describe *Assura Diva verification* which can be called upon directly from the layout editor. This tool will mark any error in the design and can also *extract* (i.e. convert to a netlist) the layout so it can be simulated.

1.7 Place and Route

The final stage of the construction of a large design is called place and route. This is the process when all the different components of the chip are placed on their locations and connected to each other. Since a design can easily consist of thousands of connection points it would be tedious and time consuming to do the connections manually. The designer might also want to try various alternatives in placing the components, output buffers, memory structures, amplifiers, etc.

The place and route tool that will be described later in this manual is named *Envisia Silicon Ensemble*. It is a very potent program that can place and route a very large design while respecting some *design constraints* (restrictions on delay and size) at the same time.

Usually Silicon Ensemble is used for *Standard Cell* designs - this is when all the cells are of the same height so they can be placed in contact (*abutted*) with each other - but it can handle other structures.

Since not all designs that are to be routed are created in Cadence this manual will describe how to run Silicon Ensemble as a standalone tool. In some other design tools the function of a digital design is described in a functional language which is then compiled (*synthesized*) into a netlist that can be fed into Silicon Ensemble.

Chapter 2

User Interface

The previous chapter gave an overview of the Cadence tools and their user interface. This chapter will penetrate a little deeper into the structure.

The user interface consist of windows, menues, forms, and boxes.

2.1 Windows

Most windows in Cadence looks like the one in figure 2.1, which is the schematic editor. Besides from the big working area, with the design, there are several other important parts of the window that carries a lot of information. At the top there is the Window Title, followed by the Status and Menu Banners. Many windows also have a row of icons for the most common commands at the left. Below the working area there is usually a mouse settings line and a prompt line.

Some windows (CIW for example) have an input line where commands can be entered directly in writing.

2.1.1 Window Title

The title gives information about the window and of the design. The name of the tool activated and what cell and view being edited is shown.

2.1.2 Status Banner

The status banner shows information about commands that are being executed in the actual window. The type of information varies with the different applications. In the example in figure 2.1 it is indicated that the command *Move* is active and no

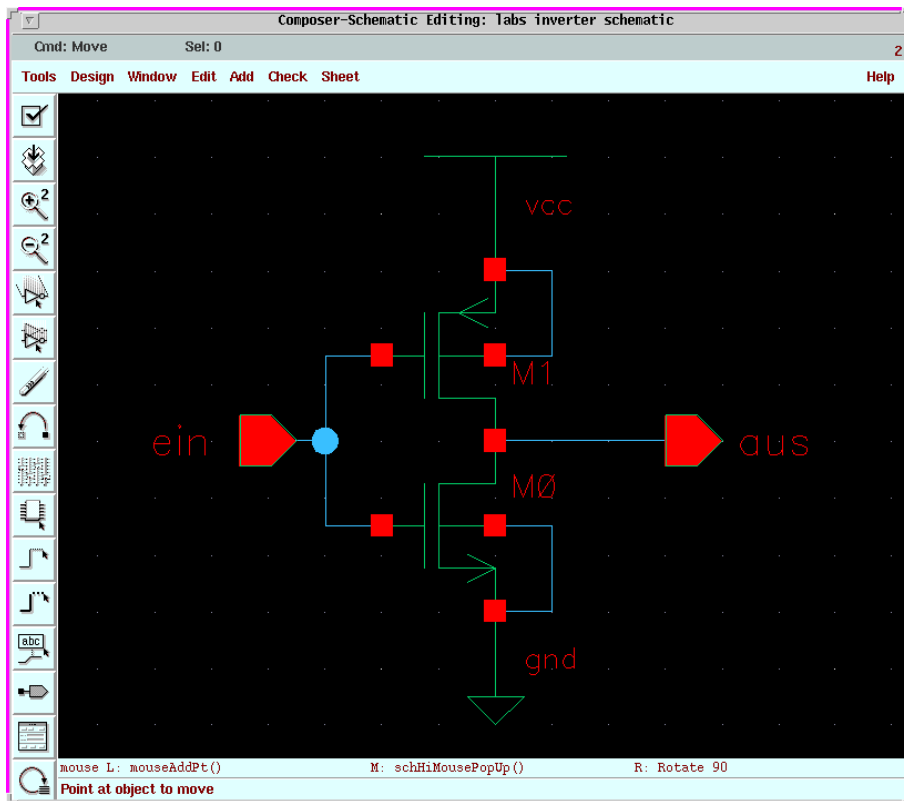


Figure 2.1: The window of the schematic editor.

objects are selected. At the far right is a window identification number which is unique for every window.

2.1.3 Menu Banner

This row contain one or more menu titles. A menu is a list of closely related commands. The menus in Cadence are of the pull-down type, i.e. they are not shown until the title is selected. Figure 2.2 shows what it might look like.

2.1.4 Mouse Settings Line

The current function of the mouse buttons can be found here. In the example it is shown that the right button will rotate the object that is to be moved.

2.1.5 Prompt Line

At the bottom of some windows is the prompt line that gives information on what actions that has to be performed in order to complete the current command.

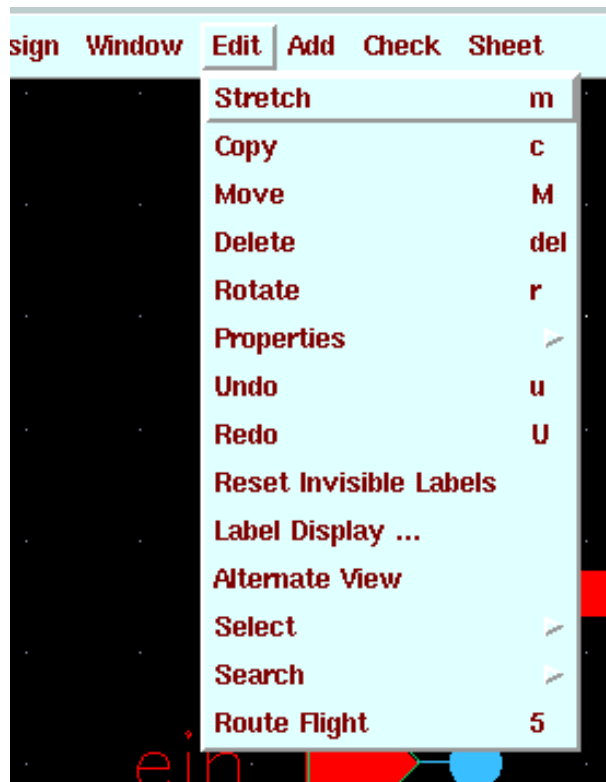


Figure 2.2: A menu banner with a pull-down menu.

A command can behave in one out of three ways:

- When the command is solitary on the row, as *Move* in fig. 2.2 it is immediately executed.
- If the command is followed by three dots (...) as *Label Display...* a form will appear that needs to be filled in with necessary information to complete the command.
- If it is trailed by an arrowhead (▷), like *Properties*, a new set of commands will appear as in fig. 2.3.

If there is a single letter at the right of a command, *c* on the *Copy* line for instance, it shows that this command can be executed by pressing this button on the keyboard.

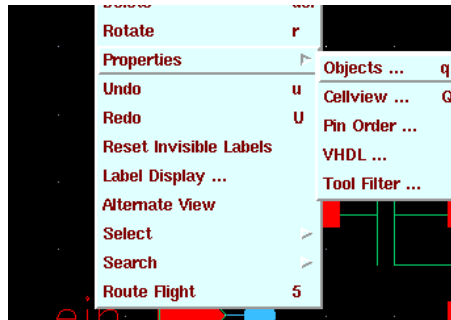


Figure 2.3: Some commands gives a new set of choices.

2.1.6 Pop-Up Menues

In some windows there is a set of commands in a pop-up menu. They will appear when the middle mouse button is pressed in the working area. The commands in these menus are closely related to the task being performed in the actual window.

2.1.7 The Cursor

The cursor can appear in many forms but the most common one is the arrow. Many commands are started by the cursor. As an exemple menu commands are executed by placing the cursor first on the menu title and clicking left and then clicking left again with the cursor on the wanted command when it appears.

The active window is the one receiving input. Only one window at a time can be active. Usually the cursors position selects what window is to be active but in some window managers it has to be explicitly selected by clicking on it.

2.1.8 The Mouse

Almost all commands and tasks within DFW are executed by the mouse buttons.

The left mouse button is used to activate menues, to make selections in forms, and to execute the commands in the icon row.

The middle button is used to activate pop-up menues by pressing and holding the button in the working area of a window. When the menu appears the cursor is slided, with the button still depressed, to the command and the button is then released.

The right button are not used much but in both the schematic and layout editors it is used to zoom in on an area by pressing and holding down the button while marking out the interesting area.

2.2 Forms

By forms some additional information required by the commands can be added. Figure 2.4 shows the plot options form.

Figure 2.4: The plot options form.

The head of the form is usually comprised of a title and a row of buttons. These buttons have the following functions:

OK: executes the command and removes the form.

Cancel: removes the form without executing.

Defaults: resets the selections to some default.

Apply: executes but retains the window.

The body consists of various fields with different choices. The alternatives are:

- A **cyclic field**, as *Plotter Name*. By clicking on this all possible choices will appear and the one wanted can be selected.
- The diamond shaped **radio buttons** at *Orientation* marks some choices of which one must be made.

- the square **toggle button** at *Send Plot Only To File* marks selections that can, but does not have to, be made. It is sometimes accompanied by a **text field** for additional information, a filename for instance. If a text field is greyed out it is not active and it is not possible to enter anything therein.

An other kind of form is the properties form show in fig 2.5 which shows the attributes for a design object, in this case an output with the name *aus*. Attributes are an integrated part of an object and can not be added or deleted but their values can be changed. Properties, on the other hand, can be added or modified.

Property	Value	Display
Name	aus	value
Direction	output	
Usage	schematic	

Figure 2.5: A property form.

2.3 Toolboxes

Toolboxes are windows that mostly consist of buttons. These buttons activates certain functions or open other forms. Toolboxes are used to collect a set of commands that has to be executed on a design.

The figure 2.6 shows an example of a toolbox with commands to do some conversions on all cells in an entire library. The toolboxes can be left opened and moved around the screen.

2.4 Dialog Boxes

During some critical stages a dialog box might appear. These gives warnings or information about what is going on. A typical case is when Cadence is shut down,



Figure 2.6: A Toolbox.

a dialog box appears and asks for confirmation of the command. Fig. 2.7 shows a dialog box. All dialog boxes must be acted upon in some manner, **OK** means go ahead, **Cancel** will abort the operation, and **Help** asks for more information about what went wrong or is about to happen.

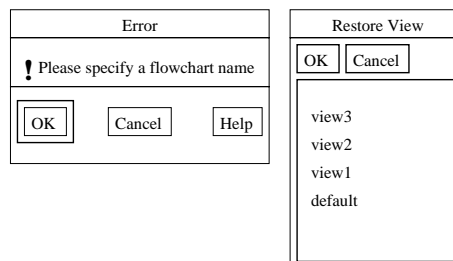


Figure 2.7: A Dialog Box and a List Box

2.5 List Boxes

Some menu commands or form buttons will cause a list box to appear. These shows a list of available objects or names that can be selected. Some list boxes allows selection of multiple choices. This is done by holding down the shift button (ctrl in some cases) while selecting with the left mouse button.

Chapter 3

Starting Cadence and Library Management

This chapter starts by describing how to initialize the environment and to start and shut down the Cadence Design Framework II Environment. After that some tips on how to make the work easier will be given. Some window commands and how to mark objects, which works in the same way for several of the tools, are also covered. The chapter ends with a description of libraries and cells and how they can be treated.

3.1 Initializing and Starting

There are several different commands to start the environment depending on what tools and modules should be included in the environment. The following commands will describe how to set up Cadence 4.45 with the design kit for AMS 0.35 μm CMOS process.

The programs are available from the department, in the es-domain, and from the efd-machines used by the students. Since the environments are not identical there are two set of start-up scripts.

Cadence should always be run in a sub-directory. The programs will copy a lot of files and place them in the current directory. However, there will still be some files placed in the home-directory.

The command **inittde dig2001** (**ana2001** for the analog course) should be used on the efd-machines. The first time this is done the script will ask for confirmaton before building the setup files. Only an answer of “yes” will do. Then Cadence will start up unless a second parameter **nostart** is supplied.

At the department it is slightly more complicated. Follow this procedure

source ~amslibs/v3.30/ams_setup defines the search paths needed for Cadence and AMS.

ams_cds -mode fb -tech csx -tool artist that generates a number of initialization files from a set of templates. This is only required the first time Cadence is started. In the future only

ams_cds -mode fb is required since the information about technology and selected tools now resides in the setup files created.

The first window to appear is the *Command Interpreter Window* (CIW), fig. 3.1, shows one. The *Library Manager* should also emanate but more of it later.

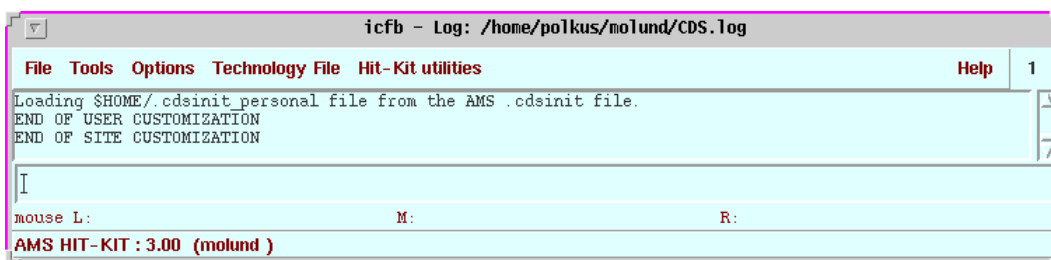


Figure 3.1: The Command Interpreter Window.

The CIW will log any messages concerning the start up phase, when all the initialization files are read, and report any errors in the window.

3.1.1 Syntax Conventions

The following typographic and syntax conventions have been used in this manual.

Design > Plot > Submit means that the command can be found among the menus in the window being currently described. Some frequently used commands also have a *bindkey* attached, i.e. a single button on the keyboard that will launch the command. This is written like this *Add > Wire Name (I)*. It could also look like

CIW: File > New > Library when the command can be found in another window, in this case the Command Interpreter window.

Create New File stands for a name of a form or a field in a form.

3.1.2 Shutting Down a Session

The various tool windows in Cadence can be separately shut down by the command *File > Exit* or *Window > Close*. When shutting down the entire session it is sufficient with *CIW: File > Exit*.

Do not quit the X-window environment with Cadence applications still running, it might cause processes to hang and block licenses.

3.1.3 Working Strategies

The CIW should always be visible. Important log and error messages concerning the design work will appear here.

When a command is activated before the object that is to be affected it will remain active, i.e. the same command can be applied directly on another object. Contrary, if the object are selected first and then the command it will be executed only once. If a form command is to be used several times it is better to execute it with **Apply** since the form disappears with **OK**.

The active command is shown in the status banner of the working window and in the prompt line at the bottom the system gives instructions on how to complete the command. Any active command can be aborted by pressing the *Esc* button. If a command results in something unpleasant it can be undone by *Edit > Undo (u)*. All the changes since last save can be removed with *Design > Discard Edits*.

Save the designs constantly during the work, both the computer system and Cadence can have a bad day.

If a form, that is called upon, does not appear on the screen it might be because it pops up below the ones already visible.

If an object cannot be selected, check the selection filters that stipulates what objects can be chosen, for the schematic editor the command is *Edit > Select > Filter*.

3.2 Menu Commands

The menu banner in the working window have some common titles for several of the different tools used. These are:

Tools The contents of the tool menu varies with the application. By choosing one of the tools the environment will change to reflect the new application. The

simulation environment are started from the schematic editor window in this way.

Design The design menu contain commands to save the design and to traverse in the hierarchy. Other designs can also be called upon.

Window This menu lists commands to zoom and pan in the design. It is also from here the tool is closed.

Edit The Edit commands are used to move, copy, modify, or delete objects.

Add or **Create** is used when new objects or instances are to be created.

3.3 Object Selection

In order to perform an operation on an object it has to be selected. The left mouse button is used for selecting objects. The selected object will be marked by a white rectangle. The objects that can be selected are controlled by *Edit > Select > Filter*. Default is all objects.

All objects selected can be deselected by clicking on a blank part in the working window or by pressing ctrl-d.

3.3.1 Selecting One Object

When the object is outlined by a dashed line it will be selected when the left mouse button is pressed. When a new object is selected the old one is dismissed.

3.3.2 Selection Sets

When several objects are chosen they form a *selection set*. Several object can be chosen at the same time by pressing, and holding down, the left button, then sliding the cursor to form a rectangle covering the interesting objects and then releasing the button. Objects can be added to the set by holding down the *shift* button when selecting, or *ctrl* to remove them.

3.4 Libraries

Cadence will store all design data in library files. Besides the schematics and layouts created by the designer a library will contain information specific to the design process, available layers, design rules etc. This information are compiled

to the library from the technology files, supplied by the vendor, when the library is first created.

3.4.1 The Library Manager

The *Library Manager* window, fig. 3.2, will start up automatically when Cadence is started, otherwise it can be started by *CIW: Tools > Library Manager*. This tool gives an overview of what libraries are available and what cells and their views they contain. Libraries and cells can be copied, renamed, or deleted from the *Library Manager*.

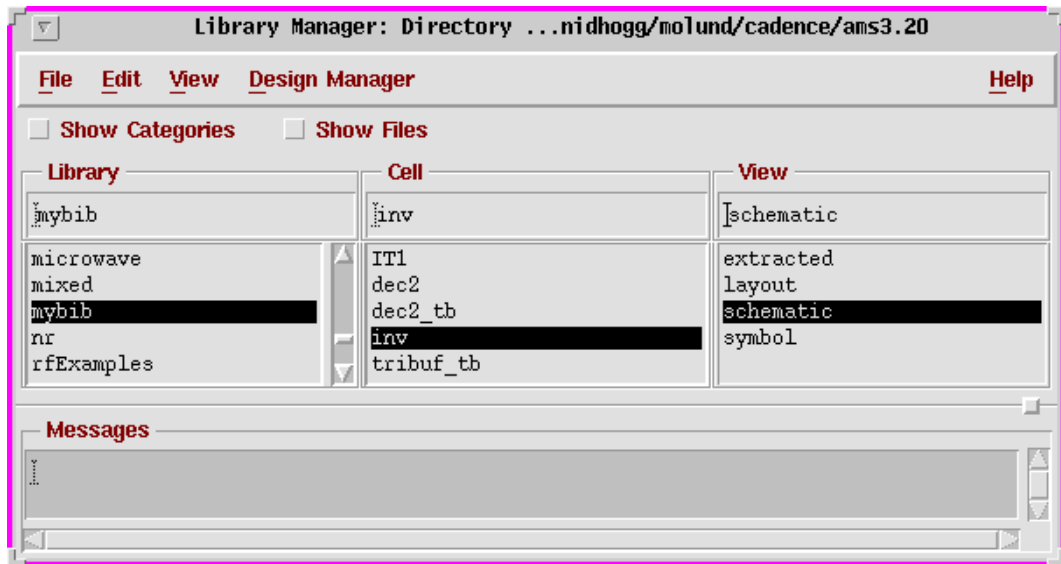


Figure 3.2: Library Manager

3.4.2 Creating a Library

Cadence starts up with a set of predefined libraries, both some basic signal and power generators from cadence and specific transistor models and cell libraries from the design kit. The designer cannot modify these so in order to do any designs a new, personal, library must be created which is done by the command *CIW: File > New > Library*. This will cause a form named **New Library**, fig. 3.3, to appear where the name of the library has to be entered. The path to the library is visible at the bottom of the form, default is the directory from which Cadence was started, which is a good choice. The button at **Attach to an existing techfile** should also be selected before pressing **OK**. In the new form popping up the technology can be chosen, select **TECH_CSI**.



Figure 3.3: Creation of a library

3.4.3 Creating a Cellview

New cells are created by *CIW: File > New > Cellview* and in the form **Create New File** appearing there are some selections to be made.

Library Name Where to store the cell. It must be a library in which the user may create new files. Usually this is only in the own libraries.

Cell Name The selected name of the cell. It is recommended that the name says something about the intended function of the cell.

View Name says what view is to be created. It can either be filled out with the appropriate name, *schematic*, *layout*, etc. or the tool can be chosen at **Tool** and the corresponding view name will be selected, *Composer-Schematic* for a schematic drawing or *Virtuoso* for layout.

3.4.4 Copying and Deleting

To copy libraries, cells, or views is a simple procedure in Cadence.

1. Mark the library or cell that is to be copied in *Library Manager*.
2. Activate the pop-up menu with the middle button on the mouse and select *Copy*.
3. In the form appearing the destination of the copy can be filled in. If a view is copied destination library, cell, and view has to be entered.

Views, cells, or libraries can also be deleted by the pop-up menu. To do this select the unwanted library object and click on *Delete*. After confirmation the object will vanish.

If a library needs to be copied but there is no path to it, say it belongs to an other designer, a link has to be defined for it. This can be done by a tool, fig. 3.4, started with *CIW: Tools > Library Path Editor*. Here the name and complete path to the name can be filled in. It is important to save (*File > Save*) the changes before quitting the *Library Path Editor*. When the link is ready the library can be treated as any other reference library.

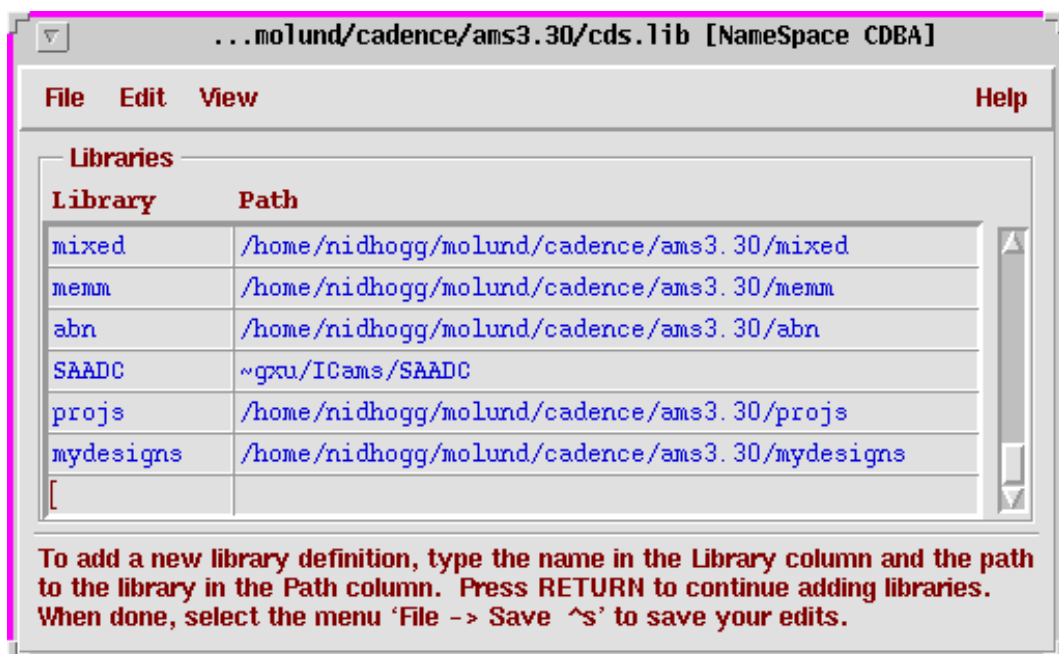


Figure 3.4: The Library Path Editor

3.4.5 Opening an Existing Cell

The pop-up menu in *Library Manager* is also used to open an existing cell for further editing. Select the wanted view and choose *Open* in the pop-up menu and the correct tool will start.

Chapter 4

Schematics and Symbols

This chapter will deal with the creation of schematic drawings and symbol generation. The tool used is called *Virtuoso Schematic Composer* which is a trademark of **Cadence Design Systems, Inc.**

More on this topic can be found in the on-line manuals *Virtuoso Schematic Composer User Guide* and the *Virtuoso Schematic Composer Tutorial*.

4.1 Creating a Schematic Drawing

The library, in which to store the cell, has to be created first. The procedure for this is described in section 3.4.2. The cell can then be created by *CIW: File > New > Cellview*. The view should be set to *schematic*. It is important that *schematic* is spelled correctly. Although Cadence can build a schematic drawing with any viewname, the hierarchy will not work properly.

4.1.1 Opening an Existing Schematic

A schematic drawing that already exists can be opened for editing by selecting the library, cell, and view in the *Library Manager* and then using the middle button on the mouse to bring forward the pop-up menu with the *Open* command.

The command *CIW: File > Open* can also be used.

4.1.2 Adding Instances

A schematic is comprised of components, input and output terminals and their interconnections. Figure 4.1 shows a memory cell consisting of two inverters, two

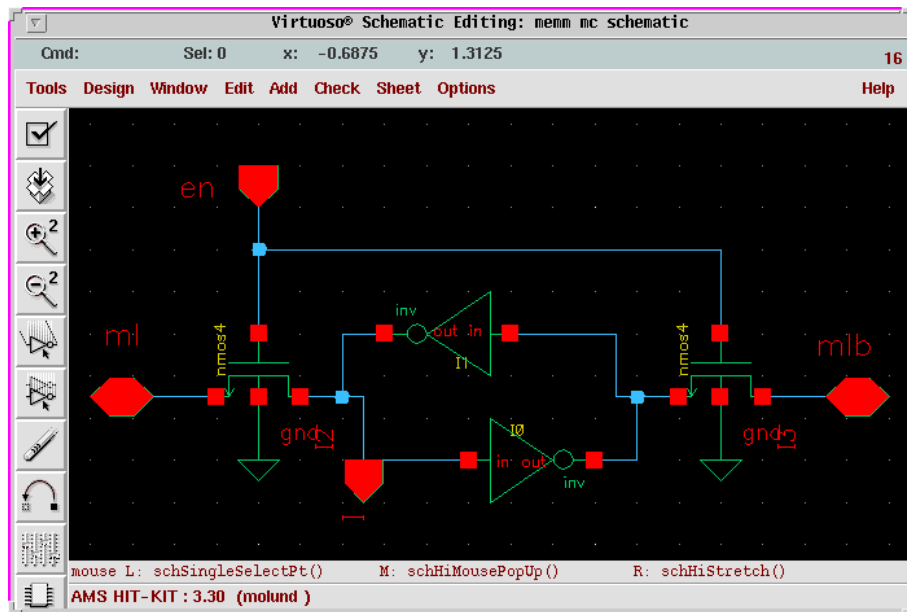


Figure 4.1: The schematic view of a memory cell.

transistors, some terminals, and a global ground connection. Components are included from a reference library as instances in the design. Bigger structures like logic gates or amplifiers can also be instantiated.

1. Select *Add > Instance* and the form **Add Instance**, fig. 4.2, will appear.
2. The fields **Libray**, **Cell**, and **View** has to be filled out to designate the cell that is to be instantiated. By clicking the **Browse** button a browser is started from which the wanted cell can be selected. The view for cells that are to be instantiated in a schematic should be *symbol*.

When the cell has been selected the form will extend to show if there are any parameters to the cell in question, width and length of a transistor for instance.

In the field **Names** a name of the instance can be given or Cadence will supply one. The orientaion can also be changed by the buttons **Rotate**, **Upsidedown**, and **Sideways**.

3. When the form is filled out an outline of the selected cell is connected to the cursor when it is moved back into the design area. If the cursor cannot be moved freely try to switch off **Gravity** in *Options > Editor*.

Place the instance by clicking left when the symbols is at the correct location in the schematic. A new instance of the same type and size appears and can be placed in the same manner. If another cell is wanted select it in the browser.

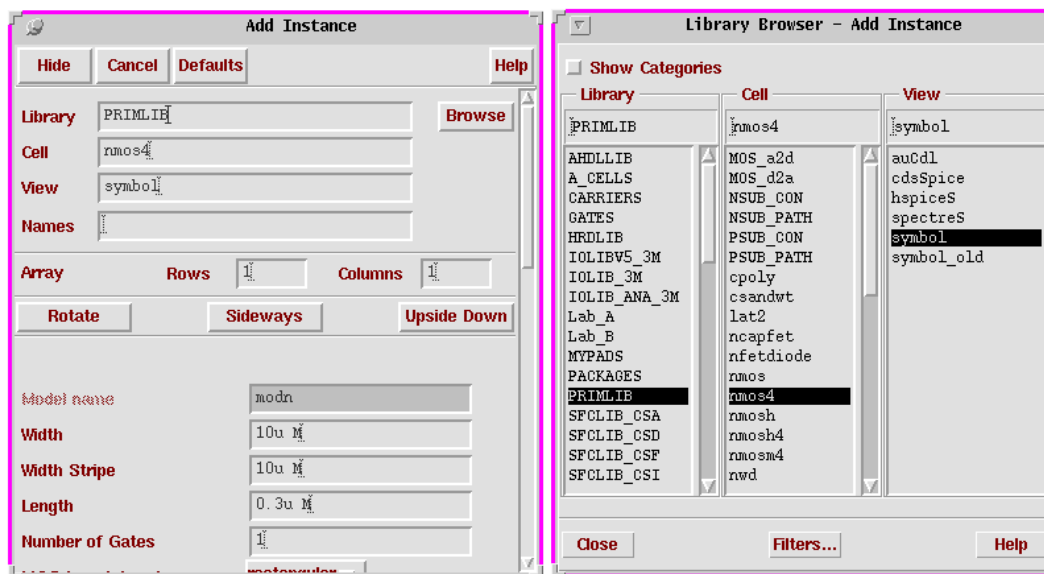


Figure 4.2: Adding an instance

- When all instances are placed click on **Cancel** in the form or press **Esc** to abort the **Add Instance** command.

4.1.3 Modifying Instances

After they are placed, instances can be modified in several ways. Their parameters can be altered, they can be moved and rotated, or even the type can be changed. It is also possible to delete them.

Changing parameters

The parameters, properties, can be modified by first selecting the object and then executing *Edit > Properties > Objects (q)*. In the form any parameter can be changed, it is also possible to change the type of the cell. Press **OK** when finished.

Moving instances

Instances can be moved around or their orientation can be changed by using the command *Edit > Move (M)* and then selecting the object to be moved in the working area. This will cause the instance to be disconnected from the schematic and it will follow the cursor like it did when it was first placed. By pressing **F3** a small form appears from which the orientation can be changed.

The related command *Edit > Stretch (m)* is used to move an instance without disconnecting it from the schematic.

Deleting instances

A selected instance can be deleted by *Edit > Delete* or by pressing the Delete button on the keyboard.

Deleted instances can be brought back with *Edit > Undo (u)*.

4.1.4 Pin Connections

In Cadence *Pins* are used as the primary input and output terminals. They are needed so that the system can connect all signals through the hierarchy and for simulation.

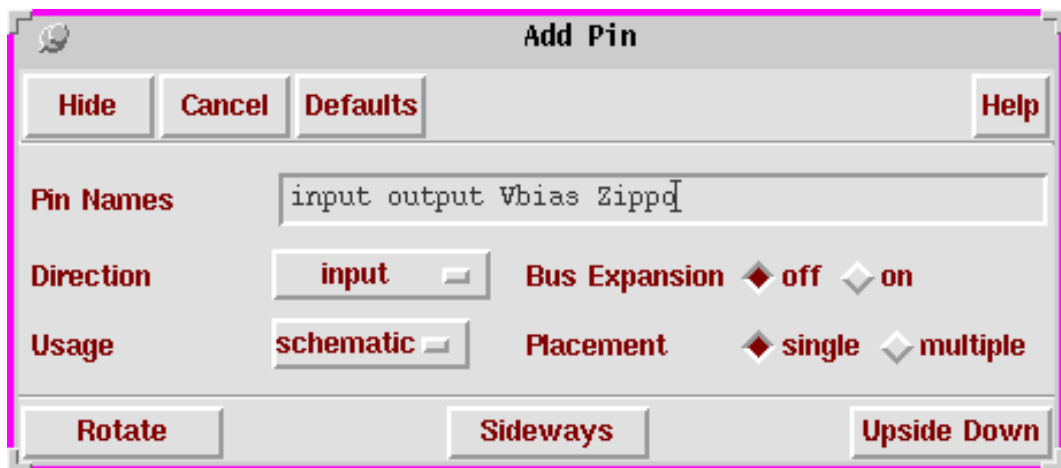


Figure 4.3: The Add Pin form.

A pin is created and placed from the **Add Pin** form which is activated by *Add > Pin (p)*. The names should be entered in the field **Pin Names**. It is possible to enter several names at the same time as in figure 4.3, separate the individual names with a space. **Direction** should be set correctly.

The first pin in the list will be connected to the cursor so that it can be placed, then it will disappear from the list and the next one becomes active.

4.1.5 Wires

Wires are used to connect the terminals of the various instances to each other.

1. Execute *Add > Wire (narrow)* (**w**) in the schematic window.
2. The starting point is selected by clicking on it or by pressing the **s** key when the little yellow diamond shaped marker appears over the terminal.
3. By clicking left on a point it is possible to change direction of the wire.
4. The connection to the end point are made in the same way as at the start, clicking left on the terminal or using **s**.
5. The wire can be ended in “thin air” by clicking twice at the same coordinates.

4.1.6 Verification and Saving

When the schematic is finished it should be checked for errors. The automatic check program will detect errors like floating wires or terminals, short circuits, etc. If a symbol view of the cell also exists, any discrepancies between the schematic and the symbol will be reported.

Use *Design > Check and Save* (**X**) to start the checker. If any errors are found a small dialog box will appear, reporting what went wrong, otherwise the cell will be saved back into the library. Any errors can be examined by *Check > Find Markers* (**g**) which will bring up a form listing the errors and what caused them.

4.1.7 Creating Symbols

All schematics that are to be used as instances by another schematic must have a corresponding *symbol* view. The two inverters in figure 4.1 are symbols of a home-made inverter schematic. Fig. 4.4 shows the windows activated for the symbol generation.

1. *Design > Create Cellview > From Cellview* will bring up the first form, **Cellview From Cellview**. The field **To View Name** should read **symbol**.
2. An **OK** will open the second form **Symbol Generation Options** in which the location of the pins, on the symbol, can be changed.
3. When this form is **OKed**, the symbol editor will start up where further editing can be done if the default rectangular box is not sufficient. The green structures in the view is what will be shown when the symbol is used in a schematic and the red rectangle should contain everything with the pins on the border.
4. Before the symbol editor is closed down the view should be saved!

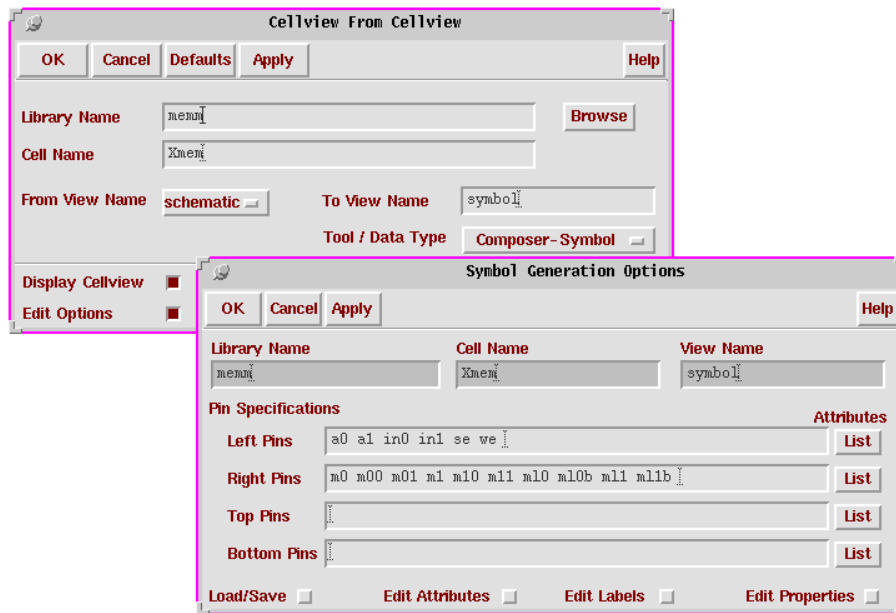


Figure 4.4: Generation of a Symbol View

4.2 Plotting a Schematic

From the departmental machines the following procedure can be used to generate a plot of a schematic.

From a schematic window, with the cell to be plotted showing, the job is launched by *Design > Plot > Submit* and the form **Submit Plot** will open. In this the button **Plot Options** gives a form where the printer name can be changed to a suitable one. The names with a trailing **bw** will convert the view to black and white before sending it to the printer, this works better for some layouts.

Since the efd-computers can not send to the departmental printers and plotters the jobs has to be written to a file instead. In the **Plot Options** form the button **Send Plot Only To File** can be selected and a file name filled in. This will produce a postscript file which can later be sent to a printer or included in a laboratory report.

Chapter 5

Simulation of a Schematic Design

The topic of this chapter is on how to verify a schematic design by functional simulation. The tool used for this are called *Affirma Analog Artist* (trademark of **Cadence Design Systems, Inc**).

The chapter starts by describing how to move from the schematic to the simulation environment, how to perform the simulation, and finally how to view and analyze the results.

More information about simulation and the tools to display the results can be found in the manuals *Affirma Analog Artist Circuit Design Environment User Guide*, *Analog Waveform User Guide*, and *Waveform Calculator User Guide* from the on-line set.

5.1 Preparation of the Schematic

Before the schematic drawing can be simulated some preparation is needed. First, the schematic should pass a check (sec. 4.1.6), to verify that all the terminals are connected and no short circuits exists.

Second a test-bench has to be created. A schematic should only consist of the components that is to exist on the final layout of the design. Therefore, a test-bench is created to contain the other components necessary for simulation. A test-bench, which also is a schematic view, for an inverter is shown in fig. 5.1. From the left it consists of a power source (**vdc**) which sets a voltage difference of 3 volts between the global vdd and gnd symbols **on all levels of the hierarchy**. A signal source (**vpulse**) is used to generate input to the inverter, which is included as an instance. Finally the output is loaded down with a capacitance.

The supply and power sources are picked from the library **analogLib** and the most

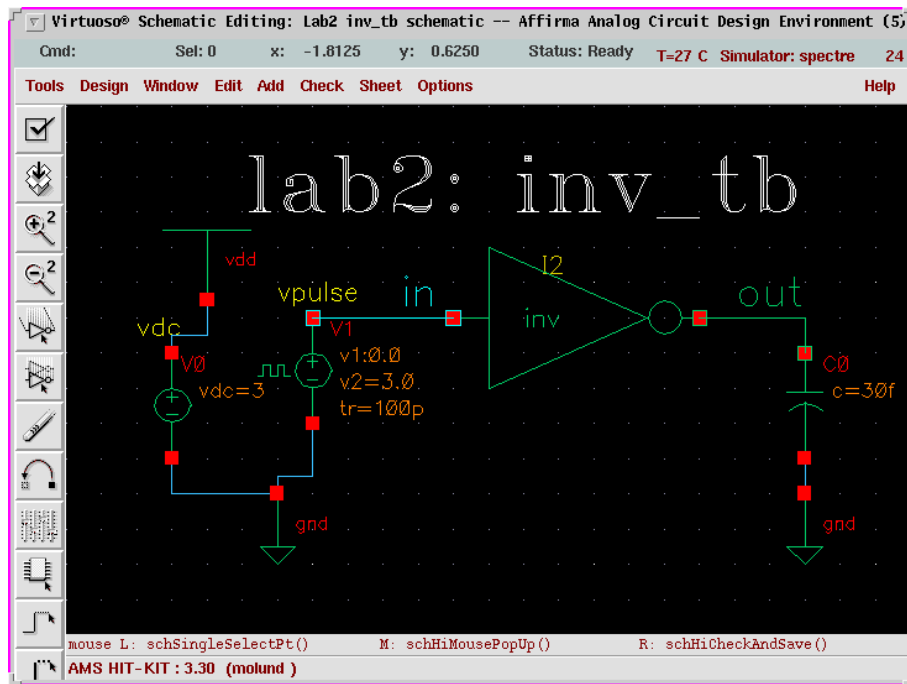


Figure 5.1: A test-bench for an inverter.

common are:

vdd, gnd: Global power connections between all levels in the hierarchy.

vdc: Power source. Parameter for voltage.

vpulse: Pulsed signal. Settings possible for voltage, delay, rise- and fall time, etc.

cap: Gives a capacitance of selected value.

5.2 Starting the Simulation Environment

The environment is started directly from the schematic editor by the command *Tools > Analog Environment* which will cause the controlling window, **Affirma Analog Circuit Design Environment**, of the simulator to materialize.

The main simulation window contains four fields with information regarding the current simulation, circuit designation, a list of the type of analyses chosen, the variables - if any, and nodes selected to be saved.

As shown in figure 5.2 the cell that is to be simulated is already filled in and the default simulator can be seen in the status banner. If it isn't **spectre** it can be

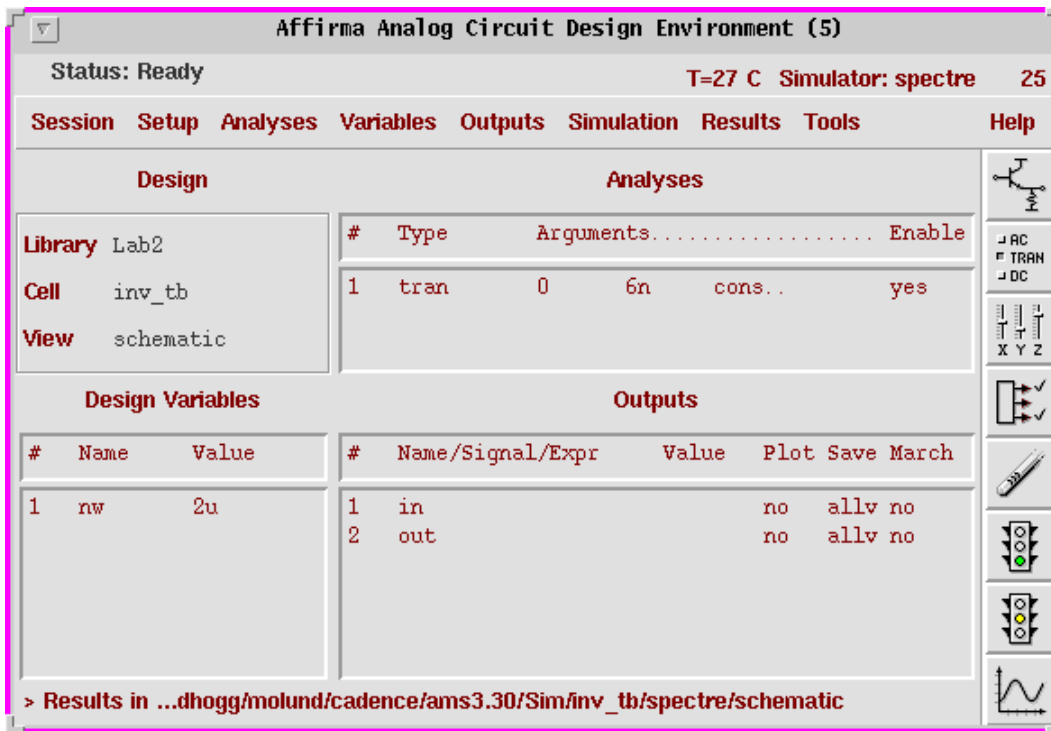


Figure 5.2: The simulator main window.

changed in the *Setup* menu. In this menu there are commands for changing the environment, what is to be simulated, where the model files are located, etc.

Temporary and result files, needed during the simulation, are stored in the unix directory `./Sim`, which is in the users own file system. During large circuit simulations this can grow tremendously in size and cause quota problems. To avoid that problem this directory can be changed at **Project Directory**, in the form gotten by *Setup* > *Simulator/Directory/Host*, to `/tmp/<user>/sim` for instance. This library can later be removed with no permanent loss since all data therein will be recreated during the next simulation.

In the *Analyses* menu are commands to select the kind of simulation to be performed, AC, DC sweep, transient, noise analysis, etc.

From *Variables*, some user defined variables for component properties can be handled. More in section 5.2.6.

The node currents and voltages to be analyzed are controlled by commands in the *Outputs* menu. Data can be observed interactively during the simulation or saved to disk for further study.

The most important commands in *Simulation* are *Netlist and Run* and *Stop*. Some

specification of initial conditions can also be made here.

Results controls all data management after the simulation. The menu is used to select what node currents or voltages are to be displayed in a waveform window.

From the *Tools* menu it is possible to start a calculator to perform further analysis of the simulation data. Here a *Parametric Analysis* can also be started. This means that a circuit's behaviour for a certain parameter is analyzed by sweeping the parameter over a defined interval of values. This is described in section 5.2.7.

5.2.1 Choosing Analysis Type

The analysis type is chosen by *Analyses > Choose*. In the resulting form the kind of simulation to be performed can be selected along with several options.

5.2.2 Node Selection and Management

There are two ways of presenting simulation data. The first is to plot the data in a waveform window continuously during the simulation. This way is useful for short simulations. The drawback is that the signals are only shown in the waveform window, it is not possible to do any further analysis of the data.

For a more complex circuit it is more convenient to save the simulation data to disk to be viewed after the simulation. Then it is also possible to perform operations on the curves shown.

Selecting Nodes

The command *Outputs > To Be Saved > Select On Schematic* in the simulator window activates selection mode. Thereafter nodes are selected by using the left mouse button.

Voltage levels are selected by clicking on the wire connections.

Currents are selected by choosing the nodes, i.e. the red boxes.

Selected wires are highlighted, nodes circled, and their names will be listed in the window.

By using *Design > Hierarchy > Descend Edit* in the schematic window it is possible to traverse the hierarchy and select nodes not visible from the top level.

In order to get the nodes automatically plotted after the simulation *Outputs > To Be Plotted > Select On Schematic* can be used. The last alternative is to have the

nodes plotted during the simulation which is activated by *Outputs > To Be Marched > Select On Schematic*.

Modifying Selected Nodes

The command *Outputs > Setup* gives a form in which the selected set of nodes can be altered. The selections of *Plot*, *Save*, or *Marched* can be changed. Nodes can be deleted and it is possible to define operations to be done on the node values.

Saving the Configuration

All the configuration done in the window can be saved and later loaded back into the simulator. This is controlled by *Session > Save State*. In the form the parts that is to be saved are selected and given a name to store it under. The files will end up in the directory **.artist_states** in the users home directory.

5.2.3 Starting a Simulation

The simulation run is started by *Simulation > Netlist and Run* or by clicking on the green traffic light among the icons. The simulator uses its own log window to report on events and errors.

A running simulation can be halted by *Simulation > Stop*. It is not possible to restart a halted simulation, it must be run from the start each time.

5.2.4 The Waveform Window

After the simulation, the signals for which **Plot** was chosen will be visible in a waveform window, like the one in fig. 5.3. Other nodes are selected by *Results > Direct Plot > Transient Signal* - if it was a transient simulation - and selecting them from the schematic, as instructed.

When the window appears all curves are plotted together. It is possible to separate them by *Axes > To Strip, ... > To Composite* brings them back together. The curveforms can be moved among each other by clicking and dragging on them.

It is possible to zoom among the curves presented and the cursor tracks the closest curve with the values visible in the status banner.

By default the window is recreated after each simulation. In order to add a curve to an existing window without causing a new one to appear the option **Overlay Plots**, in the form given by *Results > Printing/Plotting Options*, must be selected.

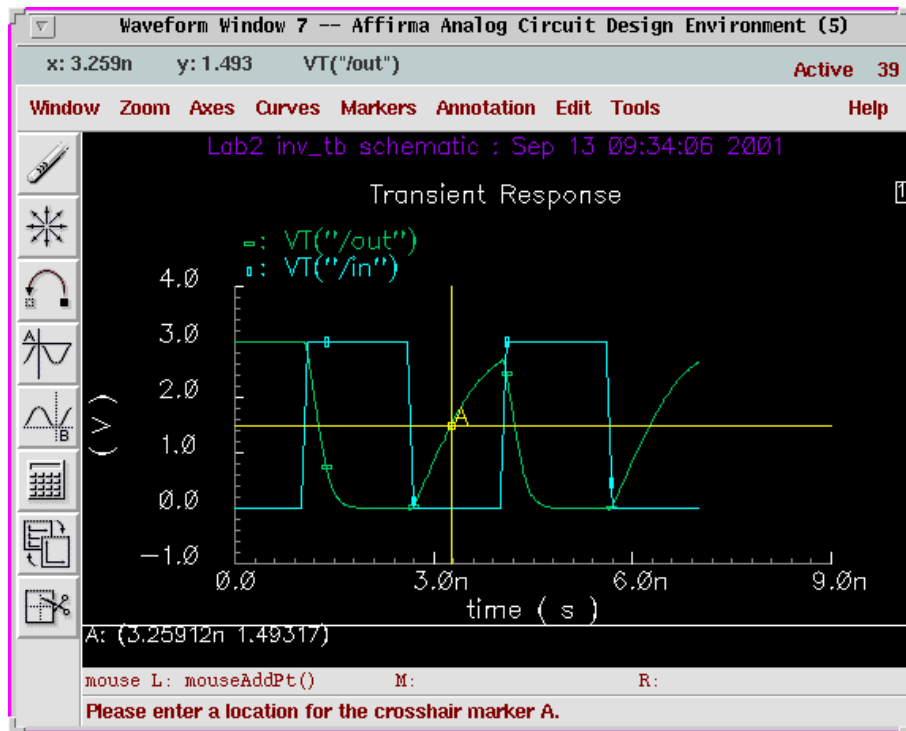


Figure 5.3: The Waveform Window.

To improve documentation text strings can be added to the waveform window. It is very useful for comments etc. *Annotation > Edit* gives a form that handles this. After entering suitable text, size, and colour a click on the **Add** button will enable the string to be placed in the window. The title and subtitle of the window can also be modified from the *Annotation* menu.

To get a plot of the waveform window there is a form activated by *Window > Hardcopy* which looks much like the one from the schematic window.

5.2.5 DC Operating Points

The DC operating points for transistors are often interesting to view. Given that a DC simulation has been run they can be viewed by *Results > Print > DC Operating Points* and selecting the transistor to be presented. The result will be presented in a new window from which the results can be saved.

5.2.6 Using Variables

By replacing component property parameters with variables, instead of specific values, it is possible to directly change the values from the simulator. Otherwise the schematic would have to be modified and saved before the new parameters are visible to the simulator.

In the properties form (*Edit > Properties > Objects (q)*) a variable is written in place of a value for a parameter. “**nwidth M**” for the width of the n-transistor and “**2.2*nwidth M**” for the width of the p-transistor in an inverter design for example. By setting a value on the **nwidth** variable from the simulator it is possible to quickly change the strength of the inverter.

The variables are read into the simulator from the form **Edit Design Variables** started by *Variables > Edit*. The button **Copy From** will include all variables from the schematic into the simulator. After editing the variables the simulator can be started directly from the variables form by clicking **Apply & Run Simulation**.

5.2.7 Parametric Analysis

By a parametric analysis several simulations are run after each other while a component parameter is varied through an interval. The procedure is related to the analysis with variables but the process of setting a new value on the variable has been automated.

The component parameter must be a variable for a parametric analysis.

From *Tools > Parametric Analysis* the **Parametric Analysis** form is started. The name of the variable to be swept are to be filled in at **Variable Name** and in the lower field the limits of the interval are entered.

As an example the value of the load capacitor in fig. 5.1 can be a variable that is then swept through several values. The result is presented as a set of curves in the waveform window as in fig. 5.4.

5.2.8 Shutting Down the Simulation Environment

When all simulations and analyses are run the environment is closed down by *Session > Quit* from the main simulation window. This will close down all associated windows in an orderly manner.

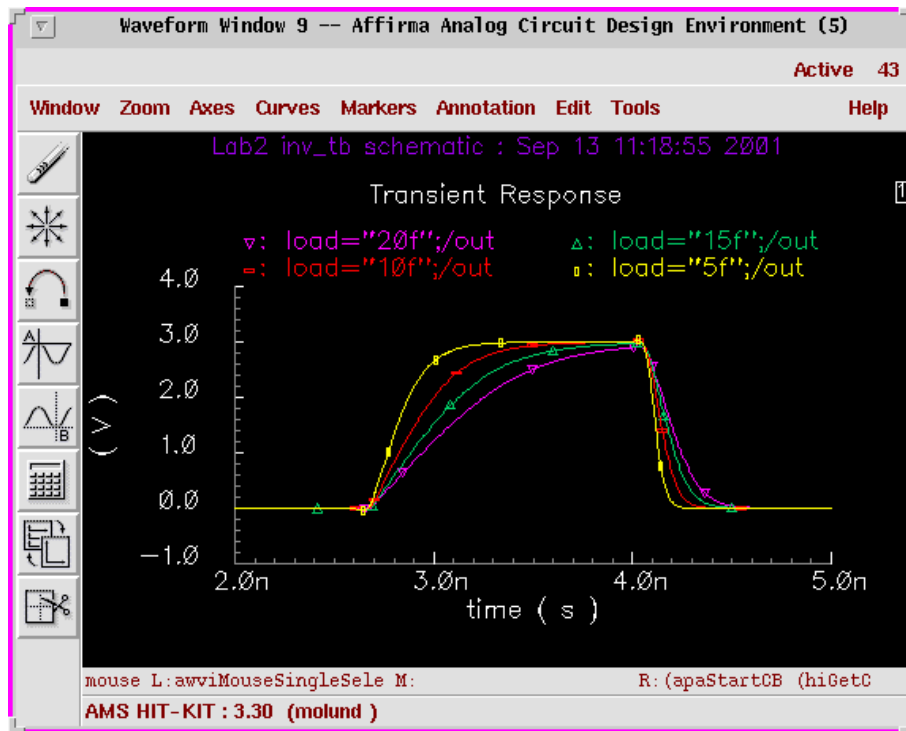


Figure 5.4: Results of a Parametric Analysis.

5.3 The Waveform Calculator

The *Waveform Calculator* is a tool for performing advanced calculations on the simulation data. Node values can be entered from the schematic view or from the waveform window. The calculator is started by *Tools > Calculator*.

The calculator works with *Reverse Polish Notation (RPN)* which means that the operands are entered first followed by the operation to be performed on them. This mode can be changed by *Options > Set Algebraic*.

5.3.1 Entering Data

There are several buttons for entering simulation data to the calculator depending on the simulation type that generated them.

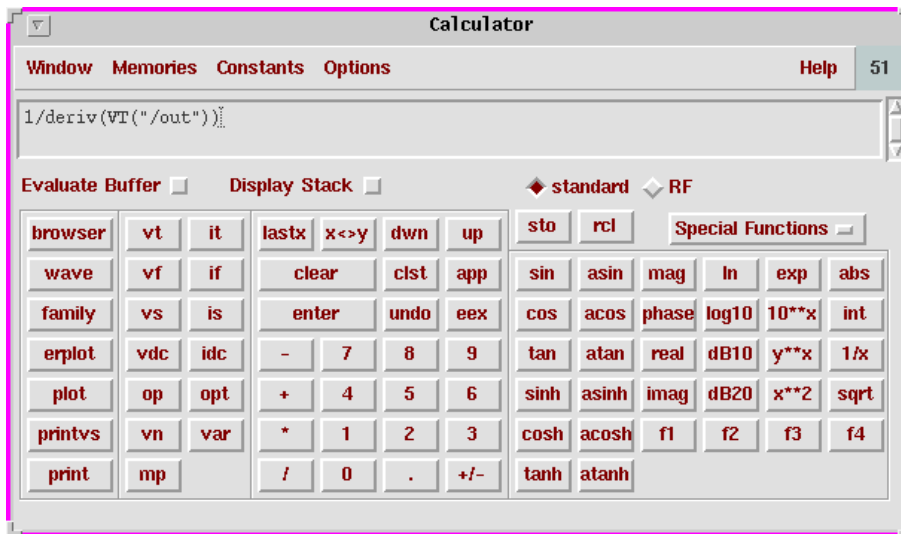


Figure 5.5: The Waveform Calculator.

vt	transient voltage	it	transient current
vf	frequency voltage	if	frequency current
vs	sweep voltage	is	sweep current
vdc	DC voltage	op	DC operating point
vn	noise voltage	opt	transient op. point
var	design variable	mp	model parameter

The data are entered by first clicking on the appropriate button in the calculator followed by selecting the wire (voltage) or node (current) of interest in the **schematic** editor window. For example, in figure 5.1 **vt** followed by selecting the output wire labeled “out” would cause **VT("/out")** to appear in the calculator window.

By using the button **wave** it is possible to enter a curveform from the waveform window.

5.3.2 Waveform Analysis

After loading a curve into the calculator several operations can be executed on it by the buttons on the calculator, inverted, squared, multiplied by a factor, etc. It is also possible to add or subtract several curves. The result of the expression shown can be viewed by pressing **Plot** which will cause the resulting curve to be drawn in the waveform window. **Print** is used for operations resulting in a value rather than a curve.

Special Functions

There is a **cyclic field** named **Special Functions** in the calculator window. This contain several functions of interest when analyzing a circuit, derivating, integrating, rise- and falltime, slewrate, etc. Figure 5.6 shows the windows involved when calculating the rise time of a signal.

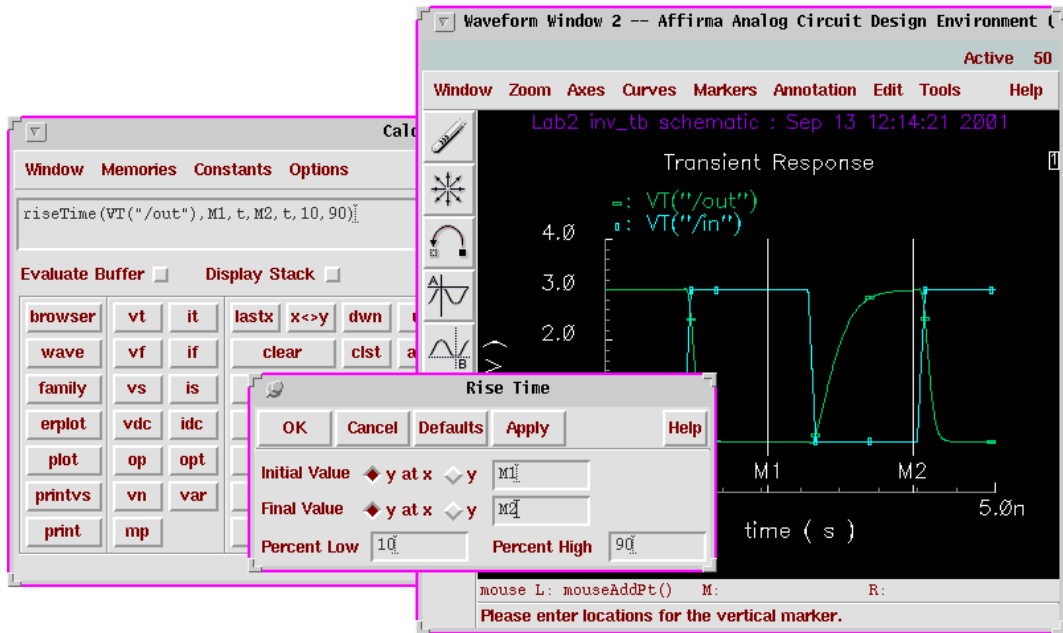


Figure 5.6: Calculation of rise time of an output.

The function **riseTime**, like many others, needs an interval in which to do its calculations. These intervals are easiest defined by setting **Markers** in the waveform window. By *Markers > Vertical Marker* and **Add Graphically** it is possible to define two points in time (**M1** and **M2**) that encompass the interesting rising flank. In the **Rise Time** form this markers can then be used. By clicking **Apply** the calculator window changes to that of fig. 5.6. Finally **Print** will cause the value to appear in a window.

A very useful command in the **Markers** form is **Display Intercept Data** which will give a text window with information of existing markers.

Chapter 6

Building with Layout

This chapter consists of two parts. The first describes the generation of layout views and the second deals with the various tools used for verifying the layout, both physical and functional. The main tool used is the *Virtuoso Layout Editor* which is a trademark of **Cadence Design Systems, Inc.**

More information can be found in the Cadence manuals *Virtuoso Layout Editor User Guide* and *Cadence Hierarchy Editor User Guide*.

6.1 Creating Layout Views

Like schematics, the library in which to store the layout has to exist before the layout can be initialized. The layout view is created by *CIW:File > New > Cellview*. The form should be filled out exactly as for the schematic, but the **View Name** should be *layout*.

6.1.1 Edit an Existing Layout

The fastest way to open an existing layout is to mark it in the Library Manager and select *Open* in the pop-up menu on the middle mouse button.

6.1.2 Layer and Selection Window

When an edit session starts, an extra window (*Layer and Selection Window LSW*) will appear, fig. 6.1. It shows the various layers available for the selected process. The little box with a two letter combination to the right of the layers denotes the

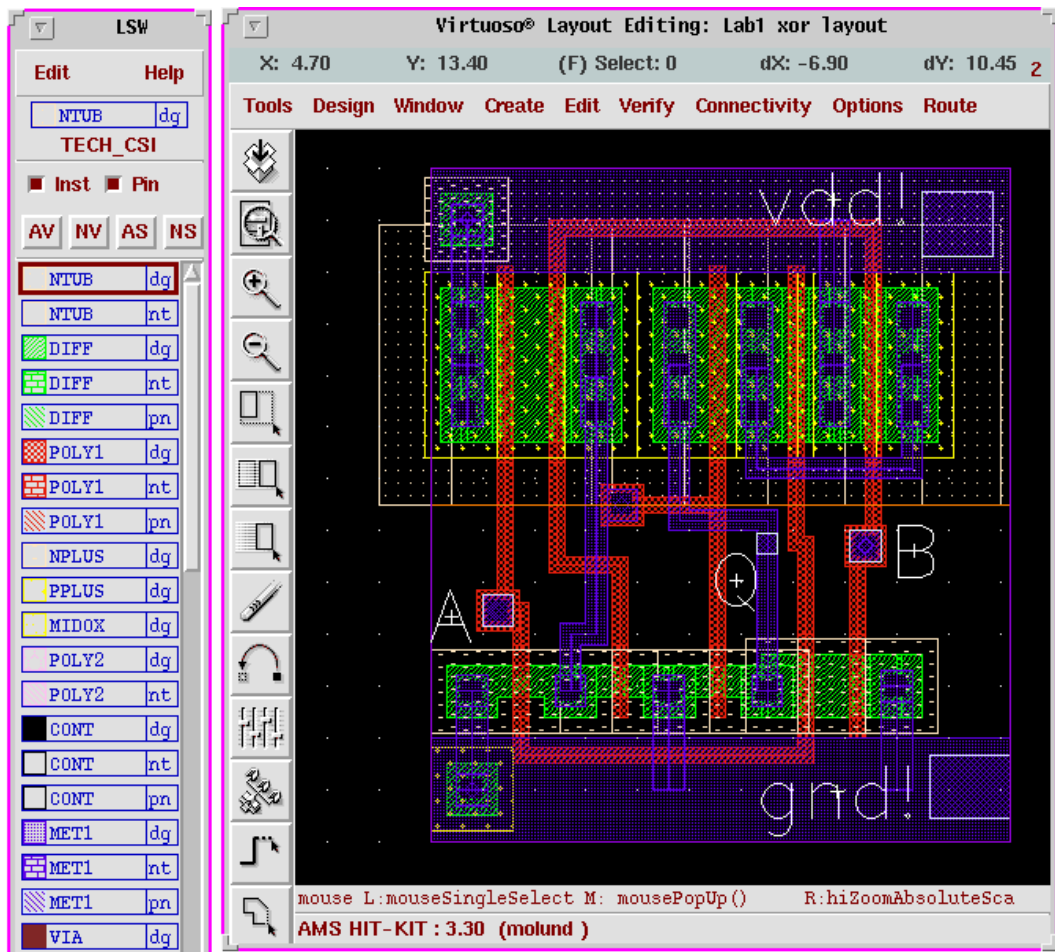


Figure 6.1: The LSW and Layout Editor windows.

special purpose of the layer. Everything that is to be manufactured must be done in the **dg** (drawing) layer. **pn** (pin) is used for the pins.

The window also controls layer visibility and selectability in the following way:

- The left button sets the current drawing layer, i.e. everything hereafter will be done in the selected layer. The drawing layer is shown at the top of the LSW along with the name of the technology.
- By using the middle button the visibility of the layer under the cursor is toggled on or off. If set to off that layer will not be drawn in the editor window after the next redraw command (*Window > Redraw (ctrl-r)*). The buttons **AV/NV** changes visibility for all layers at the same time.
- The right button (or **AS/NS**) controls weather an object in the actual layer

can be selected in the editor.

- The selectability for instances and pins are controlled by their own buttons.

6.1.3 Components of a Layout

The layout represents the final design that is to be manufactured. The fabrication masks are later generated from the layout data. A layout is composed of rectangles representing different layers with varying characteristics. Other layout cells can be included as instances to form a hierarchical design.

The various layers in the available AMS process are:

<i>MET1</i>	Metal 1 Layer	Conductive
<i>MET2</i>	Metal 2 Layer	Conductive
<i>MET3</i>	Metal 3 Layer	Conductive
<i>DIFF</i>	Diffusion Layer	Semiconductive, forms transistors
<i>NPLUS</i>	n^+ Implant Layer	Affects DIFF
<i>PPLUS</i>	p^+ Implant Layer	Affects DIFF
<i>POLY1</i>	Poly 1 Layer	Conductive and Gate
<i>POLY2</i>	Poly 2 Layer	Forms capacitor with POLY1
<i>NTUB</i>	n^- tub Layer	N Well
<i>FIMP</i>	n^- field Implant Layer	
<i>CONT</i>	Contact Layer	Opening between MET1 and DIFF or POLY
<i>VIA</i>	Contact Layer	Opening between MET1 and MET2
<i>VIA2</i>	Contact Layer	Opening between MET2 and MET3

Instances

As with schematics common components are picked from a library and instantiated in the design by the command *Create > Instance (i)*. In the form the *layout* view of the wanted component is chosen along with its orientation and parameters, if any.

Paths

The components thus instantiated in the layout view are then connected by drawing wires between the terminals. *Create > Path (p)* is used to connect to objects by a wire. The path is drawn in the current drawing layer and by pressing **F3** a form is called upon, in which the width of the wire can be altered. The wire is ended by clicking twice on the same point. Information on what the designer has to do to complete the active command is given in the prompt line in the layout editor

window. Short straight wires can also be drawn with *Create > Rectangle (r)* which will draw a rectangle specified by two corner points selected.

Contacts

Contrary to the schematic, contacts are an important concept in a layout. When two objects of the same layer (colour) touches, they are in contact with each other. If the layers differ a contact is needed to connect the layers. For instance a contact between the metal1 and metal2 layers, in a process, consists of the two layers mentioned plus a contact hole. For fabrication this means that an opening in the oxide layer, that normally separates the two metal layers, are created so that the higher metal (2) can reach and make contact to the lower one.

Fortunately contacts are predefined and can be called upon by the command *Create > Contact (o)*. The form has a **cyclic field** with all the contacts available. The **Width** and **Length** fields must not be changed since the size of the contact hole may not be altered. To change the size of the contact, **Rows** and **Columns** are used.

The contacts available in the AMS process are:

ND_C	contact between n-diffusion and metal1
PD_C	contact between p-diffusion and metal1
P1_C	contact between poly1 and metal1
P2_C	contact between poly2 and metal1
VIA_C	contact between metal1 and metal2
VIA2_C	contact between metal2 and metal3

When instantiating transistors there are some parameters that affects the contacts on the transistor. It is possible to get the transistor with or without contacts on the source and drain terminals. The generation of substrate contacts connected to the transistor can also be controlled.

Pins

Pins has nothing to do with the final layout but are essential for the workings of the hierarchical structure and functional verification of the design by naming part of the nets in the structure so that it can later be simulated.

For big designs, mostly digital, pins are also used by the automatic routing tools when creating all the connecting wires between the standard cells used.

Pins must be created in the specific pin-layer. If the wire on which to put the pin is in the **MET1/dg** layer, the pin should be created in **MET1/pn**.

Pins are created by:

1. Selecting the layer in LSW.
2. Executing *Create > Pin* and selecting **shape pin** followed by **Display Pin Name** and entering the names of the pins to be created.
3. The **I/O Type** of the pins should be set as in the schematic.
4. For the global power, **vdd!** and **gnd!** are used as names. The type is not important for the power pins.
5. The pins are then placed by marking a rectangle in the layout.

The pins should be completely covered by its corresponding drawing layer and must otherwise respect the same layout rules.

6.1.4 Instance and Object Management

Instances, contacts, paths, and rectangles are all design objects whose properties and locations can be modified.

Like in the schematic editor parameters are modified by first selecting the object and then executing the command *Edit > Properties* (**q**).

An existing object can be copied by *Edit > Copy* (**c**). **F3** will bring up a form in which the layer can be changed to a new one for the copy.

The *Move* command works as in the schematic editor but the *Edit > Stretch* (**s**) can be used to extend existing wires or rectangles in a direction by selecting one of its edges.

Sometimes it is convenient to change the origin (origo) of a design. This is executed by the *Edit > Other > Move Origin* command after which the cursor is used to select the new point of origin.

6.2 Physical Verification of a Layout

The physical verification of the layout is made in two steps. The first, *Design Rule Check* (DRC), examines the layout with respect to the geometrical design rules. Thereafter it is extracted (i.e. a netlist is created from the layout view) as a preparation for the second step, *Layout Versus Schematic* (LVS), which compares the schematic view with the extracted and reports on differences.

6.2.1 DRC

A *Design Rule Check* verifies that the layout fulfills the geometrical rules for the different layers of the selected process (minimum sizes, spacings, etc.) Errors will be reported and marked to facilitate correction. An *Electrical Rule Check* (ERC) is also run which checks for electrical errors like short circuit, latch-up, floating nodes, etc.

1. Chose *Verify > DRC* and the **DRC**-form will appear.
2. The verification can be done in a **flat** or **hierarchical** mode. In the former, all instances will be lifted up to the top level before the checking is done. In the **hierarchical** mode the DRC will check multiple occurrences of the same instance only once which should speed up the execution time on large constructions. The **flat** option should be selected for designs in the AMS process.
3. With **Set Switches** it is possible to set some switches that affect the kind of checks to be made. It is possible to generate some of the necessary implant layers by selecting “generate_FIMP” or to ignore its absence with “no_FIMP”.
4. When the setup is done click on **OK** to start the DRC.
5. The logging in the CIW will show the errors and they are also marked in the layout window. They can be studied more closely by the command *Verify > Markers > Find*.

6.2.2 Extraction of a Layout

When the layout satisfies all the layout rules it is ready for extraction. The extracted netlist will consist of the the different components and how they are connected. It is also possible to get information of the capacitances on the connecting wires.

1. *Verify > Extract* will open the extractor form.
2. It is possible to do a **flat** or **hierarchical** extraction. **Full hier** will cause the whole circuit to be extracted while **incremental hier** only processes what has been changed since the last extraction. **flat** is used for AMS.
3. Select the wanted type and click **OK**. The extractor will now create a new cellview with the same name as the layout but with the view name *extracted*.
4. Errors during the extraction phase are treated the same way as errors from the DRC.

The new view (*extracted*) can be viewed by opening it from the *Library Manger*. It looks like the layout but symbols of the components extracted has been added.

Figure 6.2: The form: Layout Versus Schematic (LVS).

6.2.3 LVS

The *Layout Versus Schematic* (LVS) will compare the layout against the extracted view and report any discrepancies. In order for this to work the pin names of the schematic and layout views has to be identical.

1. The LVS is started by *Verify > LVS*.
2. If an LVS check has been performed earlier on another cell the contents of some setup files will differ from the current form. Then an **LVS Form Contents Different** message will appear. **Form Contents** should be selected.
3. The names and views in the form (figure 6.2) must be filled out correctly.

4. For layouts in the AMS process the fields **Rules File** and **Rules Library** must look like the figure. They will be filled in if LVS is launched from the extracted view.
5. When all filling in is done a click on **Run** will start the verification.
6. After the analysis is done a small box with the message **Analysis Job Succeeded** shows up. This only means that the program has finished and not that the LVS check was satisfactory.
7. A click on **Info** in the **LVS** form will produce a **Run Information** window in which the results can be viewed. The option **Output** gives a short logfile in which the string "The net-lists match" informs that all is well.
8. If not, the log file contains information about the differences detected. Also, the button **Error Display** will start a simple error handler which can show and explain what differs.
9. The error handler is closed by **Cancel** and the **LVS** form is shut down with *Commands > Close Window*.

6.3 Functional Verification

In order to verify the function of the layout a Post-Layout Simulation is performed. This means that a simulation of the netlist created by the extraction tool is performed. The extractor can also estimate the parasitic components that always exists on the layout. These can have great impact on the function and performance of the circuit regarding delays and switching performance.

A configuration file must also be created to describe the design and what components and instances it contains. This is done more or less automatically.

6.3.1 Parasitic Extraction

The extraction program is started as before, 6.2.2. In the **Set Switches** form *capall* should be selected before the extraction is started. The extractor will calculate the parasitic components and include them in the *extracted* view.

Again **LVS** is used to compare the two cellviews. The parasitics does not exist in the schematic but **LVS** can manage that. After a succesful comparison the **Build Analog** button is selected, also **enable all**. This will convert the *extracted* view into an *analog_extracted*, that can be read by the simulator.

6.3.2 The Configuration File

The configuration file is a description of which parts (cells) that builds up the design. The view to be used, in the simulation, can be selected for each cell. For time-critical structures the analog_extracted are used and for less important cells it might be enough to use the schematic view.

The configuration file is generated from a **schematic** view that depicts the structure. This means that a *test-bench*, which is a schematic, is first constructed and then the configuration file is generated from it. The test-bench that was used to verify the function of the schematic design can be used.

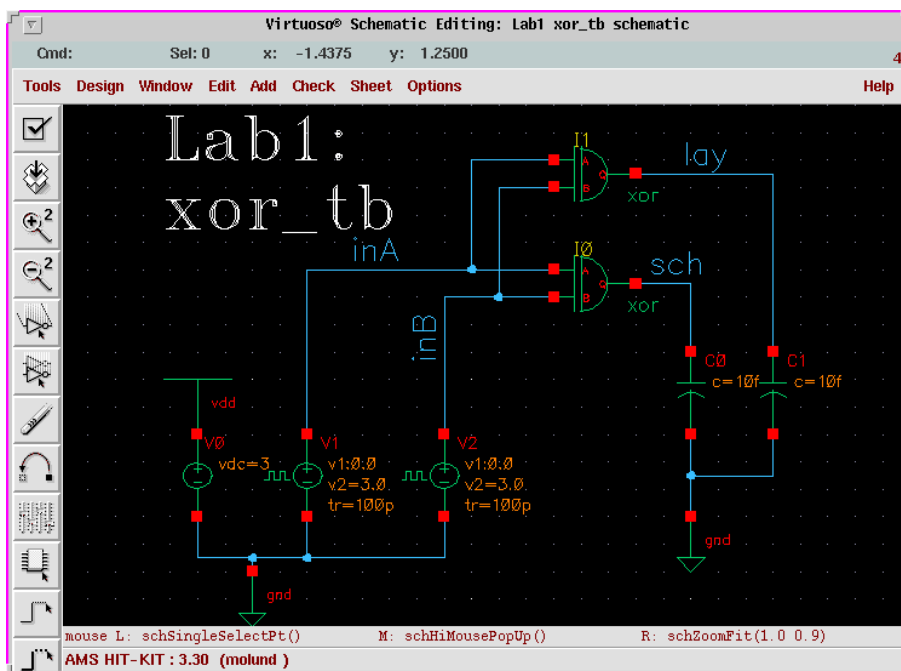


Figure 6.3: A test-bench for post-layout simulation.

Figure 6.3 shows a test-bench intended to simulate an xor-gate. The design contains two instances of the gate which are both fed by the same inputs and are loaded down equally on the outputs. The plan is to simulate one from the schematic and the other from the netlist generated by the extractor, the one with all the parasitic capacitances. Then by plotting both the outputs in the same diagram they could easily be compared.

The configuration file is represented in Cadence as a view with the name **config**. There is a special tool to handle the configuration tasks, which is called the **Hierarchy Editor**. The configuration file is generated by the following procedure:

1. Since it is a new view it is generated by *CIW: File > New > Cellview*. The

Library- and **Cell Name** should read the same as for the schematic of the test-bench but the view must be *config*.

2. Two forms surfaces. In the little one on top **Use Template** is clicked on and **spectre** selected in the form that appears. After an **OK** in the two topmost forms the **Hierarchy Editor** remains.
3. A tree structure presentation is usually preferable and the window is changed by *View > Tree*, after which it should look like the one in fig. 6.4.
4. As soon as it starts, *File > Save* followed by a click on the **Open** button will bring up the schematic view that is connected to the configuration. This is shown by the window title banner of the schematic editor, which now states that it is connected to a configuration. It is necessary to use the correct schematic so that the system can track changes in one window and update the other.

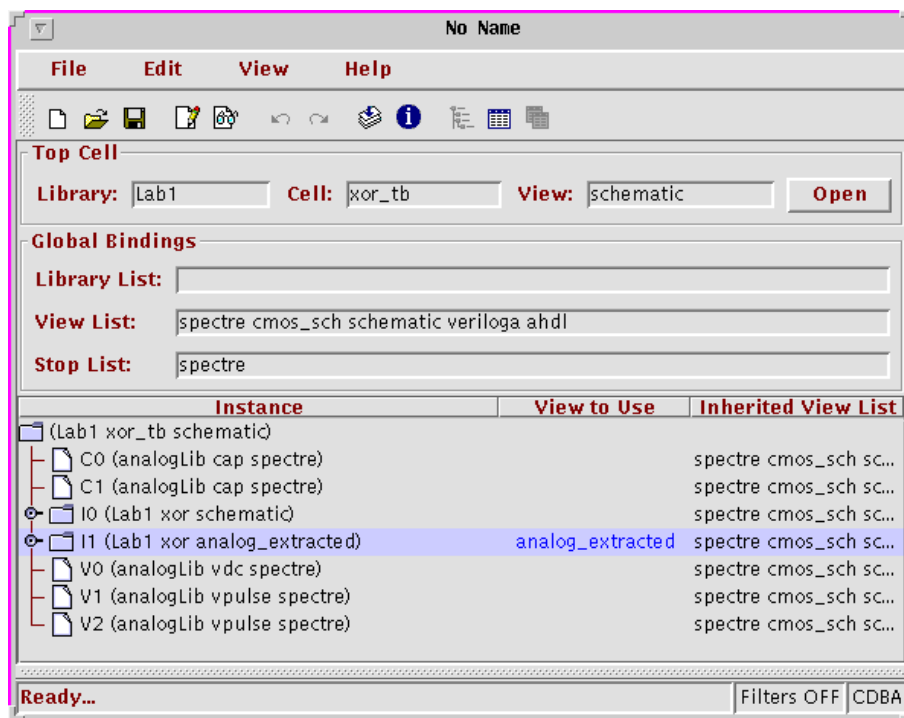


Figure 6.4: The Hierarchy Editor.

An existing schematic with a configuration file should be opened by selecting the config view and using **Open** from the pop-up menu. Then both the **Hierarchy-Editor** and the schematic editor will appear.

In the window it is possible to traverse the hierarchy and see what instances the cells are made of. By clicking on the strange symbol to the far left of a schematic

instance it is opened up and its separate parts shown. The lowest level possible should all have the view **spectre** listed which is what the simulator understands.

By selecting an instance and clicking right a pop-up menu appears. Under *Select View* > the available views are listed. *analog_extracted* should be selected to get the extracted version. By opening the view again the parasitic capacitances can be seen.

After a change the little red ↓↓ in the icon row, just below the *Help* menu, highlights. This means that the editor has detected the change and needs to save the structure. This is done by clicking on the icon and **OK**ing the next form.

6.3.3 Simulation from the Configuration File

The simulation environment are started from the schematic vindow by *Tools* > *Analog Environment*. The config view should be the one listed in the simulator window.

Simulation are then performed just as for the schematic design.