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 methododology 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 laboraties, 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 witch 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 menues. 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.

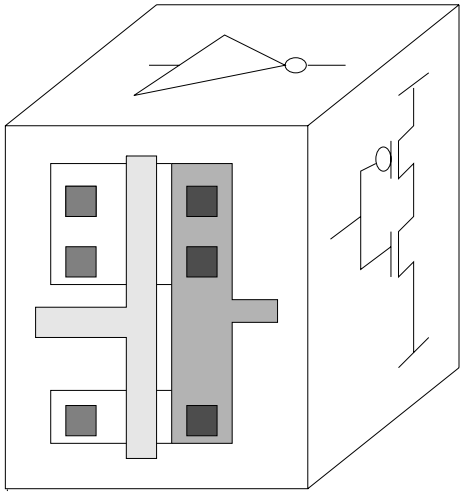

Figure 1.1: An inverter cell with three views: layout, schematic, and symbol.

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 were 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 instanciated 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 instanciated.

Thus by creating more complex structures by instanciating 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 itherefore 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.

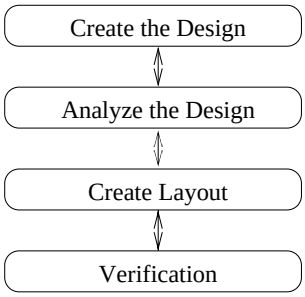

Figure 1.2: The design flow.

The step *Create the Design* consists of drawing schematic views of all cells and blocks. The schematic view contains transitor 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 instanciating 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 vide 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 this figures it is later possible to generate the final mask layers which is used in the fabrication of the design. It is possible to include other cells by instanciating 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 differnt components of the chip is placed on its 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 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 is 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 the compliled (*synthesized*) into a netlist that can be fed into Silicon Ensemble.