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 the 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

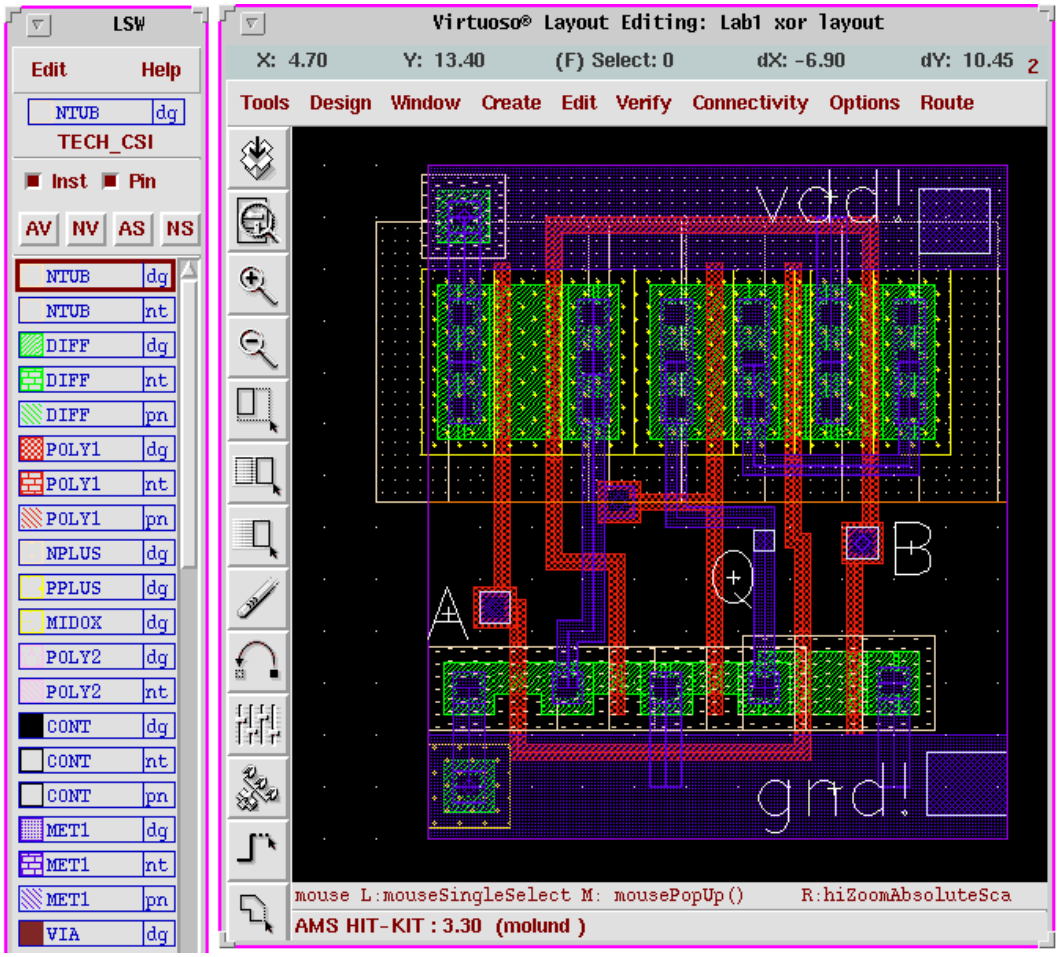

Figure 6.1: The LSW and Layout Editor windows.

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 selctability 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:

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

Instances

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

Paths

The components thus instanciated 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 instanciating 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 coverd 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 convinient 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 differencies.

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 occurences 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 extacted 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.

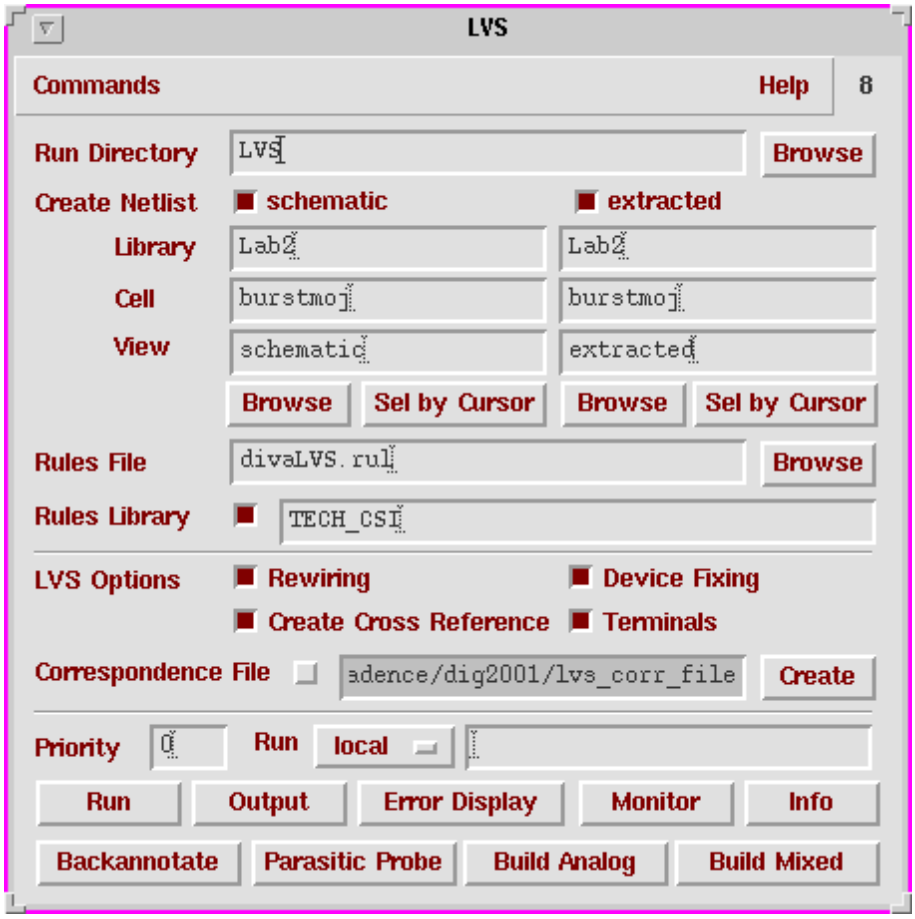

Figure 6.2: The form: Layout Versus Schematic (LVS).

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 funtion 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 comparision 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 configurartion 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.

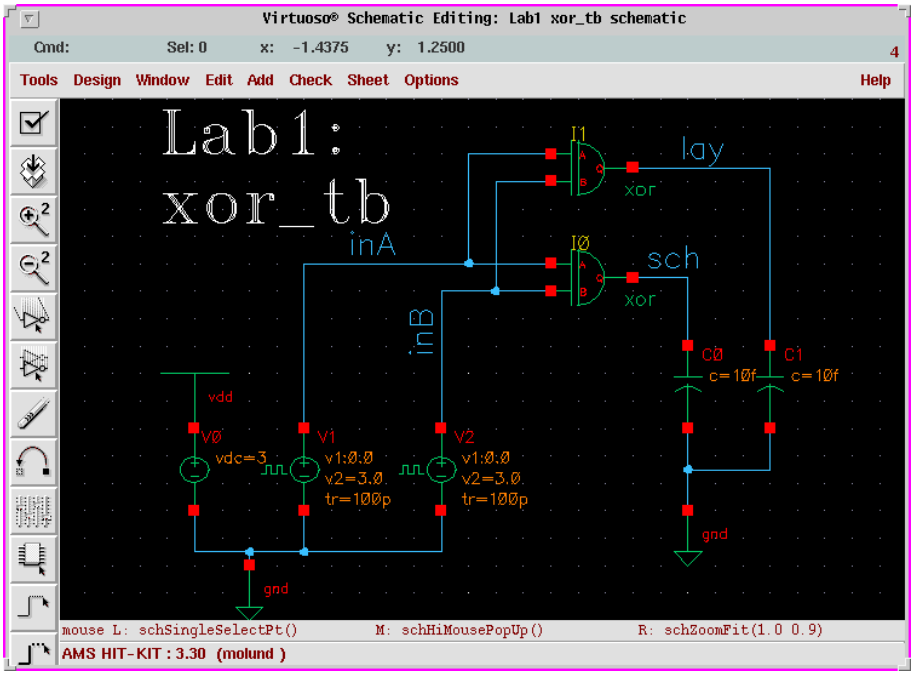

Figure 6.3: A test-bench for post-layout simulation.

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.

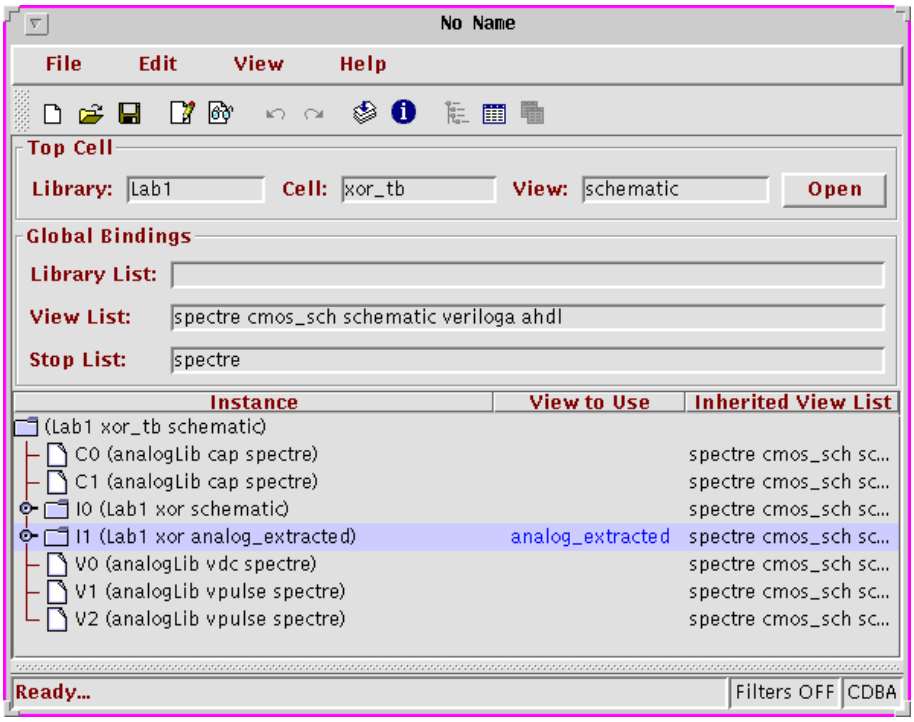

Figure 6.4: The Hierarchy Editor.

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.