Chapter

8

Design Verification with LasiCkt

An important step in the design process is verifying that the layout of an integrated circuit matches the schematics of the integrated circuit. From either the layout or the schematic we should be able to generate a SPICE circuit (a netlist) file and simulate the operation of the IC. A comparison between the netlist and nodelists generated from the layout and schematic of a cell can be used to verify the schematic and layout match. This chapter discusses design verification using the LasiCkt program.

An example chip design, useful in illustrating schematic generation and documentation of layout, is given in the directory C:\Lasi6\W2uchip. It was designed using the MOSIS Scalable CMOS Design Rules (Appendix B) with a λ of 1 μm to give examples of how to generate layouts and schematics for use with LasiCkt. The MOSIS design rules use layers that are slightly different from the layers used with the CN20 setups of Appendix A. For example, to lay out a box of n+ in the CN20 process, we simply select the n+ layer and draw a box, Fig. 8.1a. However, to lay out a box of n+ using the MOSIS design rules, we begin by laying out a box on the "active" layer

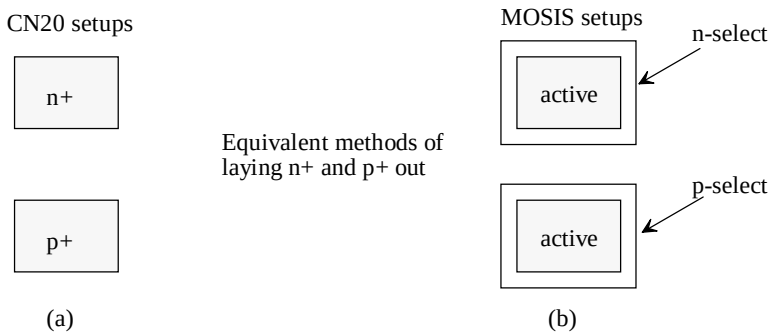

Figure 8.1 Comparison between the layouts of active area using the CN20 and MOSIS setups.

(defines openings in the FOX). To make this active area n+ we draw a box around this area on the n-select (nsel) layer, Fig. 8.1b, (defining an n+ implant). The n-select box must be at least 2 λ bigger than the active box; that is, any edge of active must be at least 2 λ away from any edge of n-select. Similarly, for p+, we draw a box on the active layer, using the MOSIS setups, and surround this box with p-select.

8.1 Fundamentals of LASICKT

The basic drawing of an n-channel MOSFET with connector nodes and labeling is shown in Fig. 8.2. This cell's name is NMOS_SCH and has a rank of 1. (Again the following drawings are available in the C:\Lasi6\W2uchip directory.) The actual symbol of the MOSFET was drawn on the schematic layer (layer 3) using a zero-width path or polygon.

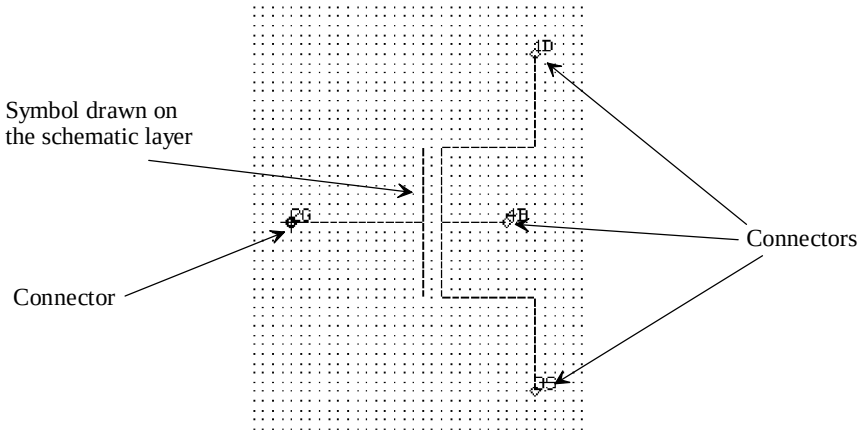

Figure 8.2 Schematic symbol of an n-channel MOSFET.

Specifying Connections to the Symbol

After drawing the symbol, the next step is to label the connections or points on the symbol that will be connected to the wires in the schematic drawing. Text (see problem 1.5) on the connector layer (layer 5) is used to label the connections (sometimes also called pins) to the symbol. The vertex of the text specifies the exact location of the connector. Remember that displaying the vertex can be enabled or disabled by pressing "**t**" on the keyboard followed by executing a **Draw** command.

Layout of the MOSFET

Figure 8.3 shows the layout of two MOSFETs with cell names N3x2 and N15x5 (both with a rank of 1). Again, connector text (on layer 5) is used to specify the connections to the drain, gate, source, and substrate. Notice that the order of the connectors is important; that is, it must correspond to the order of nodes used in SPICE.

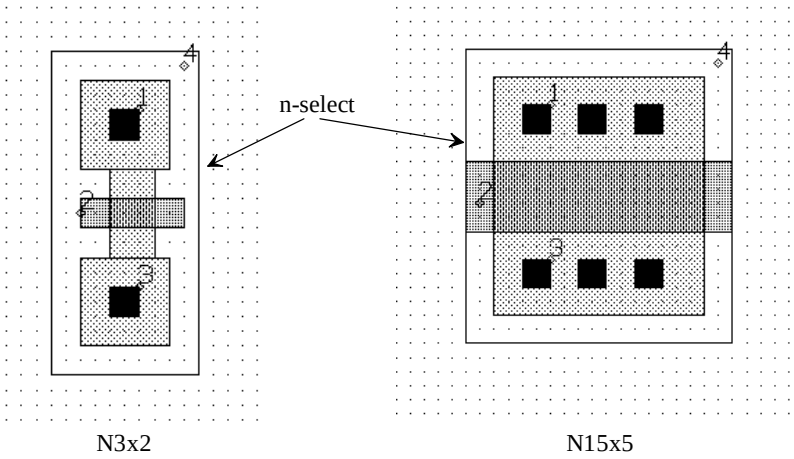

Figure 8.3 Layout with node connections of an n-channel MOSFET.

8.1.1 The Inverter

To help in understanding the process used to extract a circuit file from a schematic or layout, let's consider the design and layout of a basic inverter. Figure 8.4 shows the placement of NMOS_SCH and PMOS_SCH cells into cell INVERT_SCH (rank 2). Notice that an outline of the cell, enabled by pressing "**i**" on the keyboard, is used to show the actual boundaries of the cells. The next step in drawing the schematic of the inverter is to label the device and parameter information. We will label the NMOS M1 and the PMOS M2 using the device text layer (layer 6), as shown in Fig. 8.5. The layouts in Fig. 8.3 correspond to NMOS devices with width to length ratios of 3/2 and 15/5. Since we have a single symbol for an NMOS transistor, the size of the MOSFET

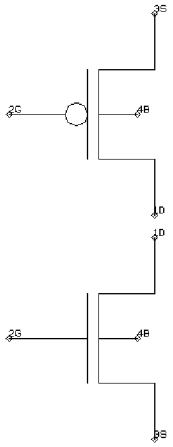

Figure 8.4 Starting the schematic of an inverter.

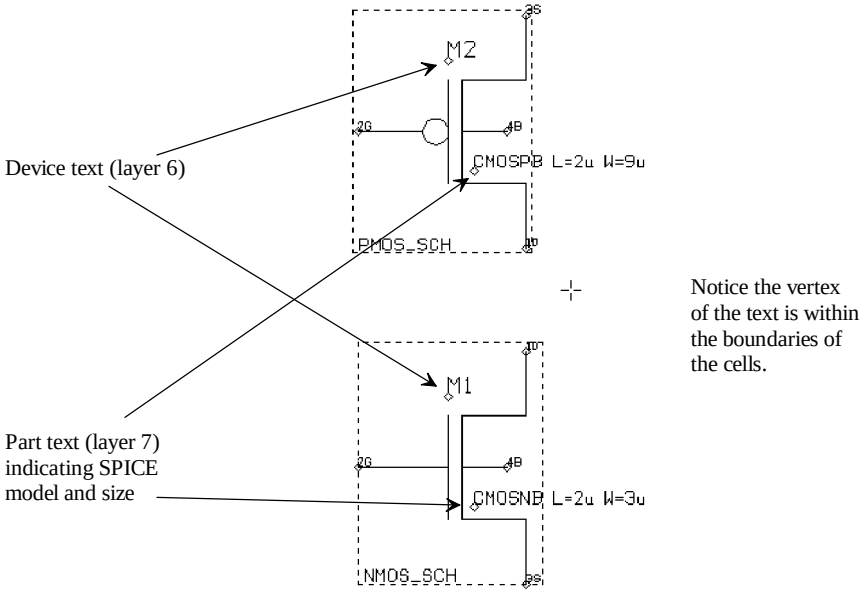

Figure 8.5 Adding device and parameter text to a schematic.

and the model used are specified on the parameter text layers (layer 7). Notice that the vertex of both the parameter (part) text and device text must be placed within the boundaries of the cell.

Labeling the Layout

The layout corresponding to the schematic of Fig. 8.5 is shown in Fig. 8.6. A cell named INVERT with a rank of 2 was created. The cells P9X2, SFRAME, and N3X2 were placed into the INVERT cell. Device and parameter text were used to specify the device name (M1 and M2) and the parameter information (SPICE model, widths, and lengths). It is important that the device name on the schematic correspond directly to a particular device in the layout.

Making Connections

Wires, for use in schematics only, can be drawn on any layer. For the schematics we draw here we will use metal 1 (layer 49). Zero-width paths (which we will call wires) are used to make contact to the device connectors (text on layer 5). We can also label the device connectors directly with a name using node text (layer 4). Figure 8.7a shows how we can label the substrate connection of the NMOS with node 0 (spice ground). A connection is made when a wire crosses the vertex of text written on the connector text layer (layer 5). Figure 8.7b shows a connection between wires. In this figure, the vertex of the wires must coincide for a connection to be made. This enables wires to cross without making a connection. Note that the **fGet** command (around a vertex) was used

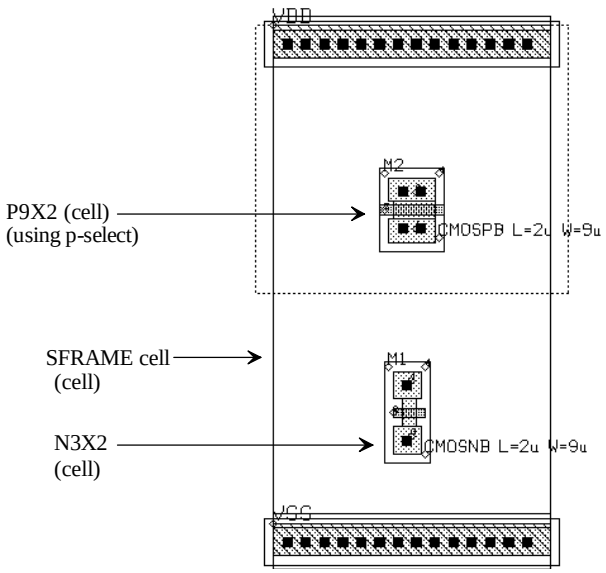

Figure 8.6 Placing the MOSFET cells in the standard cell frame and labeling.

to show the vertices of the wires in this figure. Figure 8.8 shows the schematic of the inverter with wire connections in place and a ground schematic cell added.

The layout of the inverter with interconnections is shown in Fig. 8.9. Poly1, metal1, or metal2 can be used to connect the terminals of the two MOSFETs to *VDD*, ground, the input, and the output. The layer numbers used to specify the interconnect layers, using the MOSIS setups, are 46 (poly1), 49(metal1), and 51(metal2). We might now ask the question, "How does LASICKT know which connections are the inputs,

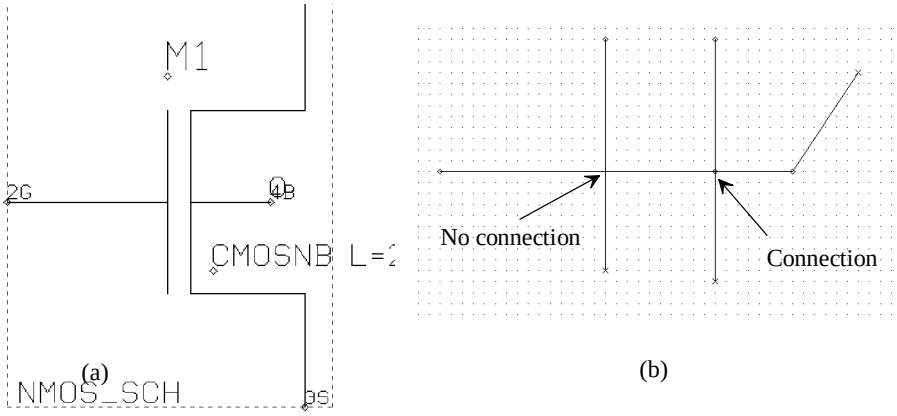

Figure 8.7 How wires make connections.

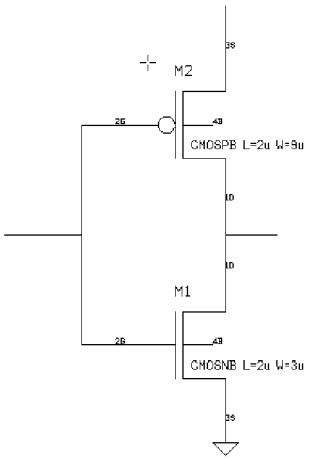

Figure 8.8 Inverter with wire connections.

outputs, and power supplies?" The answer is that we label nodes in the layout and schematic that tell LasiCkt which nodes are the inputs and which are the outputs.

Labeling the Nodes

The next step in creating a schematic or layout is to label the nodes that correspond to inputs/outputs and power supply connections. Figure 8.10 shows that the power supply node is labeled, on the node text layer (layer 4), *VDD*, the input is labeled A, the output

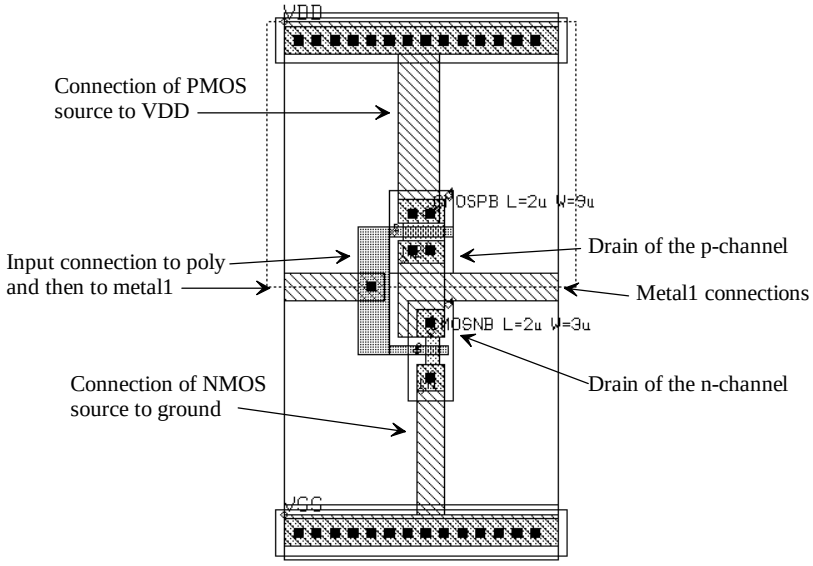

Figure 8.9 Adding connections (poly and metal1) to the layout of the inverter.

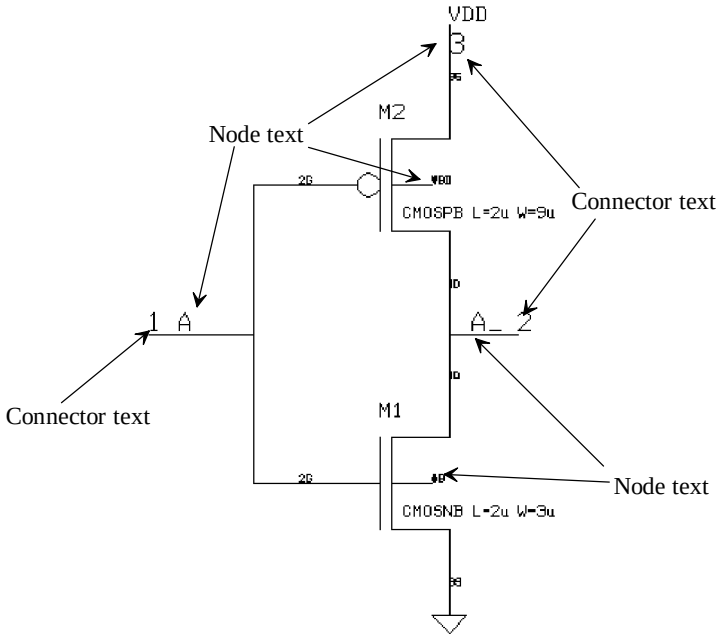

Figure 8.10 Adding node and connector text to the schematic of the inverter.

is labeled A_ and the ground connection is labeled 0 via the ground connector. (Ground is always node 0 in a SPICE circuit file.) It is important to label the input/output nodes, not letting LasiCkt select a name, so that we know how to connect the input voltage or current sources when writing the header file.

In this schematic, we have also labeled the input, output, and *VDD* nodes with connectors (on the connector text layer). Ground was not labeled with a connector since node 0 is universally ground potential. Adding connectors to the inverter allows us to use the inverter as a subcircuit in a higher ranking cell.

The layout of the inverter with labeled nodes and connectors is shown in Fig. 8.11. Notice the direct correlation between the layout and schematic. Both the schematic and layout of the inverter are used in higher ranking cells in the example chip given in the directory C:\Lasi6\W2uchip.

From this layout we can make the following observations.

1. The vertex of the node text must be located on a box or path used for interconnection within a cell. It will have no effect if located on a lower ranking cell. For example, the vertex of the *VDD* node text in Fig. 8.11 must be on the metal1 box connecting the source of the p-channel with the SFRAME (standard frame) cell. Placing it on the metal1 of the underlying SFRAME cell will not label the node.

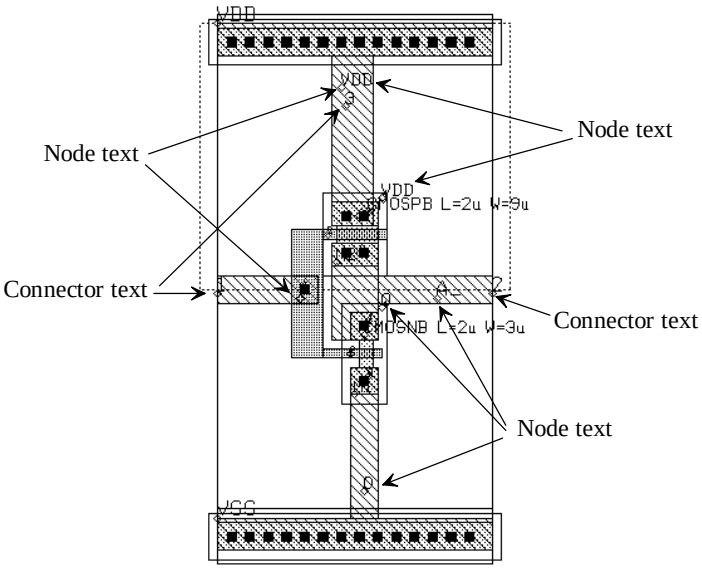

Figure 8.11 Adding node text and connector text to the layout of an inverter.

2. Body connections, substrate or well, must be labeled with a node name separately from the other nodes in the circuit.

3. The vertex of node A in Fig. 8.11 is placed at the intersection of poly1 and metal1. Poly1, metal1, and metal2 are the interconnect layers in a layout. These layers can cross without making contact. Were the layers are connected, using a contact or via, the node text is used to indicate the connection.

4. The vertex of connector text may be placed anywhere in a cell where connection is to be made to the next higher cell. This normally will be where interconnect layers join to the next higher ranked cell.

5. Since connectors are used to specify how cells are connected upward in the cell hierarchy, connectors on a top cell are unnecessary. In fact, they will be flagged as "open connector" errors when LasiCkt compiles. If the cell INVERT was the top cell, there would be no connector text placed on it.

6. There is no node or connector text on the SFRAME cell because it is not an active device. Unless it has an actual SPICE model, it will simply be ignored by LASICKT (unless check for floating cells is enabled).

Adding the additional text to label parameters, device type, nodes, and connectors can clutter a layout or schematic to the point that it becomes unreadable. Using the **View** command button in LASI can limit the number of layers displayed in a drawing.

8.1.2 Running LASICKT

To launch the LasiCkt program, return to the system menu and click on the LasiCkt command button. The screen shown in Fig. 8.12 will appear after selecting the **Setup** button. Filling in the setup information described below, pressing **OK** and then the **Go** button will start **LasiCkt**. If the schematic and layout are drawn using the methods just discussed, LasiCkt will generate SPICE netlist files. The **Comp** command button can be used to compare the schematic and layout netlists (*.CIR files generated with LasiCkt) to test agreement between schematic and layout. The SPICE outputs resulting from the simulation of each circuit file can be compared as well.

LASICKT Inputs

Name of Cell - LASICKT will compile a circuit file based on the contents of the cell specified in this field. The default is the cell that was open the last time LASI was in cell mode.

Header File - Name of file that LASICKT will insert at the beginning of the circuit file. This file generally contains information such as power supply voltages and input sources.

Footer File - Name of file that LASICKT will append to the end of the circuit file. Generally, this file contains the SPICE models and can be common to all cells in the drawing directory.

Interconnect Layers - Layers used to connect the cells together. For the MOSIS setups used in C:\Lasi6\W2uchip layer 46 (poly1), layer 49 (metal1), layer 51 (metal2),

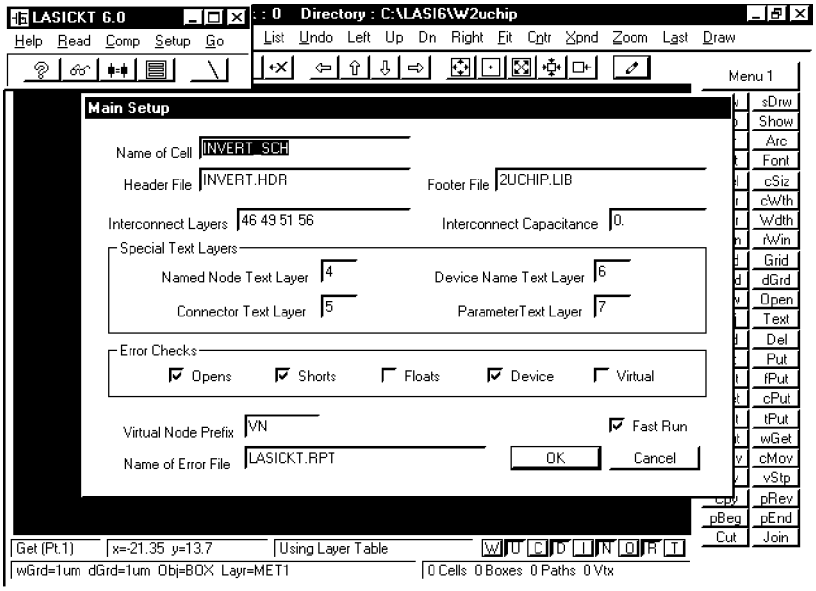

Figure 8.12 LASICKT system screen.

and Layer 56 (poly2) are used. Other layers such as n+ or p+ can be used for connections with the appropriate labeling in the layout.

Interconnect Capacitance - A value that can be used to estimate the capacitance of interconnects based on the area of the interconnection.

Additional information covering the operation or meaning of the inputs used in LasiCkt can be found in the on-line help available by pressing **Help** while at the LasiCkt command menu.

The Header File

The following header files are provided for simulating the operation of the circuits provided in C:\Lasi6\W2uchip.

INVERT	Simple inverter.
COMP1	Comparator with amp and a single buffer output.
COMP2	Comparator with latch and double output buffers.
DFF	D-type flipflop.
NAND	Simple 2-input NAND gate.
OR	Simple 2-input OR gate (NOR+INVERT).
SRFF	SR type flipflop.
TRANGATE	Transmission gate.
WSOTA	Wide-swing operational transconductance amplifier.

The contents of INVERT.HDR are shown below:

```
V1  VDD 0  DC 5V AC 0 0
V2  A 0  DC 0 AC 0 0 PULSE (0 5V 10n 1ns 1ns 50ns 100ns)
.options reltol=0.1 abstol=1u vntol=50mv
.probe
.tran 1ns 150ns
```

The first line in this file specifies the DC voltage source used by the circuit. Note that *VDD* is a node name that corresponds to the labeled nodes in the inverter layout or schematic. The second line is the input pulse to the inverter, discussed in more detail below. The .options statement is used to help with convergence.

.tran 1ns 150ns

This statement is used to specify a transient analysis (the x-axis in probe is time) from 0 to 150ns with a maximum print step of 1ns. Increasing the print step can cause the output to become jagged. Small-print step size can result in a very large output data file (the file used by probe). The full specification for a transient analysis is given by

.tran (print-step) (stop-time) (delay-time) (maximum step size) (UIC)

The delay-time is the time when the data starts being saved to a file. If we wanted to simulate the operation of a circuit from 0 to 100ns but we were only interested in the results from 50ns to 100ns we could use: .tran 1ns 100ns 50ns and only the data from 50 to 100 ns would be saved in the probe data file. The maximum step size will limit the next incremental increase in time as SPICE is simulating the operation of a circuit (and can help with smoothing of SPICE output). Specifying UIC at the end of a .tran statement causes SPICE to use the initial conditions specified in the circuit, for example, the initial voltage across a capacitor.

Pulse Statements

General form:
PULSE(Vl V2 TD TR TF PW PER)

Examples:
VIN 3 0 PULSE(-1 1 2n 2n 2n 50n l00n)

PARAMETER	DEFAULT	UNITS
V1 (initial value)		Volts or amps
V2 (pulsed value)		Volts or amps
TD (delay-time)	0.0	Seconds
TR (rise-time)	TSTEP	Seconds
TF (fall-time)	TSTEP	Seconds
PW (pulse width)	TSTOP	Seconds
PER (period)	TSTOP	Seconds

Piecewise Linear Source

General Form:
PWL (T1 V1 <T2 V2 T3 V3 T4 V4 ...>)

Examples:
VCLOCK 7 5 PWL (0 -7 10NS -7 11NS -3 17NS -3 18NS -7 50NS -7)

Extracting a Circuit File from the Inverter Schematic

Pressing **Go** on the LasiCkt system screen shown in Fig. 8.12 will generate a circuit file, shown below, in the file INVERT_SCH.CIR. A circuit file generated from the cell INVERT and placed into the text file INVERT.CIR results in an identical circuit file to INVERT_SCH.CIR.

```
*** SPICE Circuit File of $INVERT***
* START OF INVERT.HDR
V1  VDD 0  DC 5V AC 0 0
V2  A 0  DC 0 AC 0 0 PULSE (0 5V 10n 1ns 1ns 50ns 100ns)
.options reltol=0.1 abstol=1u vntol=50mv
.probe
.tran 1ns 150ns
.plot tran all
.print tran all
* END OF INVERT.HDR
```

```
* MAIN CIRCUIT
M1 A_ A 0 0 CMOSNB L=2u W=3u
M2 A_ A VDD VDD CMOSPB L=2u W=9u

SPICE models not shown

.end
```

SPICE Simulation Results

The simulation results shown in Fig. 8.13 were generated using the circuit file given above and SPICE3. Since the SPICE circuit file generated using the schematic of the inverter and the layout are the same, the SPICE outputs are the same.

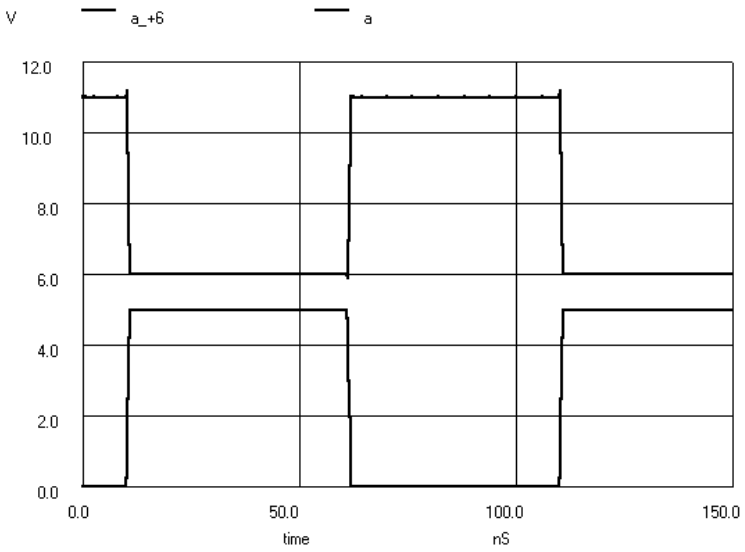

Figure 8.13 Simulation output for the inverter of Figs. 8.10 or 8.11.

8.1.3 Higher-ranking Cells; The OR Gate

The schematic and layout of the inverter can be used in higher ranking cells. To illustrate this, consider the implementaion of an OR gate made using a NOR gate and an inverter. The schematic and layout of the NOR gate with rank 2 (named NOR_SCH and NOR in C:\Lasi6\W2uchip) is shown in Fig. 8.14. These cells were created following the same procedures used to create the inverter cells discussed earlier. SPICE circuit files can be compiled from these cells, keeping in mind that the header file must be changed to NOR.HDR, and the operation of the cells can be simulated.

Creating an OR gate schematic begins with creating a cell, in LASI, named OR_SCH with a rank of three. The INVERT_SCH and NOR_SCH cells are placed into

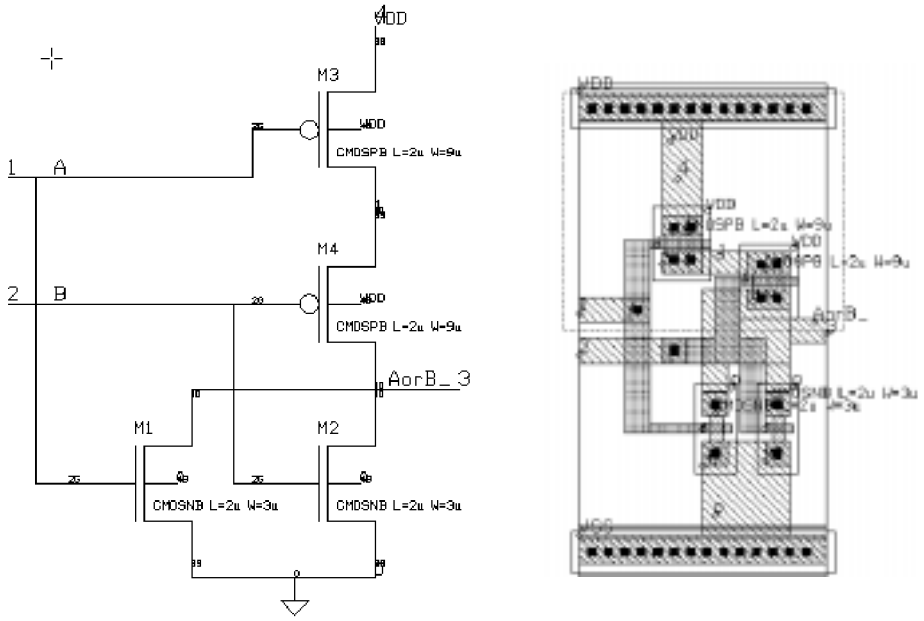

Figure 8.14 Schematic and layout of a NOR gate.

this schematic (Fig. 8.15). Also shown in this figure are the added node and connector names. Figure 8.16 shows this OR gate cell with the NOR and NAND cells drawn as outlines. This figure reveals the location of the text and wires added to the cell. Let's

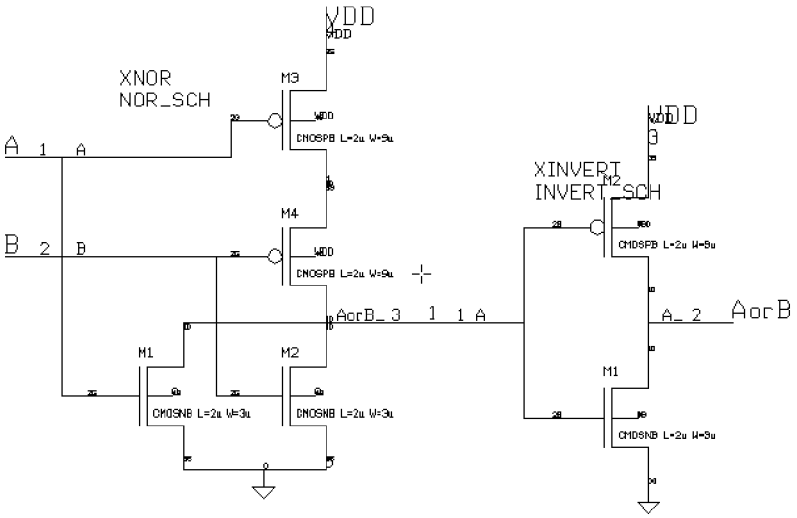

Figure 8.15 Schematic of the OR gate.

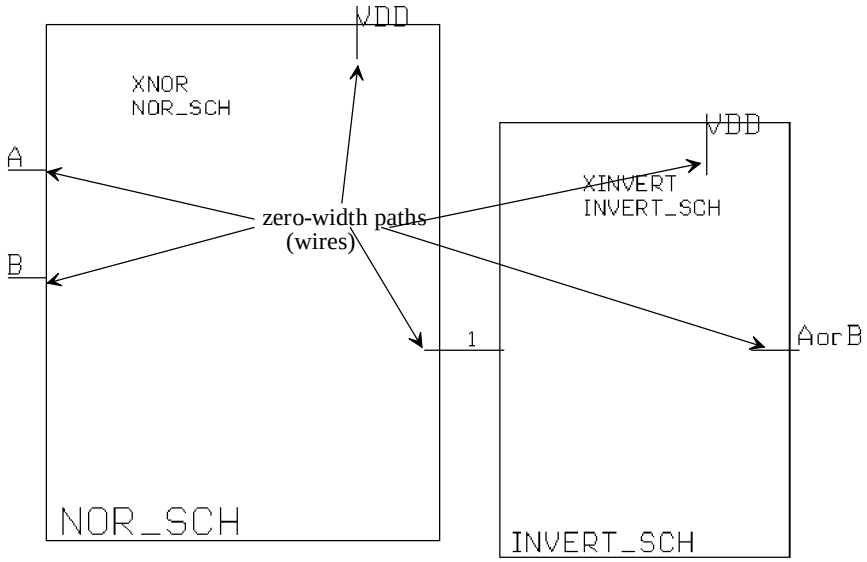

Figure 8.16 The added zero-width paths and text to the OR gate cell.

discuss the labeling shown in this figure, keeping in mind that the first step in generating the OR gate schematic and layout was the addition of the lower ranking cells.

Labeling Devices and Parameters

The "devices" in this cell are labeled XNOR and XINVERT. Since the NOR and the INVERT cells are implemented in SPICE as subcircuits, the first letter in their name must be an X. This is similar to the MOSFET specification that the first letter, when labeling a MOSFET, has to be an M. The vertex of the parameter text is placed within the outline of the cell. The param text is a label telling LASICKT the name of the cell, for this case NOR_SCH and INVERT_SCH.

Labeling Nodes

The most important thing to remember, before labeling any of the nodes in a circuit, is that the vertex of the node text must lie on a box, polygon, or path. *Node text placed on an underlying cell will have no effect unless it is placed exactly on a connector*.

To connect *VDD* to the NOR_SCH and INVERT_SCH cells, we begin by placing a zero-width path on the schematic layer (layer 3) over the vertex of the power connector text. For the NOR_SCH cell schematic, the power supply connection is node 4 on the connector text layer. For the INVERT_SCH cell, the power supply connection is node 3 on the connector text layer. The next step is to place node name text, layer 4, on the added zero-width paths on the schematic layer. As shown in Fig. 8.15, it appears that the source of M3 in the NOR gate cell is labeled twice as *VDD*. This is true. For each level of the schematic in the cell hierarchy, the *VDD* nodes must be labeled.

The next step in the generation of the schematic is to connect the cells together with, and label nodes on, zero-width paths. In this schematic, we have labeled the node connecting the NOR_SCH cell to the INVERT_SCH cell "1." We don't have to label this node. LASICKT will assign a virtual node number if nodetext is not placed on the zero-width wire. Note that the inputs to OR_SCH are labeled A and B, while the output is labeled AorB. Also note that we did not label inputs, output, and power supply with connector text. If we want to use the OR_SCH gate schematic in a higher ranking cell, the connector text must be added.

Labeling the Layout

The layout of the OR gate, with the cells drawn in outline mode, is shown in Fig. 8.17. Notice the added metal1 boxes in the layout. These boxes are placed over the connector text in the lower ranking cells. Note that node 1, in this figure, was not placed on a box; it was placed directly on the connector vertex of both lower ranking cells.

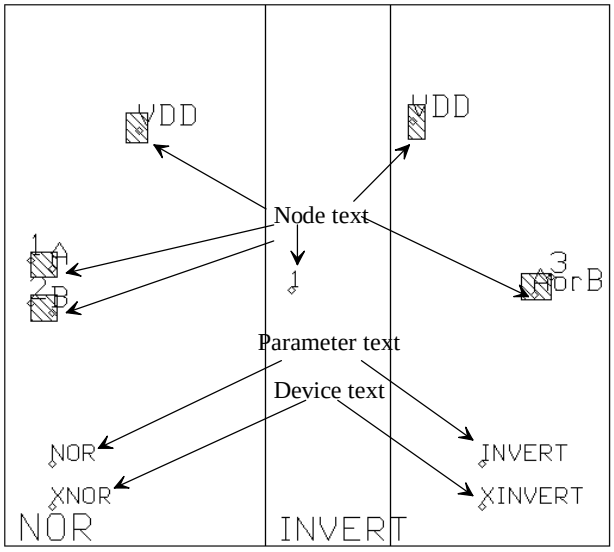

Figure 8.17 Showing the labeling of the OR cell layout.

Simulating the Operation of the OR Gate

LASICKT was used to generate the SPICE circuit file, shown below, using OR.HDR for the header file. The simulation results, using this circuit file, are shown in Fig. 8.18.

```
* START OF OR.HDR
V1  VDD 0  DC 5V AC 0 0
V2  A 0  DC 0 AC 0 0 PULSE (0 5V 5ns 1ns 1ns 50ns 100ns)
V3  B 0  DC 0 AC 0 0 PULSE (0 5V 10ns 1ns 1ns 100ns 200ns)
.options reltol=0.1 abstol=10u vntol=10mv
```

```
.probe
.tran 1ns 200ns
.plot tran all
.print tran all
* END OF OR.HDR

.SUBCKT INVERT A A_ VDD
M1 A_ A 0 0 CMOSNB L=2u W=3u
M2 A_ A VDD VDD CMOSPB L=2u W=9u
.ENDS

.SUBCKT NOR A B AorB_ VDD
M3 1 A VDD VDD CMOSPB L=2u W=9u
M4 AorB_ B 1 VDD CMOSPB L=2u W=9u
M1 AorB_ A 0 0 CMOSNB L=2u W=3u
M2 AorB_ B 0 0 CMOSNB L=2u W=3u
.ENDS

* MAIN CIRCUIT
XINVERT 1 AorB VDD INVERT
XNOR A B 1 VDD NOR

SPICE models not shown
.END
```

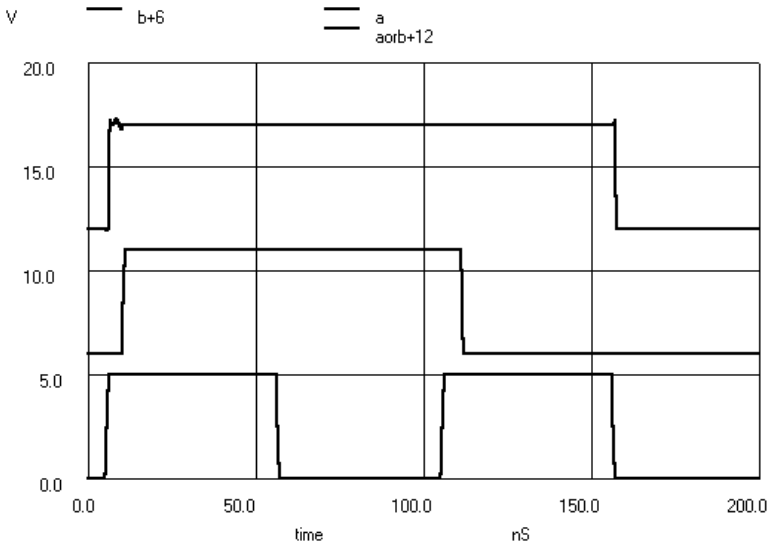

Figure 8.18 Simulation results for the OR gate.

REFERENCE

[1] Boyce, D.E. *LASICKT Help Manual*, Available by pressing Help while running LasiCkt.