Tutorial

2

SAN JOSE STATE UNIVERSITY

Electrical Engineering Department

Bottom up IC design-flow using CDS tools

IC DESIGN GROUP SAN JOSE STATE UNIVERSITY

A tutorial guide for using CDS tools for IC design

David W. Parent
Assistant Professor
Electrical Engineering, SJSU
One Washington Square
San Jose, CA 95192-0084
Phone 408.924.3963 • Fax 408.924.2925

Table of Contents

CHAPTER 1: DESIGN FLOW OVERVIEW 6
- Specification: 7
- Initial Design (Chapter 2): 8
- Schematic Capture (Chapter 3) 8
- Layout (Chapter 3) 8
- Design Rule Checking (Chapter 3) 8
- Circuit Extraction (Chapter 4) 8
- Layout versus Schematic (Chapter 4) 8
- Post Extraction Simulation (Chapter 4) 9
- Fabrication and Test 9

CHAPTER 2: INITIAL DESIGN 10
- Technology Choice: 10
- Logic Minimization: 10
- Transistor Sizing: 10
- Power Estimation: 14
- Summary: 15

CHAPTER 3: GETTING STARTED WITH SCHEMATIC CAPTURE AND SPICE SIMULATION 16
- Setting up your account: 16
- Remote access: 16
- Starting a terminal: 17
- Design entry through schematic capture: 19
- Compiling a new project directory: 20
- Creating a schematic view: 21
- Creating a symbol view: 32
- Creating a test bench: 35
- Simulation in Spectre Spice using the Affirma environment: 40
- Transient Analysis: 41
- DC Analysis: 42
- Adding Wire Names: 44
- Choosing which nets to plot. 45
- Simulation Problems: 47
- Measuring Propagation Delays and V_{INVTH}: 48
- Measuring Power: 50
- Saving your State: 53
- Summary: 54

CHAPTER 4: LAYOUT 55
- Cell Design Tutorial 55
- Preliminary Steps to Layout of the Inverter: 57
  - Floor Planning: 57
  - Sizing the VDD and Ground Lines for Electromigration Failure Prevention: 58
  - Determining the Number of Parallel Transistors to use: 58
  - Summary: 59
- Layout of the Inverter: 61
  - To draw the Ground line: 64
  - To draw the VDD line: 66

Next, we will add the ntaps and ptaps:.... 66
Viewing cells as an outline only. 70
To add the PMOS and NMOS Transistors:.... 71
Adding the Input Layer:.... 73
Wiring up the Transistors with Metal1: 75
Connecting the input to all the gates:.... 77
Adding Pins:.... 79
Design Rule Checking (DRC) 82
Summary: 82
Circuit Extraction: 83
Layout versus Schematic (LVS) 86
Summary: 89
Post Extraction Simulation 90
Summary: 93

APPENDIX A: FURTHER WORK AND AC ANALYSIS 94
Further Work:.... 94
AC Analysis: 96

LIST OF FIGURES:

Figure 1.: Starting a Terminal 17
Figure 2: Terminal 17
Figure 3: Making a directory and listing the contents of a directory. 18
Figure 4: Changing your present working directory 18
Figure 5: Starting CDS tools (icfb-IC Front to Back) 18
Figure 6: CIW 18
Figure 7: Library Manager 19
Figure 8: Creating a Library 20
Figure 9: CIW messages while creating a library 21
Figure 10: Updated library manager showing AMI16 project directory. 21
Figure 11: Creating a new cell view 22
Figure 12: The schematic capture tool. 23
Figure 13: Adding an instance 24
Figure 14: Browsing for components 24
Figure 15: Getting parts 25
Figure 16: Getting vdd and gnd. 25
Figure 17: Getting vdd. 26
Figure 18: Stamping down vdd 26
Figure 19: Getting an nmos4 device W_N=52.2µm. 27
Figure 20: Getting a pmos4 device W_P=93.92µm 28
Figure 21: Major symbols of the CMOS inverter 29
Figure 22: Wiring a CMOS inverter. 30
Figure 23: Add pin pop-up. 31
Figure 24: Changing direction on the Y pin. 31
Figure 25: Completed schematic. 32
Figure 26: Creating a symbol 33
Figure 27: Symbol editor. 33
Figure 28: Importing a symbol. 34
Figure 29: The inverter's symbol. 35
Figure 30: Creating the inverter test bench. 36
Figure 31: Adding the inverter symbol to the test bench. 36
Figure 32: Adding the power supply (V_{DD} from specification). 37
Figure 33: Adding a load capacitor (500f F from specification). 38
Figure 34: Adding a test vector (From specification). 38
Figure 35: The completed test bench 39
Figure 36: Affrima Analog Environemt. 40
Figure 37: Setting up the simulator to use 41
Figure 38: Setting up the model path 41
Figure 39: Setting up the analysis 42
Figure 40: Setting up a DC analysis. 43
Figure 41: Choosing to sweep DC voltage 43
Figure 42: Completed DC Analysis Setup 44
Figure 43: Adding Wire Names. 44
Figure 44: Schematic with Net Names. 45
Figure 45: Schematic with Nets selected for Plotting. 46
Figure 46: Selecting which vectors to plot. 46
Figure 47: Plots of Transient and DC Analyses. 47
Figure 48: Adding a Horizontal Marker. 48
Figure 49: Results from a Horizontal Marker. 49
Figure 50: Final Results. 50
Figure 51: Adding a Power Meter. 51
Figure 52: Schematic with Power Meter Added. 52
Figure 53: Editing the Capacitor's Value. 52
Figure 54: Waveform with Power (19.78mW) 53
Figure 55: Saving the State. 53
Figure 56: Documentation Browser 56
Figure 57: Cell design tutorial 56
Figure 58: Netscape Browser with Tutorial 57
Figure 59: Creating a layout view of the inverter 61
Figure 60: Layout Editor 62
Figure 61: LSW 63
Figure 62: Setting Display Options. 64
Figure 63: Starting a Path 65
Figure 64: Drawing the Ground Line. 65
Figure 65: Inverter with VDD and Ground Lines. 66
Figure 66: Create Instance Pop-Up. 67
Figure 67: Browsing for the ntap. 67
Figure 68: ntap with 7 Rows of Contacts. 68

Figure 69: Placement of the ntap. 68
Figure 70: Ptap Pop-up. 69
Figure 71: Placement of the ptap. 69
Figure 72: Setting the Display Levels. 70
Figure 73: Viewing cells in Outline Only. 70
Figure 74: PMOS Transistor with a Multiple of 5. 71
Figure 75: NMOS Transistor with a Multiple of 5. 72
Figure 76: Placement of the NMOS and PMOS Transistors. 73
Figure 77: Metal1 to Poly Contact. 74
Figure 78: Placement of the Metal1 Poly Contact. 74
Figure 79: Connect Sources of the PMOS Transistor and the ptap.Figure 79. 75
Figure 80: Connect Sources of the NMOS Transistors and ptap. 76
Figure 81: Connecting Drains to form the Output. 77
Figure 82: Connecting up the Gates of the Transistors to the Input. 78
Figure 83: Layout of CMOS Inverter. 79
Figure 84: DRC with no Errors. 79
Figure 85: Pin Creation 80
Figure 86: A input pin. 80
Figure 87: Y Out put pin. 81
Figure 88: Completed INV Layout. 81
Figure 89: Running DRC 82
Figure 90: No DRC Errors. 82
Figure 91: Running the extractor. 83
Figure 92: Selecting Extract parasitic capacitances. 83
Figure 93: Extractor all set to go. 84
Figure 94: Extracted inverter with no errors. 84
Figure 95: Extracted View of CMOS inverter. 85
Figure 96: Running an LVS Check 87
Figure 97: Successful completion of an LVS check. 87
Figure 98: Output of LVS Check. 88
Figure 99: LVS Error Display. 89
Figure 100: Setting for post extraction simulation. 90
Figure 101: Inserting the extracted view. 91
Figure 102: The extracted simulation. 92
Figure 103: Propagation Delay from Extracted Circuit. 92
Figure 104: Power used by Extracted Circuit. 93
Figure 105: Output of Sine Wave Input 95
Figure 106: Setting up vsin for an AC Analysis. 96
Figure 107: Setting up an AC Analysis from 1Hz to 5GHz. 97
Figure 108: Results of a Transient an AC Analysis. 97

Acknowledgements:

This tutorial is based on the NCSU design kit. For more information, see http://www.ece.ncsu.edu/cadence/CDK.html. This tutorial also follows the design flow used by WPI at http://vlsi.wpi.edu/cadence/.

This tutorial was created through a generous grant from CANENCE. Thanks to Sue King, Denny Schneider, Linda Stone, and Tina Fujijara.

Chapter

1

Chapter 1: Design Flow Overview

This chapter will give a brief overview of the bottom-up design flow we will use to design Digital ICs. This tutorial will present the usage of Cadence Tools along with the design of a .5 µm effective channel length CMOS inverter based on the AMI06 process Technology.

This technology can be accessed for classes, senior design projects, and individual research at accredited universities free through MOSIS (www.mosis.org).

We have also developed instruction for getting a design ready for tapeout including such items as proposals, padframes and GDSII conversion.

Specification:

The specification will explain in as much detail as possible the functionality of the circuit, the conditions under which it will operate, and design constraints (for example: speed, area, target yield, and power). The technology for this design is AMIS C5N. The completion date is not specified in this example.

Table 1: Inverter Specification.

Attribute	Specification
Functionality (logic) using static CMOS circuits	Y=not(A)*
Speed	$\tau_{PHL}=\tau_{PLH}$=.1ns Driving a 500f F load†
Area	Cell Height is mandated to be 20 microns. Cell Width is not constrained.
Yield	Yield needs to be high so a minimum of 2 contacts per S/D is required.
Reliability	Needs to be high so electromigration needs to be minimized.
Noise	Would like equal noise margins so $V_{TH}=V_{DD}/2$.
Power	Less than 20mW@ 1GHz driving 500f F load, VDD=5V
Frequency	1GHz
Duty Cycle	50%
Test Vector (DC)	DC SWEEP A 0 to VDD, Measure Y out put voltage. Plot VOUT vs. VIN. V_{TH}.
Test Vector (TRAN)	VPULSE with a period of 1ns, trise=.1ns and tfall=.1ns pulse width equal to .5ns.

* For more complex functionality, a truth table would be used, or the functionality might be described in an HDL (hardware description language) such as VHDL or verilog.

† Propagation delays must always specify a load to be driven since propagation delay is directly proportional to the load capacitance.

Initial Design (Chapter 2):

In the initial design phase, you decide which technology to use, the functionality of your circuit, and the area and power budget. At this point, you would do any logic reduction possible using K-Maps or any other logic technique. You would size the widths and lengths of the transistors using analytical equations for width and length of your transistors. You also draw the circuit schematic out on paper. You develop test vectors to prove that your design will work.

Schematic Capture (Chapter 3)

You enter in the schematic and symbol for your circuit, and create a test bench for simulation. You simulate the circuit to make sure it functions properly, and it meets the time and power specification. Since analytical equations are not as accurate as spice simulation you might need to run change widths of the PMOS and NMOS transistors to get the time and power specifications to simulate properly.

Layout (Chapter 3)

Once the schematic matches your specification in terms of power, timing, and functionality, you draw out how the circuit would look under a microscope. These pictures are used to make the photolithography masks that are used to define your circuit on silicon.

Design Rule Checking (Chapter 3)

Once the circuit is laid out, you have to complete a design rule check to make sure that the circuit will not have yield problems when it is fabricated.

Circuit Extraction (Chapter 4)

Once you have laid out a circuit you need to extract its electrical properties to make sure that you drew the correct functionality and to estimate the parasitic resistances and capacitances that degrade circuit performance. The extracted view can be sent to the simulator.

Layout versus Schematic (Chapter 4)

Once you have an extracted view of your circuit, you need to run a layout versus schematic check. This makes sure that the electrical properties of your schematic match those of your extracted view. This is faster than running all your test vectors on the extracted view.

Post Extraction Simulation (Chapter 4)

Once you are sure that the circuits are equivalent, you run the simulation again using the extracted view. This will take into account parasitic capacitors. You might have to change the widths of your transistors slightly to match your specification. The problem at this point is that you have to change the drawing now to change the transistor and then re-extract the circuit to finally meet your spec. Then you have to go back and change your schematic as well. The better job you do at predicting circuit performance at the initial design stage and schematic capture stage, the less re-work you have to do at the layout stage.

Fabrication and Test

The circuit is sent out for fabrication and comes back. You test it using the test vectors you developed earlier. If the circuit does not meet specification you have to start the process at the beginning using the feedback, you received from the actual devices that were made to fabricate your design.

Chapter 2

Chapter 2: Initial Design

We will choose the technology minimize the logic, and come up with our estimation of the sizes of the transistors in our inverter driver in this section.

Technology Choice:

This is not an easy decision to make. The smallest feature sized process may not be the most cost effective for the design you are trying to create. We choose the AMI06 process because circuit design with this technology in class can be fabricated free. We choose the static CMOS for its low power, ease of design, and low noise characteristics.

Logic Minimization:

Since this circuit is an inverter, no logic simplification is necessary. More complex design examples exist (or are in the process of being created).

Transistor Sizing:

In Table 2 and Table 3, we see the spice parameters for the AMI06 process. We will use a subset of these to design the lengths and widths of our NMOS and PMOS transistors. These are the nominal values and these values can vary by as much as 10% due to process variations. Standard static CMOS is not as prone to fluctuations of these parameters as are analog circuits or Digital circuits that use differential elements. Methodologies for layout to combat these variations will not be covered in this tutorial.

Table 2: Spice Parameters for the NMOS Transistors.

Parameter (unit)	Value
W_N Minimum (cm)	$1.5x10^{-4}$
L_N Minimum (cm)	$0.6x10^{-4}$
TOX (cm)	$.014x10^{-4}$
CGDO (F/cm)	$1.99x10^{-12}$
CGSO (F/cm)	$1.99x10^{-12}$
CJSW (F/cm)	$3.825632x10^{-12}$
CJ (F/cm^2)	$4.233802x10^{-8}$
VT (V) @ Ln $=0.6x10^{-4}$	.6
KNP A/V^2	$46.3x10^{-6}$
Length of Source or Drain (cm)	$1.5x10^{-4}$

Table 3: Spice Parameters for the PMOS Transistors.

Parameter (unit)	Value
W_P Minimum (cm)	$1.5x10^{-4}$
L_P Minimum (cm)	$0.6x10^{-4}$
TOX (cm)	$.014x10^{-4}$
CGDO (F/cm)	$2.4x10^{-12}$
CGSO (F/cm)	$2.4x10^{-12}$
CJSW (F/cm)	$3.114708x10^{-12}$
CJ (F/cm^2)	$7.273568x10^{-8}$
VT (V) @ $L_P = 0.6x10^{-4}$	-.82
KNP A/V^2	$30x10^{-6}$
Length of Source or Drain (cm)	$1.5x10^{-4}$

To size the NMOS and PMOS transistor lengths we will choose the minimum length due to that fact that area and power are a concern. If reliability or process variation (V_T vs. Channel length for instance) were a concern, we could increase the channel length to compensate.

We will use the following equations to size the widths of the NMOS and PMOS

transistors: $$W_N := \frac{\left(C_L + C_{JSWN} \cdot 4 \cdot L_D\right)}{\left[\frac{\tau_{PHL}}{L_N \cdot A} - (1 + Ratio) \cdot \left(C_{JSWN} \cdot 2 + C_{JN} \cdot L_D\right)\right]} \quad (1)$$

$$A := m1 \cdot \tau_{PHL} + b_1 \quad (2)$$

$$Ratio := m2 \cdot \tau_{PHL} + b_2 \quad (3)$$

$$W_P := Ratio \cdot W_N \quad (4)$$

Table 4 shows the values and explanation of the variable used in equations 1-4. Equations 1-4 are accurate to with in 12% or less when compared to spectre spice simulation, and have been tested with C_L ranging from 50f F to 10pF and propagation delays from .1ns to .5ns

To find W_N and W_P:

1. Solve for A using equation 2. A=12300Ω in this example.
2. Solve for the Ratio using equation 3. Ratio=1.8 in this example.
3. Solve for W_N using equation 1. W_N=52.2μm.
4. Solve for W_P using equation 4. W_P=93.9μm.

Note 1: These values are our first estimate. To verify our design we still have to test the inverter with a more accurate model in schematic capture.

Note 2: We need to estimate the power of this circuit to see if it will meet our power specification before we begin working with the Cadence tools.

Note 3: The parameter A is for a gate length of .6μm. If you change the gate length, then V_T will change and as a result A and Ratio will not be as accurate.

Table 4: Explanation of terms for CMOS design.

Parameter	Description	Value	Units
W_N	Width of the NMOS	Design to Spec	cm
W_P	Width of the PMOS	Design to Spec Use Equation (4)	cm
L_N	Length of NMOS	$.6x10^{-4}$	cm
C_L	Load Capacitance	500f	F
C_{JSWN}	NMOS sidewall Capacitance	$3.825632x10^{-12}$	F/cm
C_{JN}	NMOS junction Capacitance of Drain bottom	$4.233802x10^{-8}$	F/cm^2
L_D	Length of the Drain	$1.5x10^{-4}$	cm
τ_{PHL}	Delay measured from $V_{DD}/2$ of input to $V_{DD}/2$ of output during the high to low transition of the output	In this case .1n	s
A	Fitting parameter used to design W_N	Use equation (2)	Ω
Ratio	Used to scale W_P to give symmetric propagation delays	Use equation (3)	unit less
m_1	Fitting parameter (slope) to find A	$-5x10^{12}$	Ω/s
m_2	Fitting parameter (slope) to find Ratio	$-250x1^{06}$	Hz
b_1	Fitting parameter (y intercept) to find A	12800	Ω
b_2	Fitting parameter (y intercept) to find Ratio	1.825	unit less

Power Estimation:

We will use the following equations to estimate the power used by our circuit:

$$C_{DTOTAL} := C_{JSWN} \cdot 2 \cdot \left[W_N \cdot (1 + \text{Ratio}) + 2 \cdot L_D \right] + C_{JN} \cdot W_N \cdot L_D \cdot (1 + \text{Ratio}) \quad (5)$$

$$C_{TOTAL} := C_L + C_{DTOTAL} \quad (6)$$

$$\text{Power} := \alpha \cdot f \cdot C_{TOTAL} \cdot V_{DD}^2 \quad (7)$$

Table 5 gives the details of parameters used in equations 5-7.

$\text{Power} = 17.67 \times 10^0 \ \text{m} \cdot \text{W}$ in this example, so we are with in specification and may proceed with the rest of the design.

Table 5: Explanation of terms for CMOS design.

Parameter	Description	Value	Units
C_{DTOTAL}	Total NMOS and PMOS drain capacitance	Use equation	F
C_L	Load Capacitance	500f	F
C_{TOTAL}	Total Capacitance the inverter must drive	Use equation	F
f	Frequency	1G	Hz
α	Activity Factor (A clock has an activity factor of 1)	1	unit less
V_{DD}	Supply Voltage	5	V
W_N	Width of the NMOS	$52.2x10^{-4}$	cm
W_P	Width of the PMOS	$93.9x10^{-4}$	cm
L_N	Length of NMOS	$.6x10^{-4}$	cm
C_{JSWN}	NMOS sidewall Capacitance	$3.825632x10^{-12}$	F/cm
C_{JN}	NMOS junction Capacitance of Drain bottom	$4.233802x10^{-8}$	F/cm^2
L_D	Length of the Drain	$1.5x10^{-4}$	cm

Summary:

We have introduced some equations for sizing the widths and lengths of the inverter example as well as estimating power consumption. We have come up with an initial design and verified that our circuit will meet the timing and power portions of our specification. Next, we will verify the timing and power consumption of our circuit using the schematic capture, and spectre spice simulation tools.

Chapter

3

Chapter 3: Getting started with Schematic Capture and Spice Simulation

Now that we have estimated W_N, W_P and power of our inverter we need to verify the design using a more sophisticated model.

This Chapter will go over:

1. Setting up your account
    a. Design Entry through schematic capture (input W_N and W_P)
    b. Create INV schematic capture
    c. Create INV symbol
    d. Create INV Test bench (use our test vectors indicated in the specification)
2. Simulation using CDS's Spectre (Bsim3 model simulator, different underlying algorithms than spice or hspice)

Setting up your account:

Your account should have the all paths set to run the software. You need only to log in and start a terminal. There should be a terminal icon on your Common Desktop Environment. Just double click on it and a command window will appear (Figure 2). You type commands in this window just like an MSDOS command line except that the commands are different.

Remote access:

Instructions on remote access can be found at:
http://www.engr.sjsu.edu/~dparent/ICGROUP/UNIX.pdf

Starting a terminal:

To start a terminal right click in the middle of the screen of the CDE. While holding the right mouse button down move, the mouse down until tools is highlighted, and then move the mouse over to tools and highlight terminal. (This should look like Figure 1) Let up the right mouse button and a terminal should start.

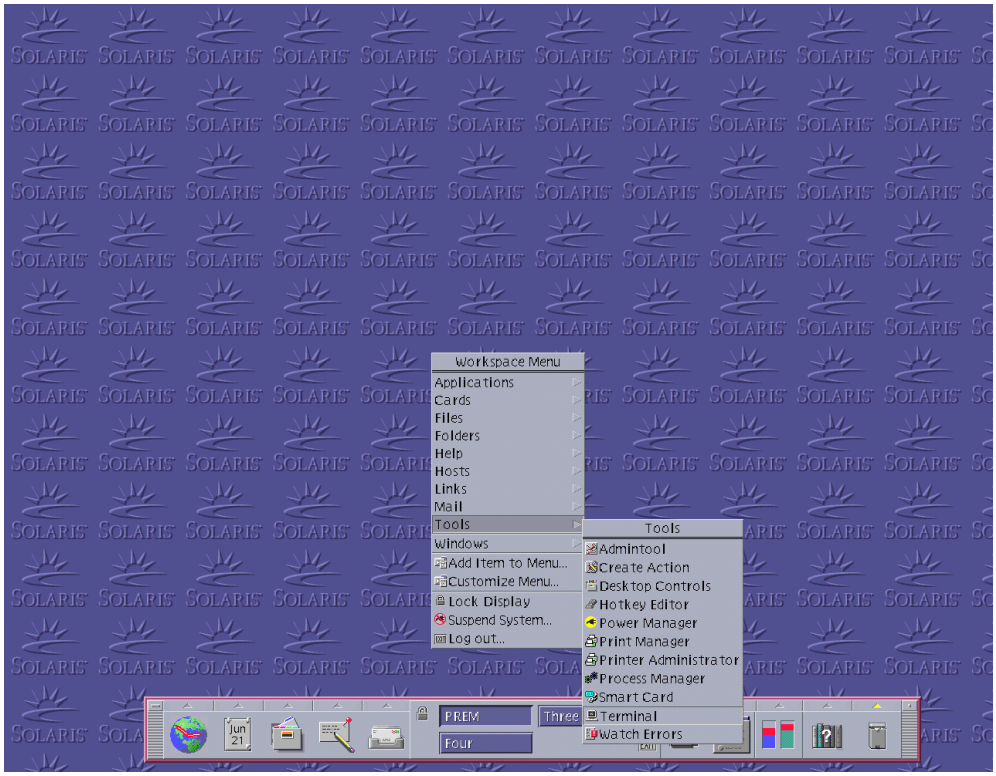

Figure 1.: Starting a Terminal

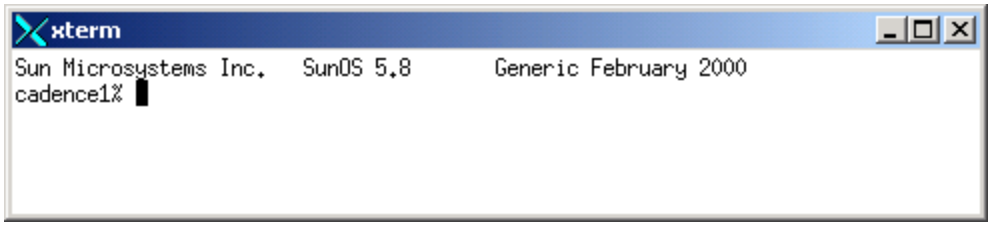

Figure 2: Terminal

After you have started the csh, you need to create a project directory. To do this, type the command **mkdir cell** at the command line. This command makes a directory cell in your home directory. **You only have to create this directory once!** Type the command **ls**. You should get

the result cell as in Figure 3. The command ls lists out the content of your present working directory.

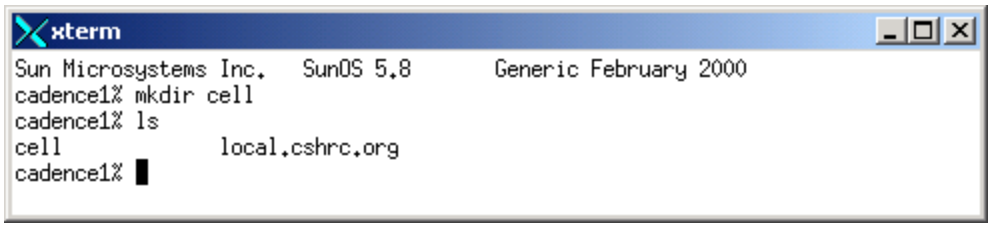

Figure 3: Making a directory and listing the contents of a directory.

In order to start CDS tools so that your project files are available you need to change into your project directory before starting the tools. Type in the command **cd cell** at the command line. This changes your present working directory to cell. If you type the command **ls**, you should see no files (Figure 4).

Figure 4: Changing your present working directory.

To start CDS tools type in the command **icfb &**. You should see messages similar to Figure 5. After some time, the CIW (Command Interface Window) will pop up (Figure 6). Once the CIW come up you will not need to use the command line.

Figure 5: Starting CDS tools (icfb-IC Front to Back)

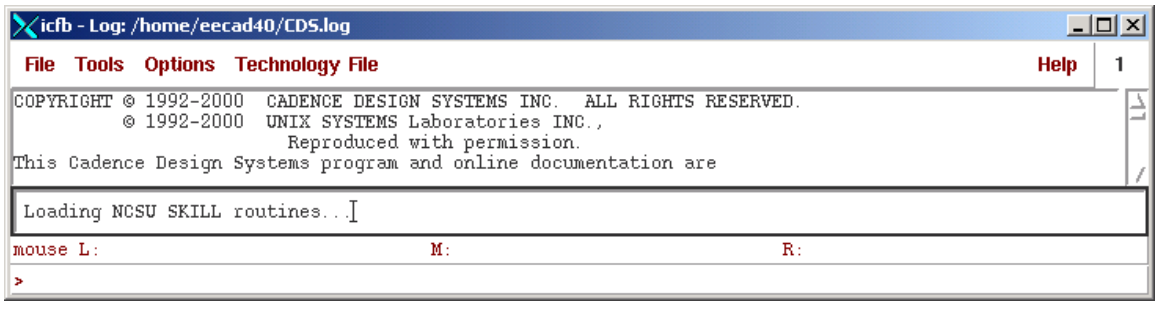

Figure 6: CIW

Design entry through schematic capture:

This is the beginning of the design flow as far as using CDS tools is concerned. Prior to starting the tools you need to have define your logic function, decided on an implementation, reduced the number of gates, and done some initial sizing of the lengths and widths of your transistors.

Now that you have created your project directory, whenever you want to work on the design in that directory you just have to:

1. Log in
2. Start a terminal
3. Type in the command**, cd cell** (or the name you selected for your project)
4. Type in the command, **icfb &**.

We will be using the NCSU design kit, which automatically starts the library manager (Figure 7). You should see three NCSU libraries and a library named basic and cdsDefTechLib. There is also a sample directory called ANALOG and PADFRAME.

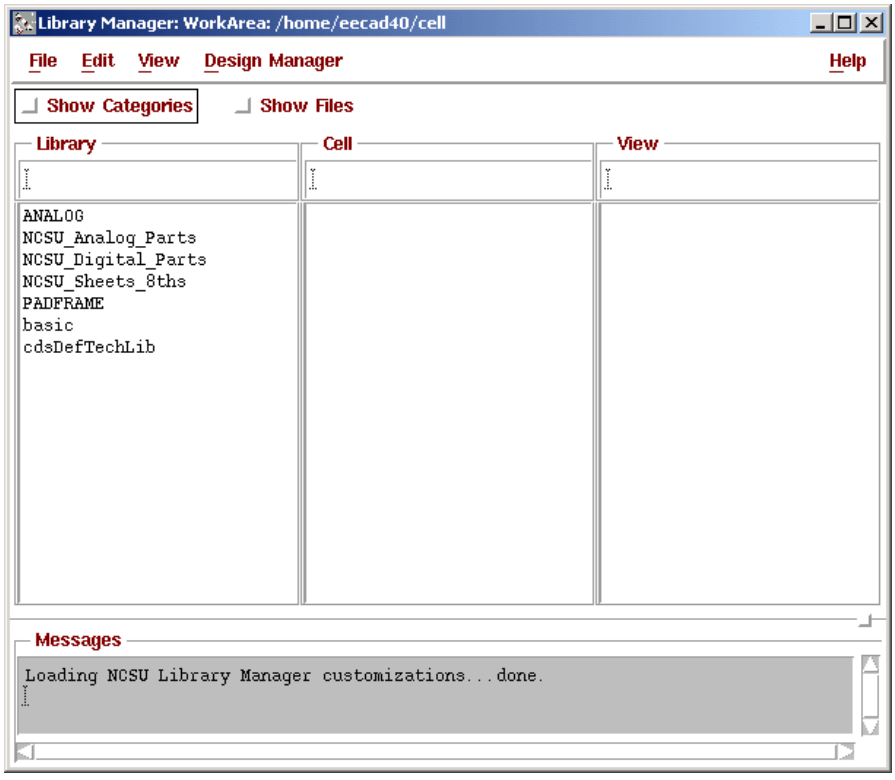

Figure 7: Library Manager

Compiling a new project directory:

We need to create a project library and compile a technology library to it. The tech library contains all the process specific information needed to design an IC using a particular fabrication house's process. We will be using AMI 0.6u C5N (3M, 2P, High Res) process.

To compile your new project library:

1. Goto the library manager popup and goto: file… New… library. A pop-up like Figure 8 should appear.

2. Fill out the pop up exactly according to Figure 8. Make sure to click on compile tech library.

3. Click OK. You should see messages in the CIW similar to the ones in Figure 9.

4. Your library manager should now show your new project directory as in Figure 10. Make sure that your library has all the components listed in column 2 for Figure 10.

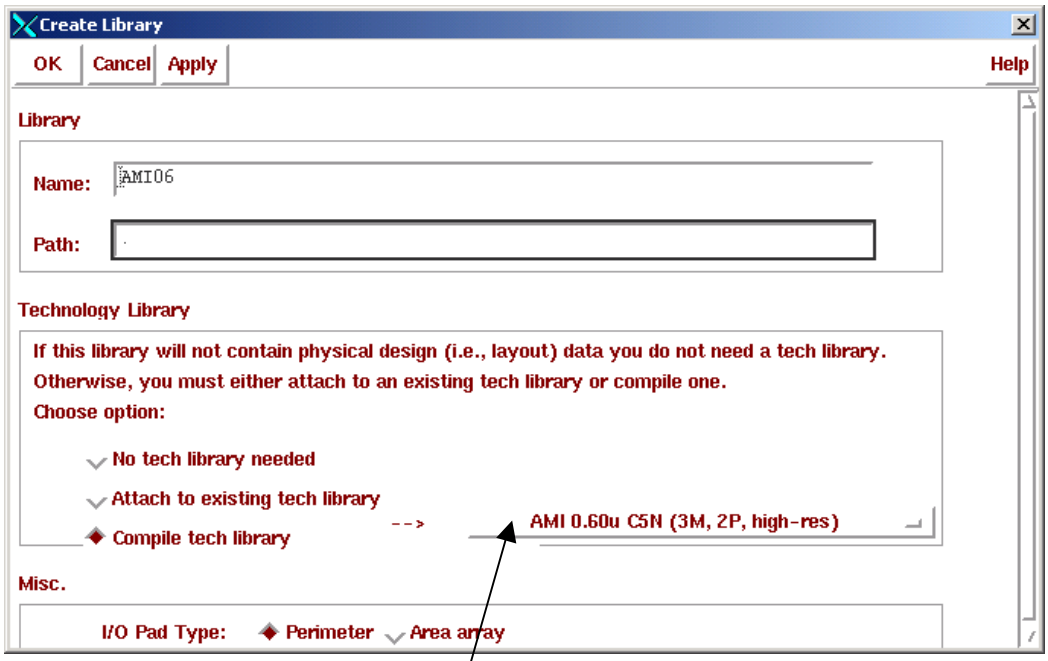

Figure 8: Creating a Library

Make sure you set the tech library to AMI0.6u and not AMI16u. You will have to redo your tutorial if you do this incorrectly!

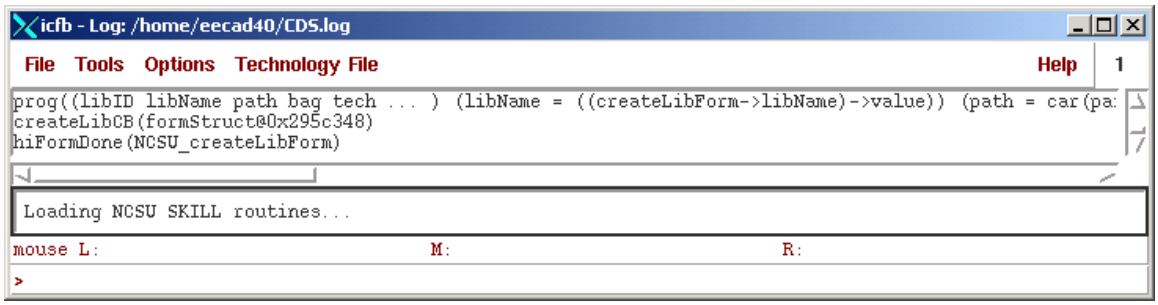

Figure 9: CIW messages while creating a library

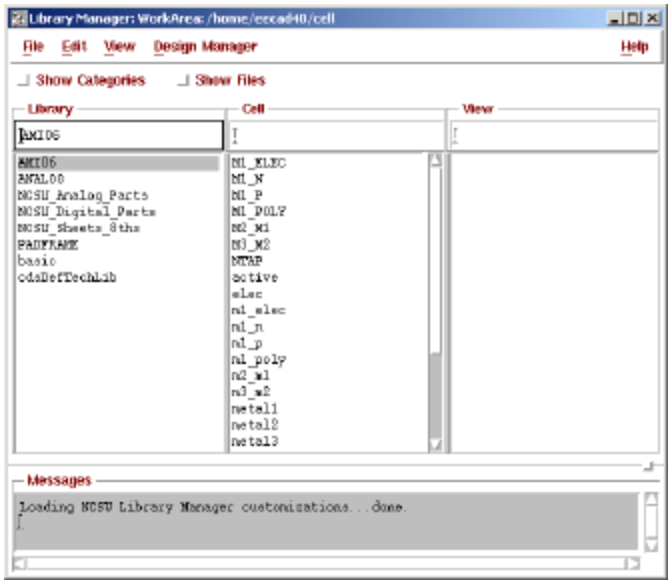

Figure 10: Updated library manager showing AMI16 project directory.

Creating a schematic view:

After creating the library AMI06, we can now start adding our circuit views. Each circuit will have different views. For example, our inverter will have a schematic, symbol, layout, and extracted views. There are more views than these, but these are the only ones we will use in this course.

1. Schematic view: The circuit is described by electronic symbols connected by wires. A spice run deck can be compiled directly from this view. For example, an inverter would have two transistors, a power supply, ground, and import and output ports.

2. Symbol view: The circuit is described by one symbol. For example, an inverter would look like the Boolean logic symbol for an inverter.

3. Layout: The circuit is drawn, as it would appear if you looked at in under a microscope. Each layer corresponds to a process step.

4. Extracted: CDS tools can extract a schematic from the drawn layers. This is helpful to make sure you drew the circuit so that it will perform like the schematic. The extracted

view will include parasitic capacitances and resistances from the drawn layers that the schematic view cannot generate.

To create a schematic view:

1. Goto the library manager popup and goto: file… New… cell view. A pop-up like Figure 10 should appear.

2. Fill out the pop up exactly according to Figure 11.

3. Click OK. The schematic entry tool should appear (Figure 12).

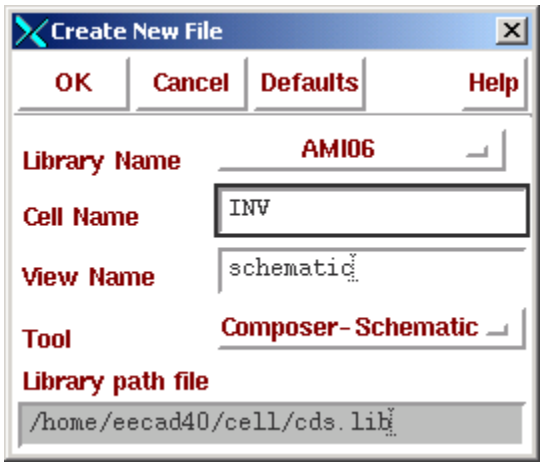

Figure 11: Creating a new cell view

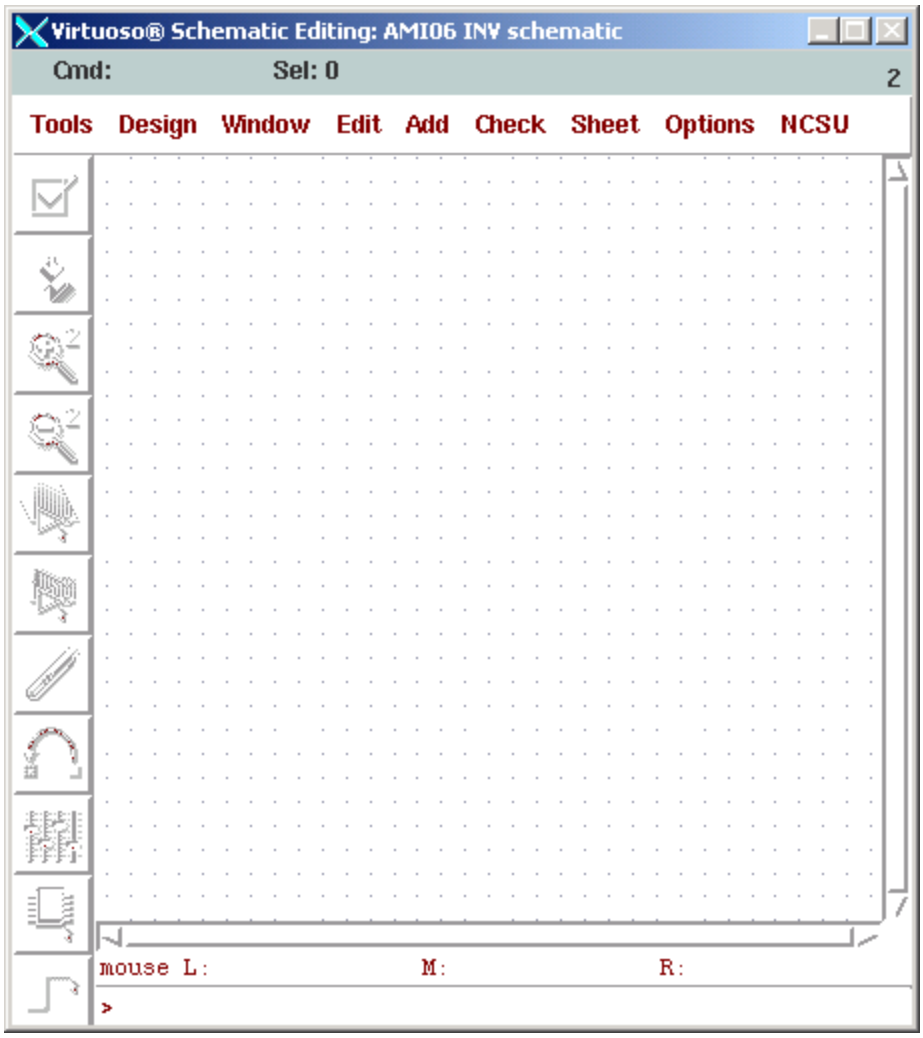

Figure 12: The schematic capture tool.

The schematic capture tool will be used to draw the schematic representation of your inverter.

To draw the inverter you need to add a pmos4 transistor, nmos4 transistor, ground connection, power supply connection, input pin, output pin, and wire it together. Use the lengths and widths calculated in Chapter 2.

To add items to your schematic goto add… instance in the composer tool, or press the letter i. The add instance pop-up appears like in Figure 13.

Click on browse to graphically get components. A pop-up like Figure 14 should appear.

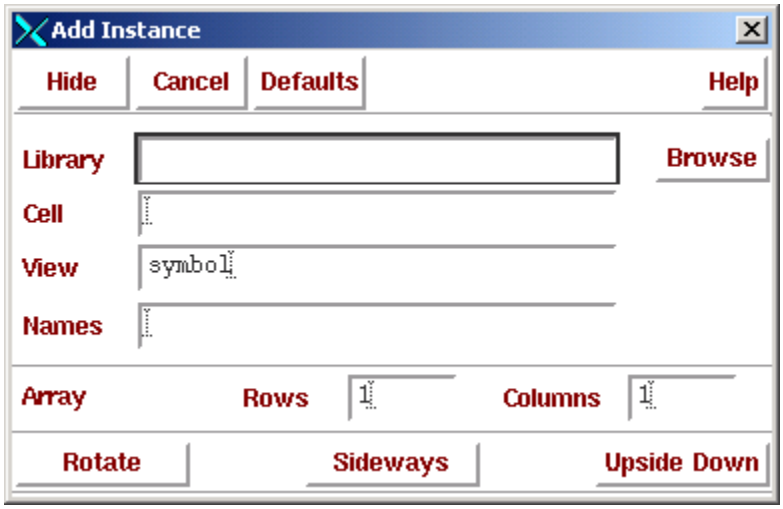

Figure 13: Adding an instance

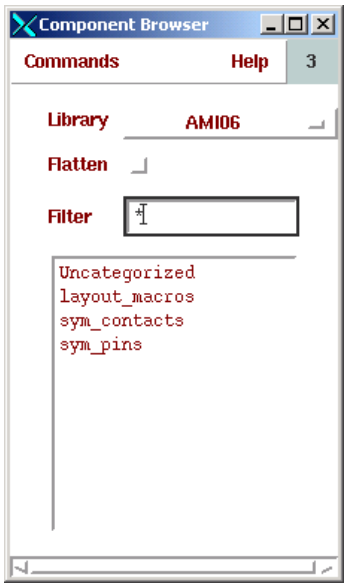

Figure 14: Browsing for components

To get the parts we need change the library according to Figure 14.

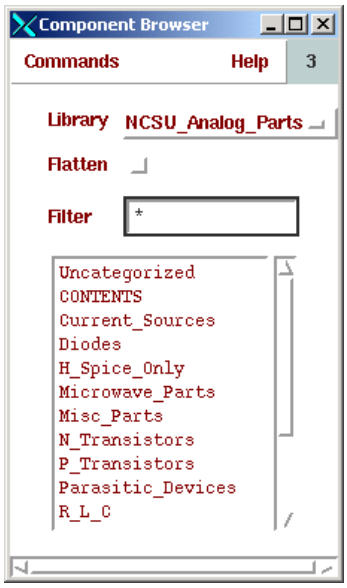

Figure 15: Getting parts

Click on supply nets and the pop-up should look like Figure 16. The items in supply nets are actually global signals. Global signal are automatically given pins. This makes our symbols cleaner because we only show logic ports.

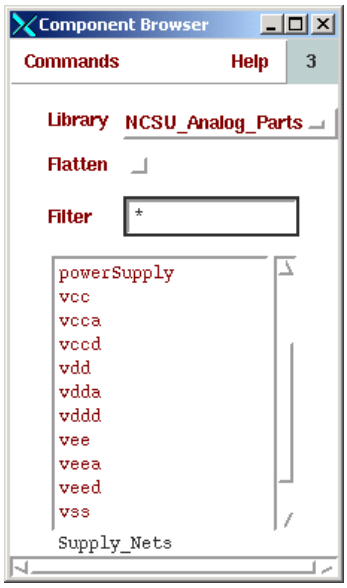

Figure 16: Getting vdd and gnd.

The add instance pop-up should look like Figure 17.

Stamp it down in your schematic like in Figure 18.

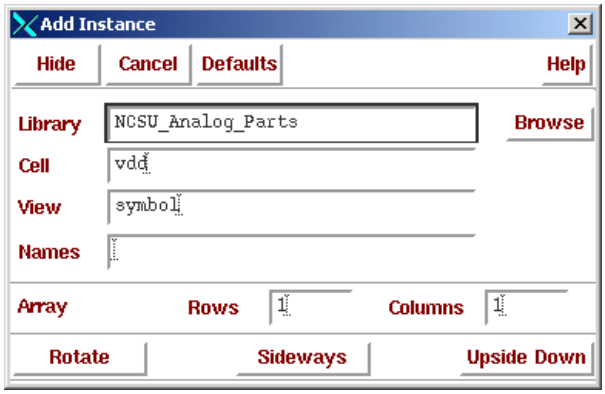

Figure 17: Getting vdd.

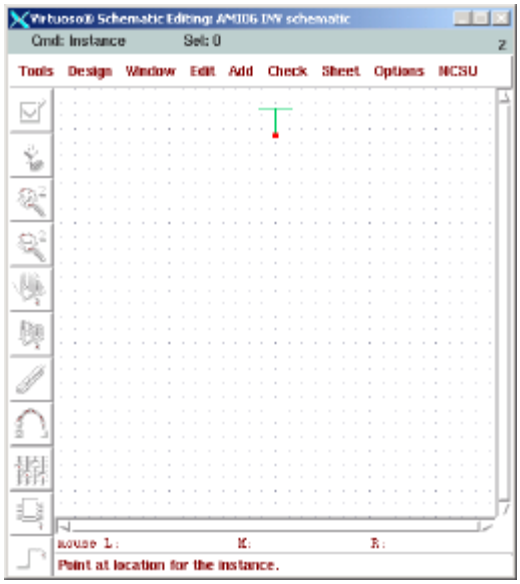

Figure 18: Stamping down vdd.

Get the gnd symbol and stamp it down in the same manner as you did for vdd.

To add the transistors click on N_Transistors for the nmos4 and P_Transistors for the pmos4. Add an nmos4 and a pmos4 transistor in a similar manner. The pop-up for the nmos4 should look like Figure 19, and the pmos4 pop-up should look like Figure 20**. NOTE: If your transistor does not have the widths and lengths like in Figure 20 you did not compile the tech library correctly. You need to start over!**

Use the W_N and W_P values from chapter 2!

Stamp them down like Figure 21. It does not have to be exact but neatness will help further.

If you make a mistake and need to get out of add instance mode, press the esc key.

Click on the object you want to delete and press the del key.

If you cannot enter in the width exactly as shown or it does not say ami06N under model name, you have not complied the directory correctly. You will have to delete this directory and start over!

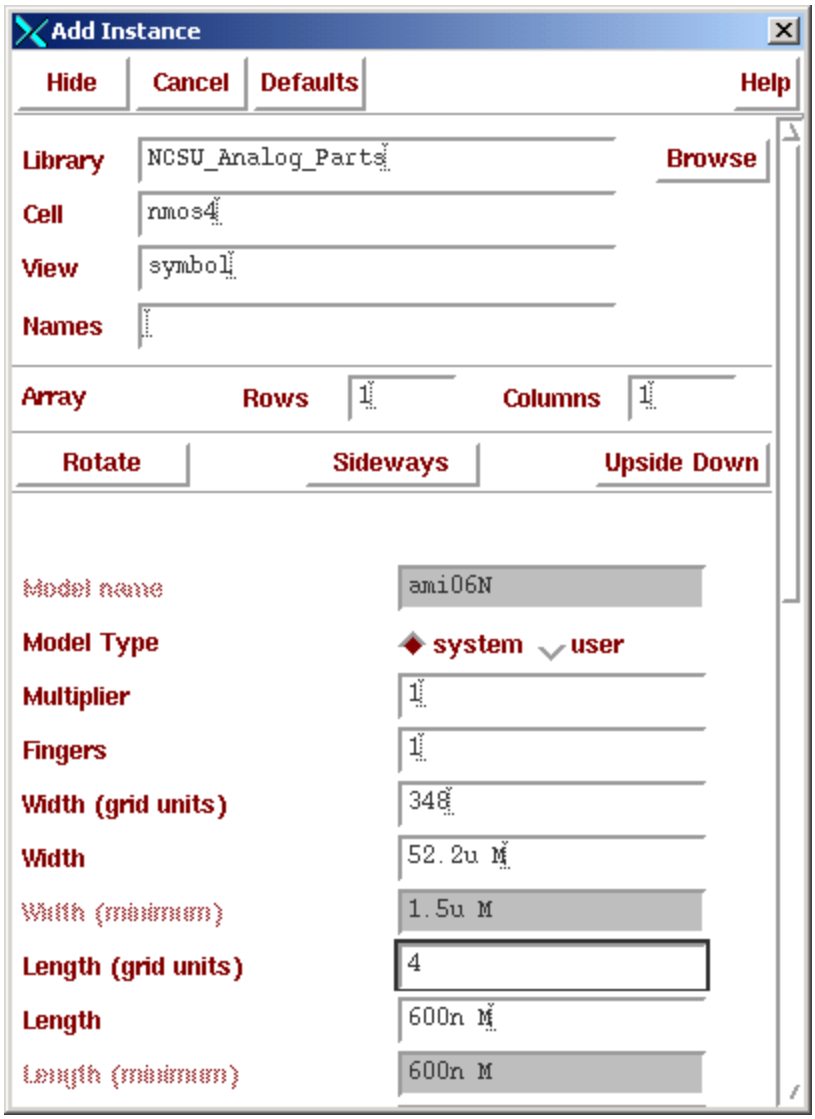

Figure 19: Getting an nmos4 device W_N=52.2µm.

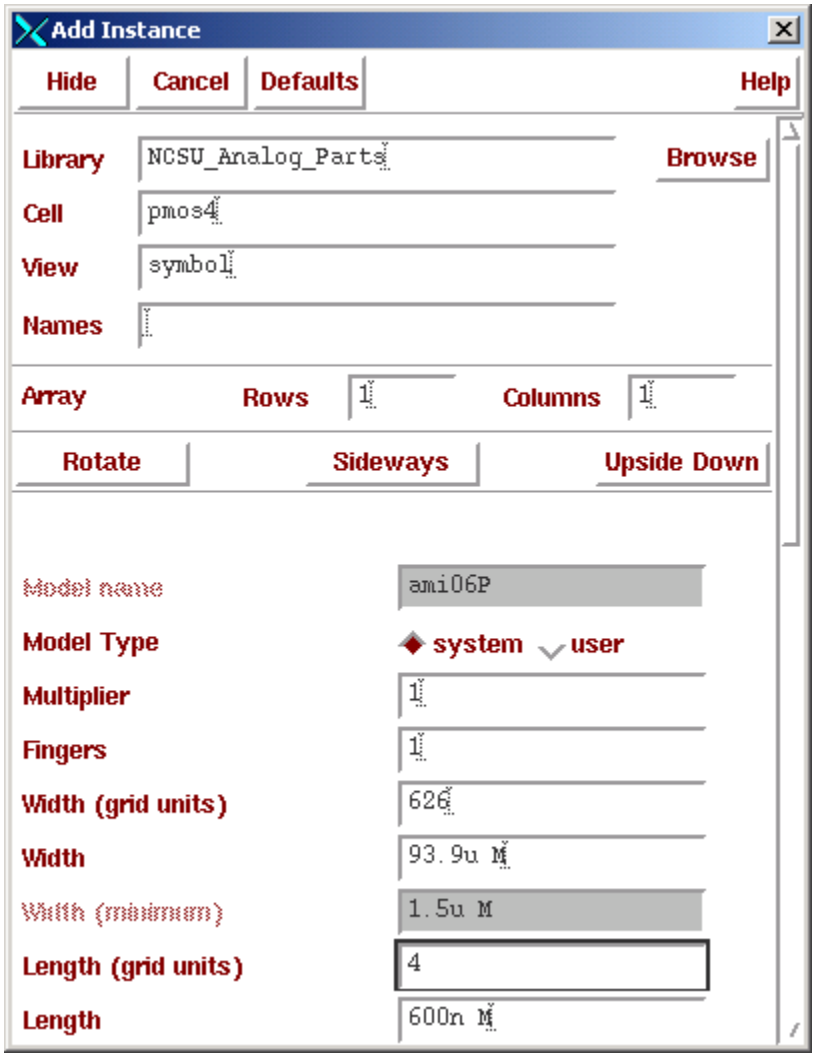

Figure 20: Getting a pmos4 device W_P=93.92µm.

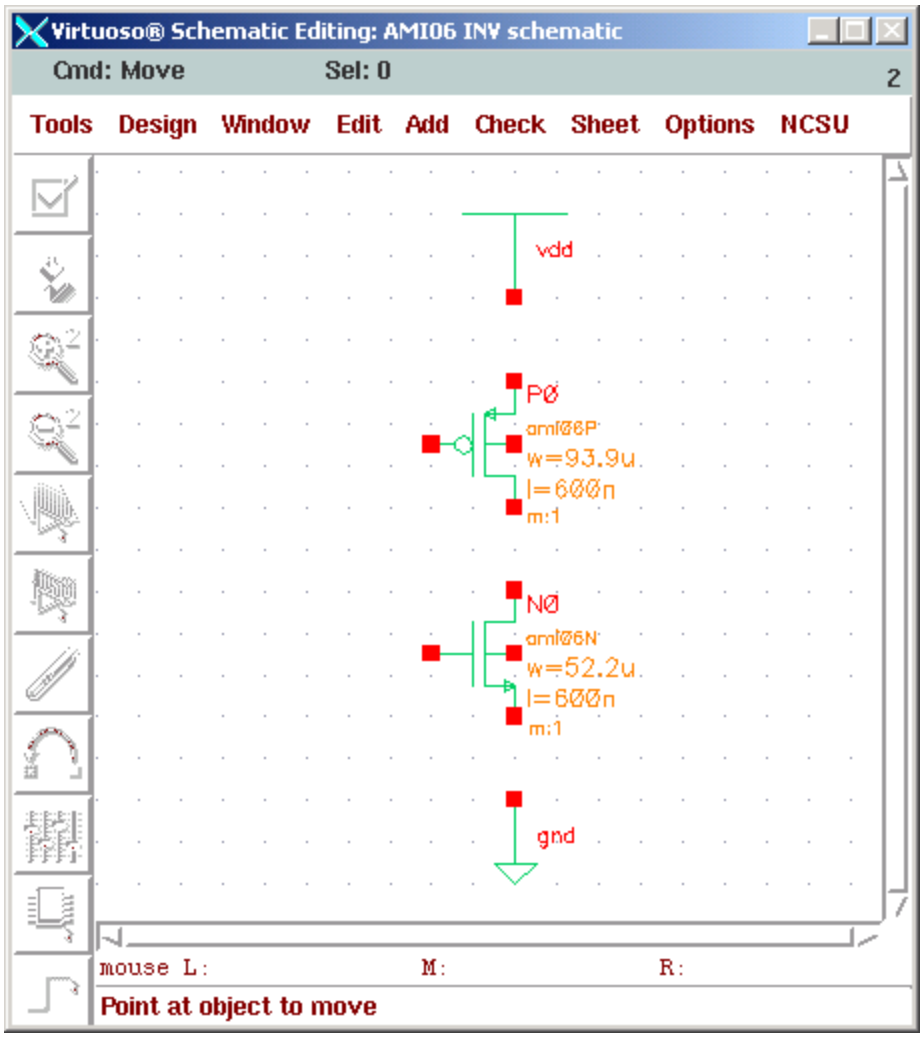

Figure 21: Major symbols of the CMOS inverter

All that remains to be completed is to wire the connections together and add inport and output pins.

To add a wire, press w. The wire will snap to place at the proper ports of each device.

Wire it up like Figure 22.

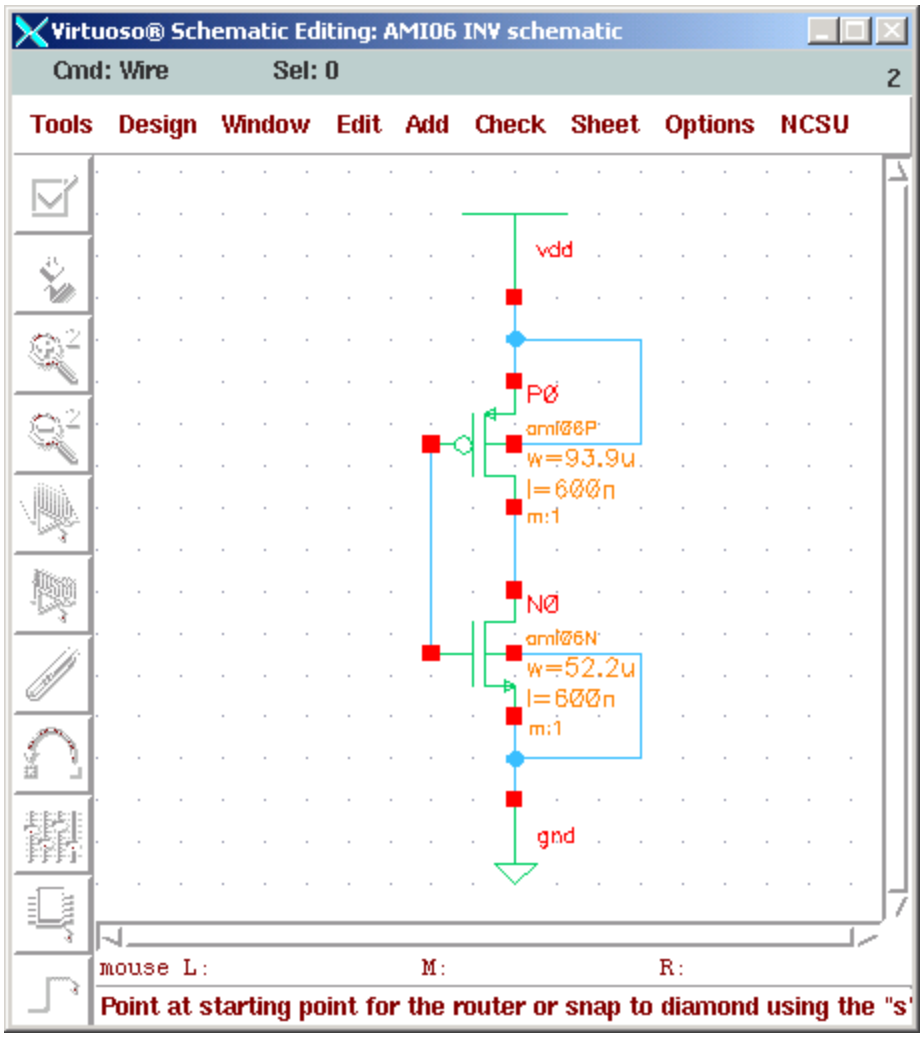

Figure 22: Wiring a CMOS inverter.

To add the input and output pins, goto add... pins and a pop up like Figure 22 should appear. Fill it out exactly like Figure 22. The pins names must match the pins names of the symbol view we are going to create later. **If they do not match (direction or name), the design, check & save routine will fail.**

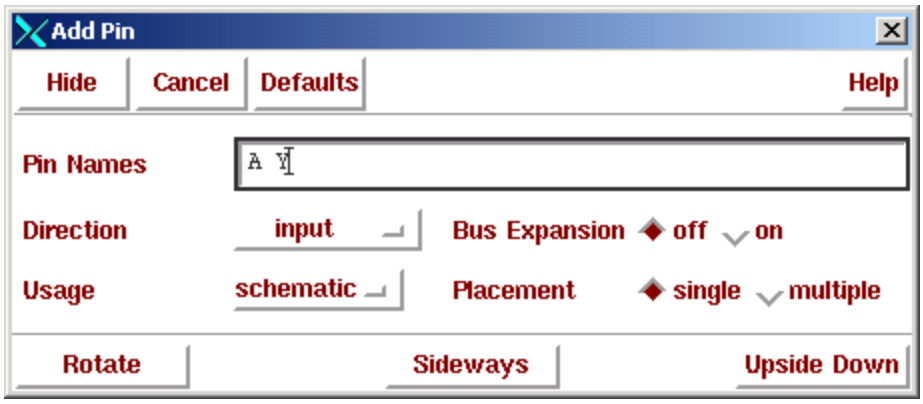

Figure 23: Add pin pop-up.

Stamp down the input pin like in Figure 25.

Change the direction of the Y pin in the add pin pop-up to output as in Figure 24.

Stamp it down like in Figure 25.

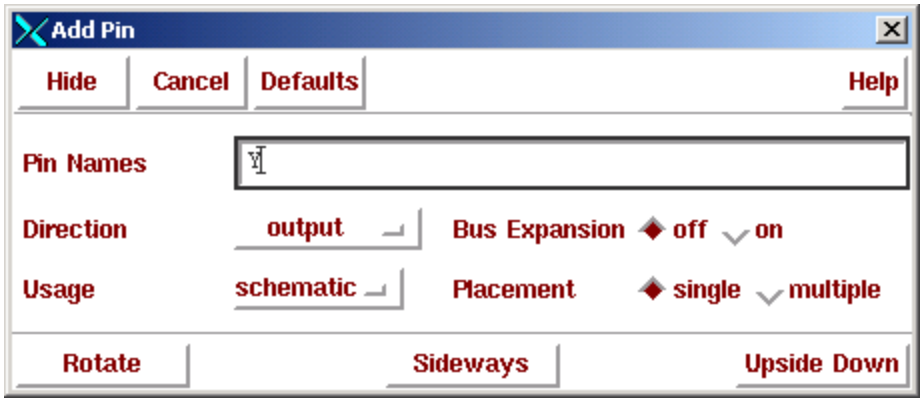

Figure 24: Changing direction on the Y pin.

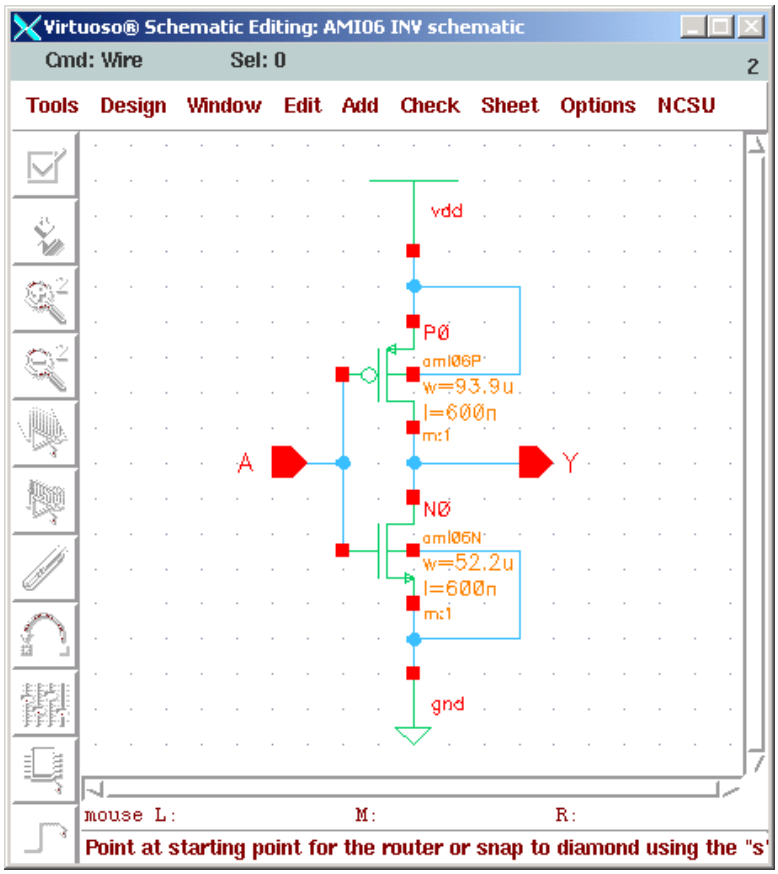

Figure 25: Completed schematic.

To connect the substrate of the pmos transistor to vdd, press the esc key to get out of the add pin mode and then press w to enter the wire mode. While in wire mode connect the nmos substrate to ground as well.

To save and check your schematic for errors, goto... design... check &save. Any errors will be highlighted in the schematic window. Usually something is not connected, or a port name is wrong. You can exit the schematic capture tool at this time (goto Window .... Close).

Creating a symbol view:

We have entered our initial design into a schematic view of the inverter. To use a test bench we must have a symbol view of the inverter. **Do not use this inverter's schematic view as a testbench other wise it will not pass LVS!**

To create a symbol view:

1. Goto the library manager popup and goto: file… New… cell view. A pop-up like Figure 26 should appear.

2. Fill out the pop up exactly according to Figure 25. Click OK. The symbol editor tool should appear (Figure 27).

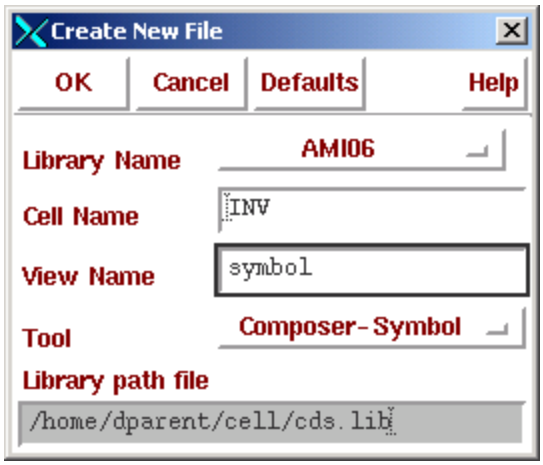

Figure 26: Creating a symbol

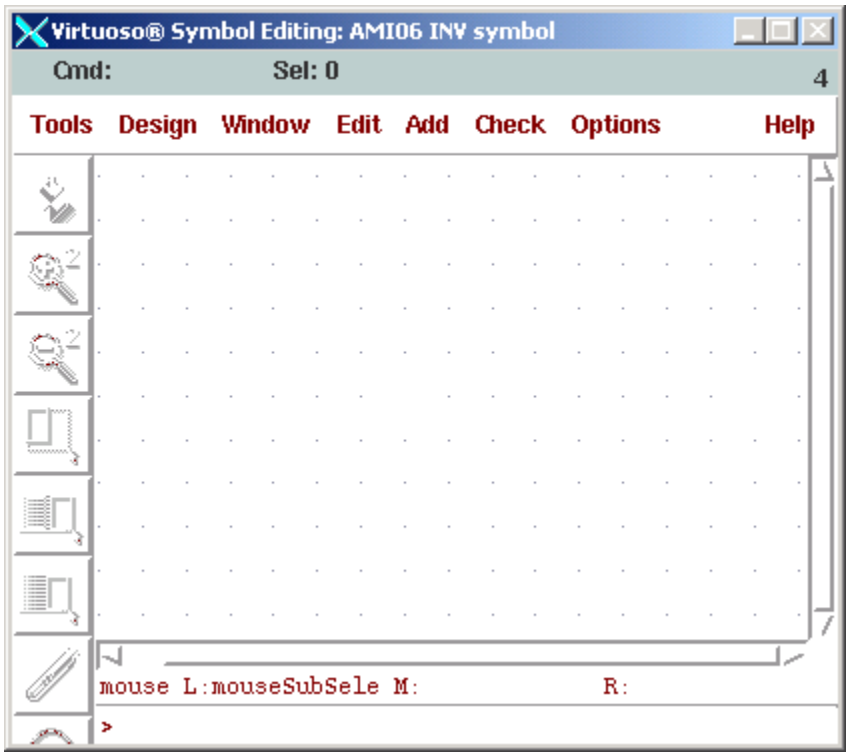

Figure 27: Symbol editor.

The symbol editor allows you to draw any kind of shape to represent you logic gate. The important part is to have the input and output pins match whatever it is you are representing. In this case, we need to match the input pin A and output pin Y of the inverter. Rather than

drawing an inverter symbol from scratch, it is much easier to copy one that has already been drawn.

To import a symbol:

1. Goto add… import symbol. A pop-up like Figure 28 should appear. Click on browse and select the inverter in the digital library of the NCSU kit. The pop-up should look exactly like Figure 28.

2. Stamp it down in the symbol editor like in Figure 29.

3. Press the esc key to get out of add symbol mode.

4. Press the f key to fit the symbol in your window.

5. Goto design… check & save. You should have no errors. If there are errors, the pins must be wrong in your schematic. Go back to your schematic and fix the pin assignments. Check for direction and name. If there is one pin misnamed there will be two errors:

1. A pin in the schematic view was not found in the symbol view.

2. A pin in the symbolic view was not found in the schematic view.

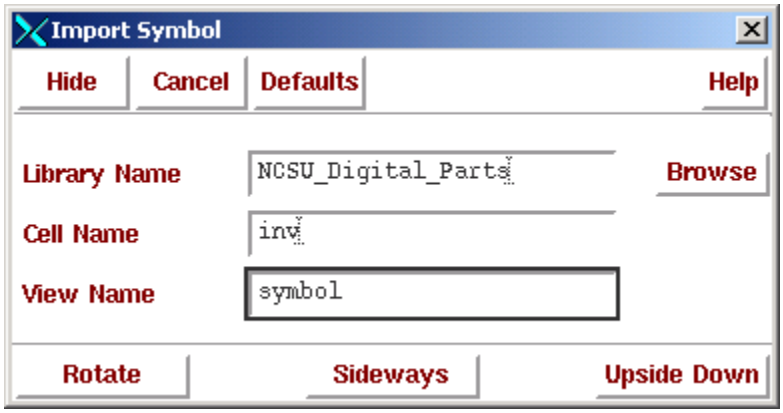

Figure 28: Importing a symbol.

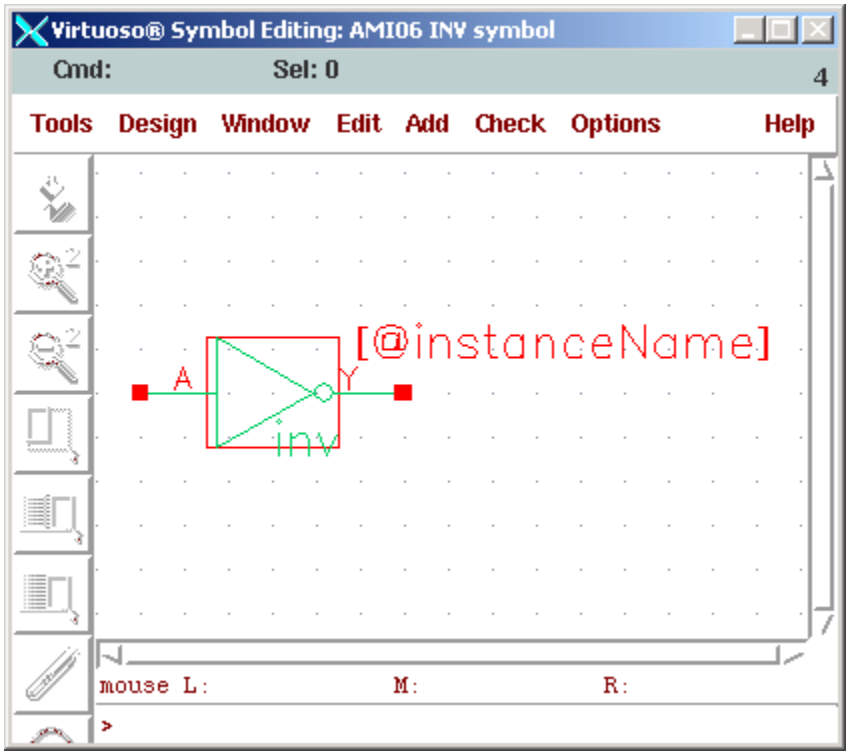

Figure 29: The inverter's symbol.

To close the symbol editor, goto window… close.

Creating a test bench:

Now that you have created a schematic and symbolic view of your inverter, it is time to create a test bench. This will be a virtual test bench, but it will have to have a power supply, input vectors, and a ground (plus the inverter) to test its transient response and DC response.

To create the inverter test bench:

In the library manager, goto file… new… cell view. A pop-up like Figure 29 should appear. Fill it out exactly like Figure 30 and click on ok. The schematic editor will appear.

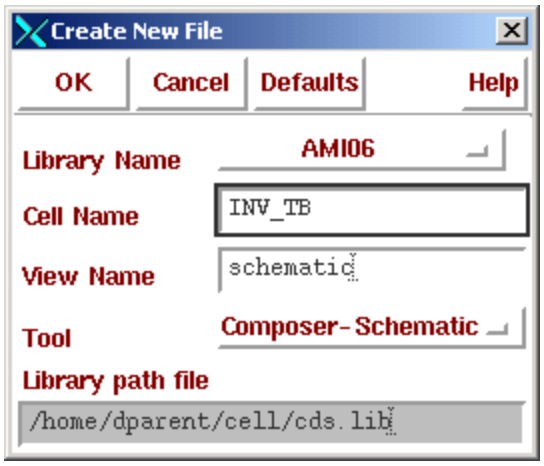

Figure 30: Creating the inverter test bench.

In the schematic editor press i or goto add... instance. Click on browse, select library AMI06, click on flatten and select your inverter as in Figure 31. Stamp it down in your schematic.

While still in add instance mode add the gnd and vdd global symbols just like you did when creating the schematic for your inverter.

Add the power supply (vdc from voltage sources in NCSU_Analogparts) according to Figure 32. Make sure set the DC voltage to 5 volts.

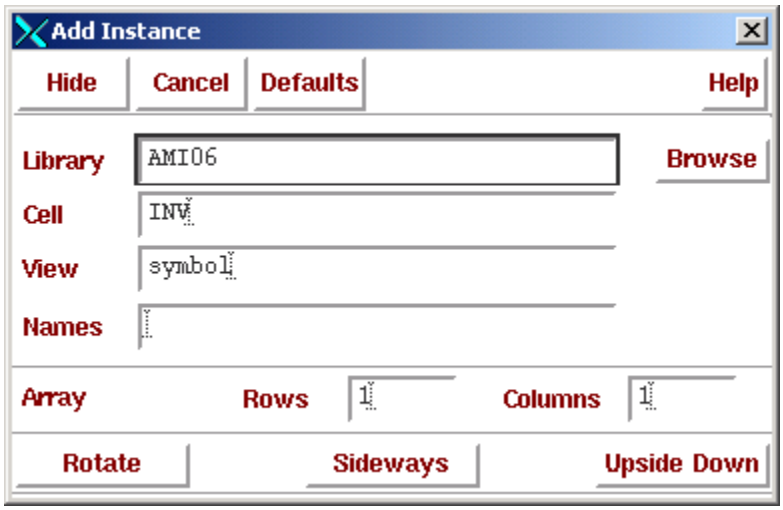

Figure 31: Adding the inverter symbol to the test bench.

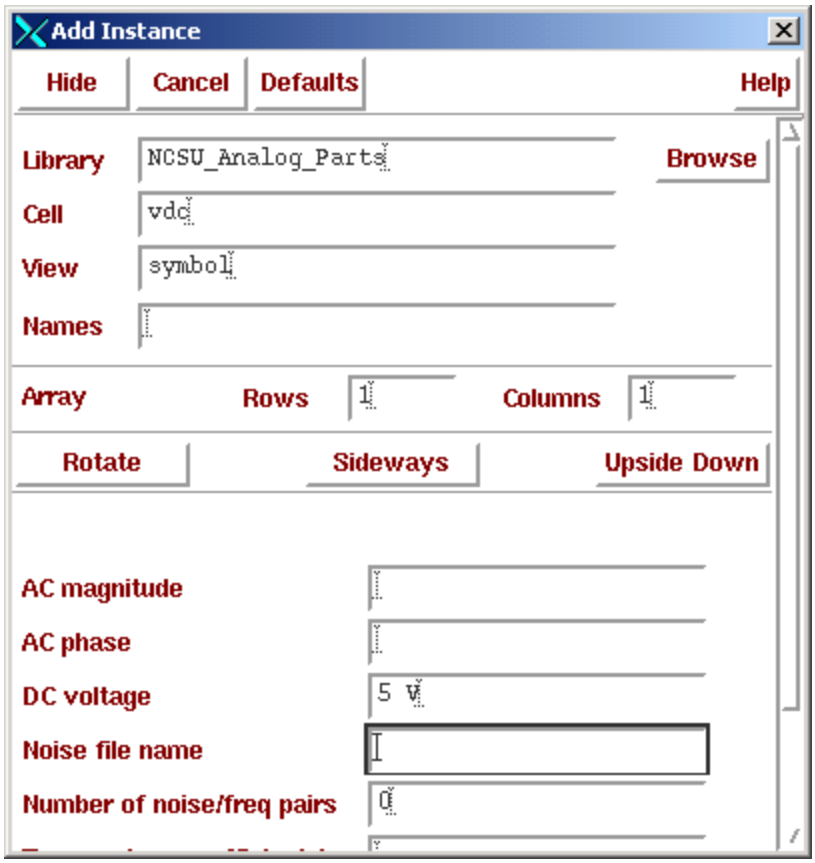

Figure 32: Adding the power supply (V_{DD} from specification).

Add the load capacitor according to Figure 33. **Make sure it is 500f F not 500 F! This simulation will not work if the capacitance is too big!**

Add the test vector (vpulse) according to Figure 34.

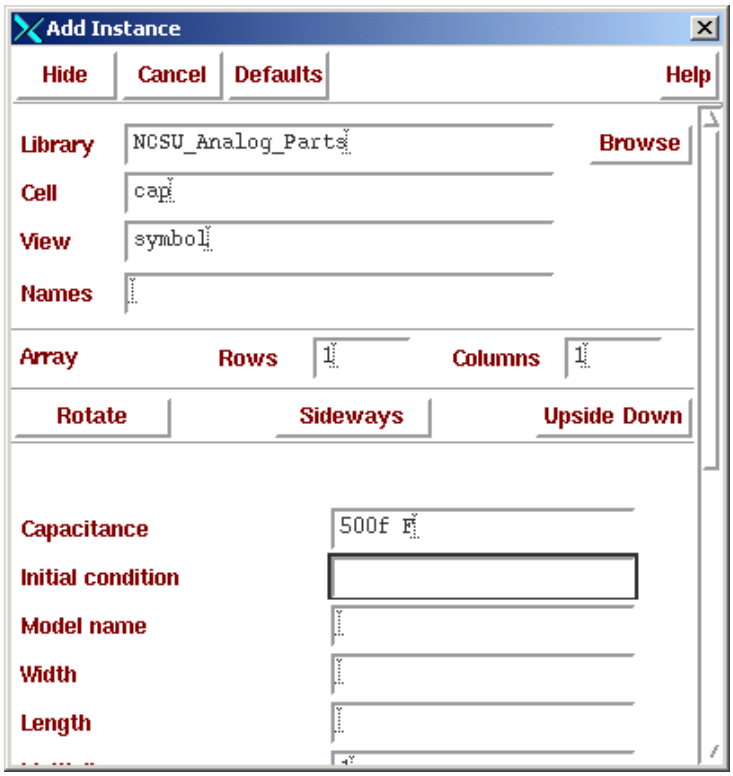

Figure 33: Adding a load capacitor (500f F from specification).

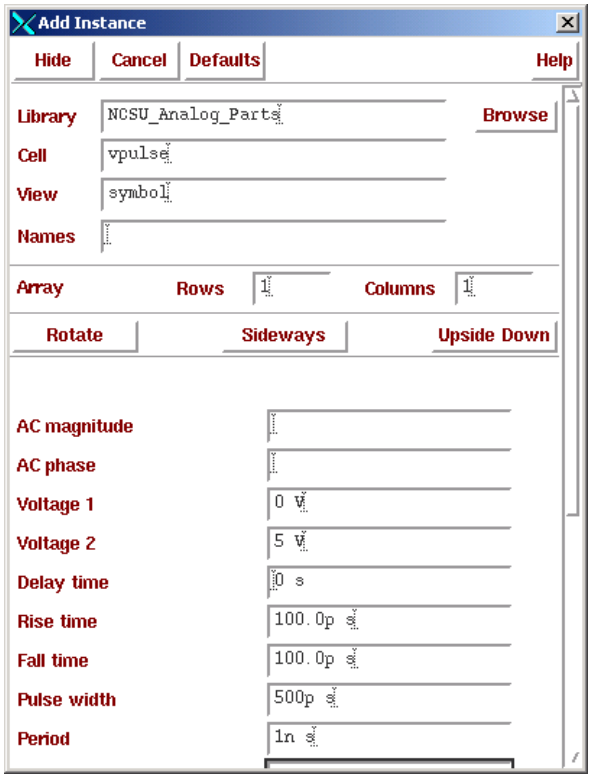

Figure 34: Adding a test vector (From specification).

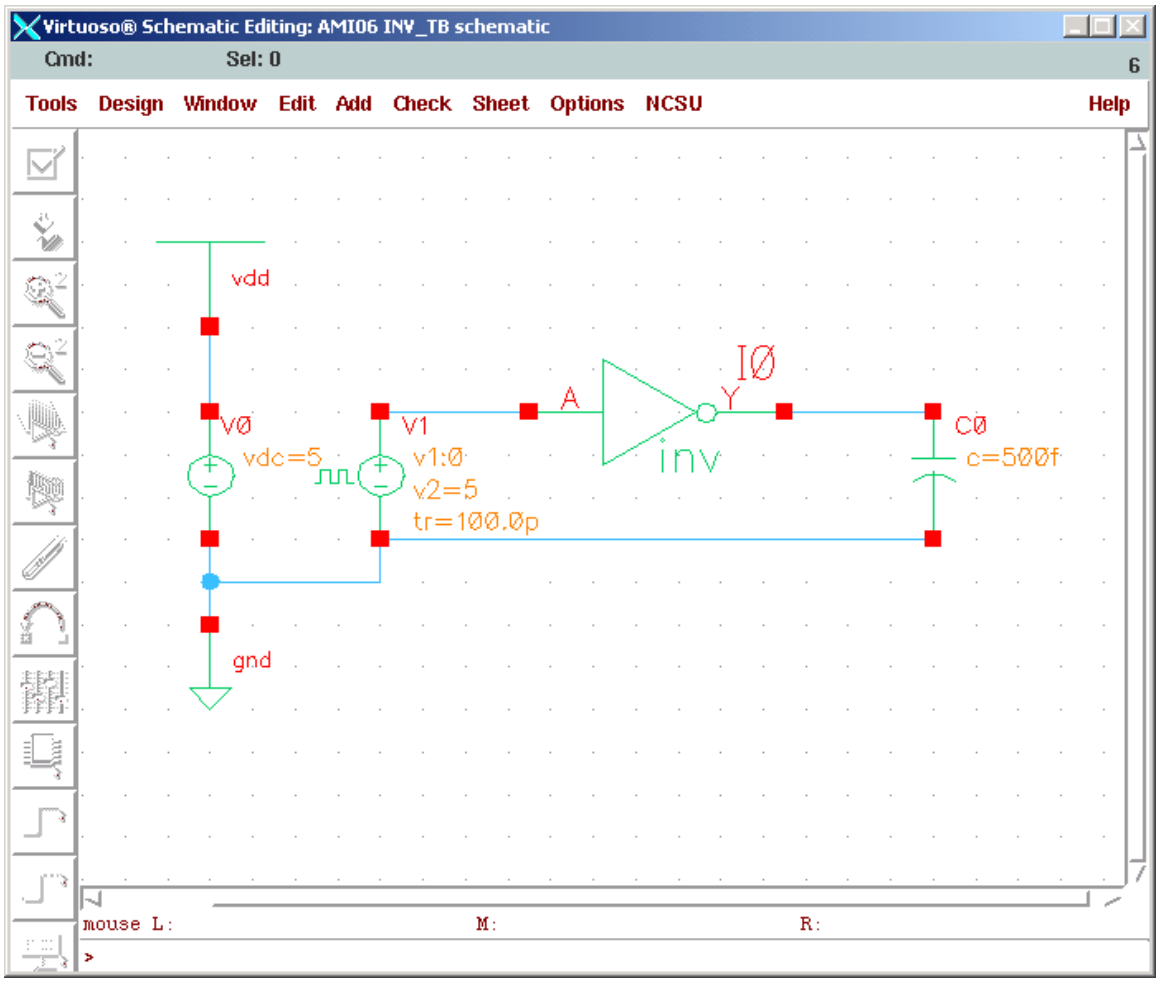

Figure 35: The completed test bench

Press the esc key to get out of add instance mode. Move the symbols around by pressing the m key for move.

Press the esc key to get out of move mode and wire up the schematic according to Figure 35.

Goto Design... Check & Save to check for errors.

You are now ready to simulate your inverter!

If you receive warnings and errors about pins being in one view and not the other, you have either name differences in the schematic vs. symbol or pin direction differences in schematic vs. symbol.

The symbol should be perfect so check the schematic you did to make sure the pins are labeled correctly and the directions are correct.

Simulation in Spectre Spice using the Affirma environment:

Now that you have created a schematic and symbolic view of your inverter and created a test bench to test your inverter, it is ready to run the Affirma Analog environment tool. Do not be confused because we are in a digital design class! We are simulating a digital circuit in a physics based environment that solves for voltages and currents over time. This is different that an HDL, which is just doing a high, low, Z or X analysis.

To verify your circuit:

Goto tools... analog environment. A pop-up like Figure 35 should appear. You will need to customize your environment the first time you run Affirma.

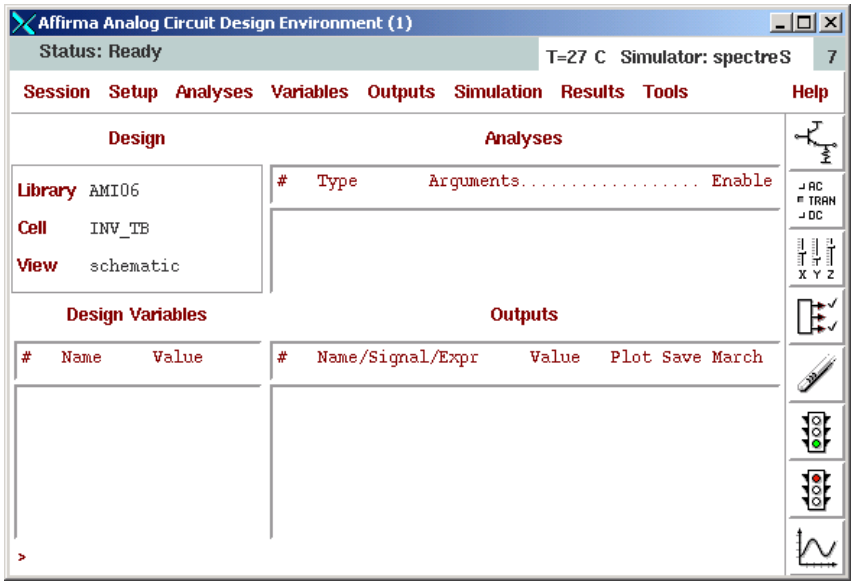

Figure 36: Affrima Analog Environemt.

You will be using the Spectre Spice simulator, which is a spice like simulator, using BSIM3 model decks, but the underlying algorithms are different. These algorithms are transparent to the user.

Goto Setup... Simulator/Directory/Host. A pop-up like Figure 36 should appear. Make sure the Simulator is set to spectreS. Click ok when done.

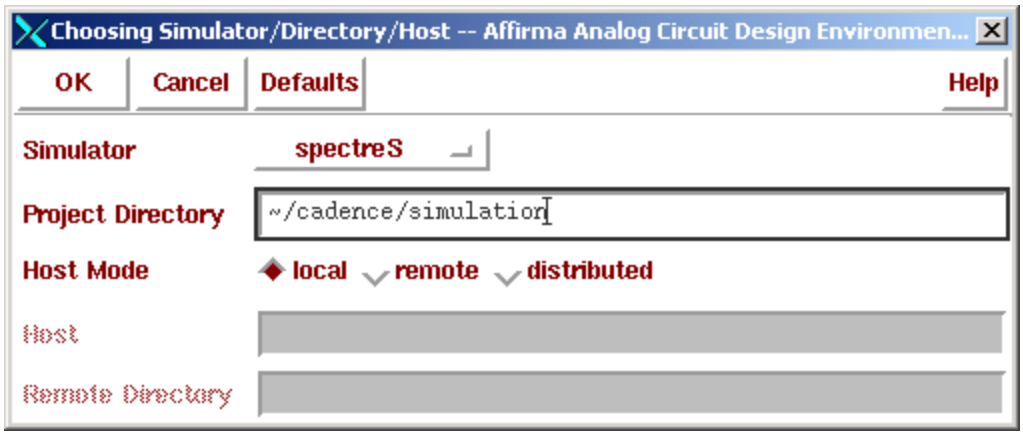

Figure 37: Setting up the simulator to use.

Now you need to make sure that Affirma can find the model decks for our process.

Goto Setup... Model path and a pop-up like Figure 37 should appear. Make sure it matches exactly the model path in the pop-up.

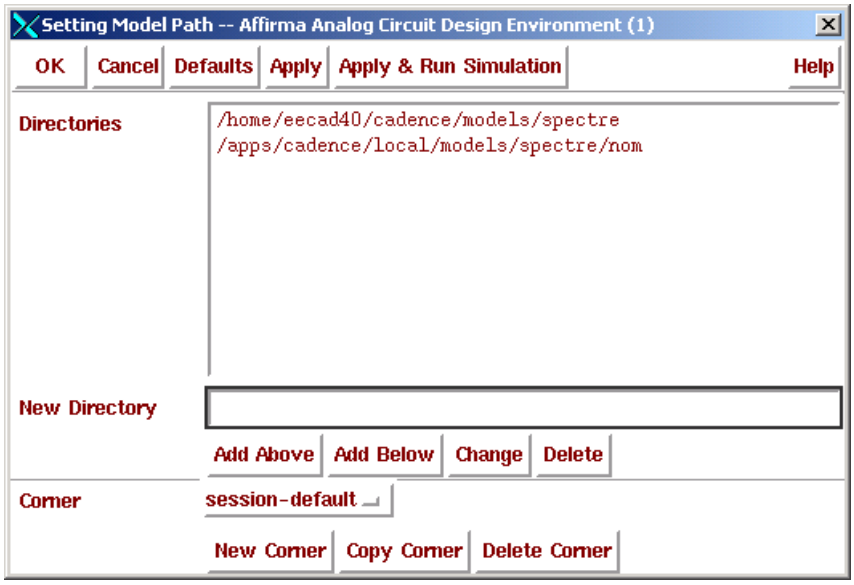

Figure 38: Setting up the model path.

Transient Analysis:

The transient analysis will allow us to measure propagation delay times. Our specification calls for 100ps for both propagation delays. To set up a the transient analysis: Goto Analyses... Choose and fill out the pop-up according to Figure 38. In order to measure V_{INV} (Where $V_{OUT}=V_{IN}$) we need to do a DC analysis as well. V_{INV} is a measure of the noise performance and the closer V_{INV} is to $V_{DD}/2$ the better the noise performance of the inverter.

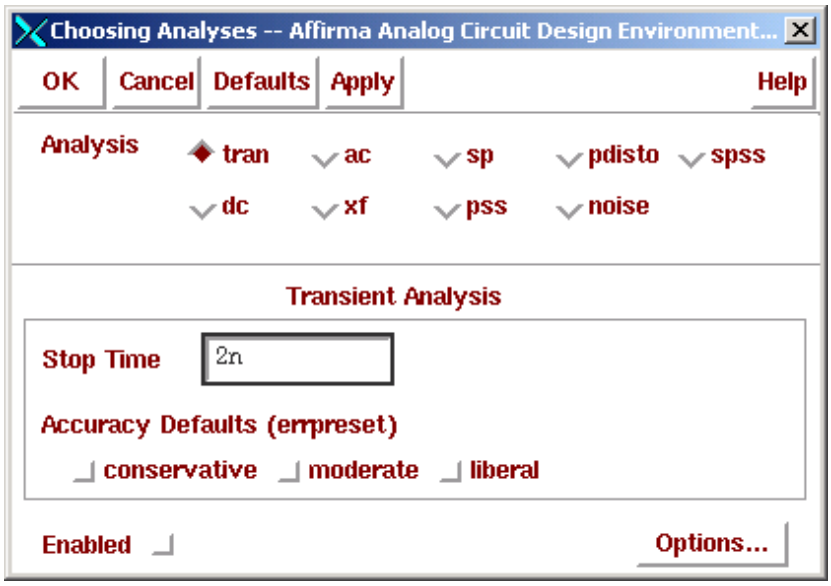

Figure 39: Setting up the analysis

DC Analysis:

In order to measure V_{INV} (Where $V_{OUT}=V_{IN}$) we need to do a DC analysis as well. V_{INV} is a measure of the noise performance and the closer V_{INV} is to $V_{DD}/2$ the better the noise performance of the inverter.

Goto Analyses... Choose and fill out the pop-up according to Figure 40.

Click on Select Component and then in the schematic click on the vpulse.

Select VDC when the pop-up like Figure 41 appears.

Select the start and stop voltages to be 0 and 5. Your pop-up should look like Figure 42 Note: **When simulating a Schmitt trigger you will have to do this analysis twice 0 to V_{DD} and V_{DD} to 0 to get the hysterisis curve**.

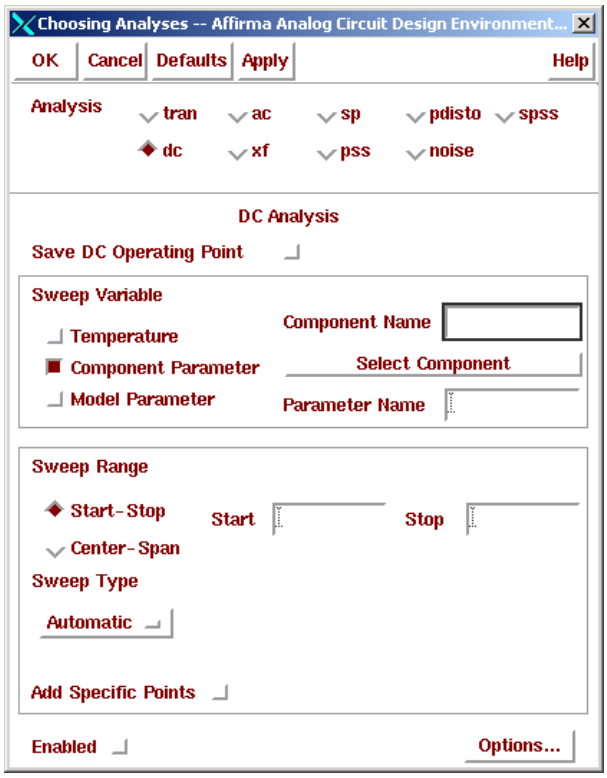

Figure 40: Setting up a DC analysis.

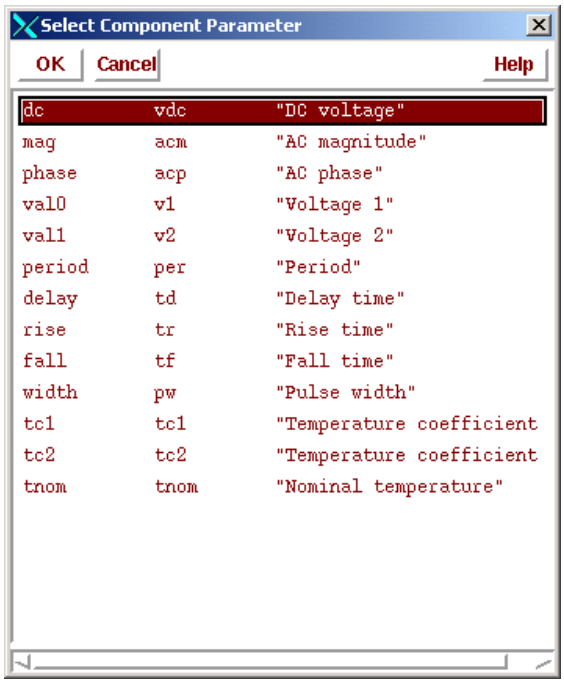

Figure 41: Choosing to sweep DC voltage.

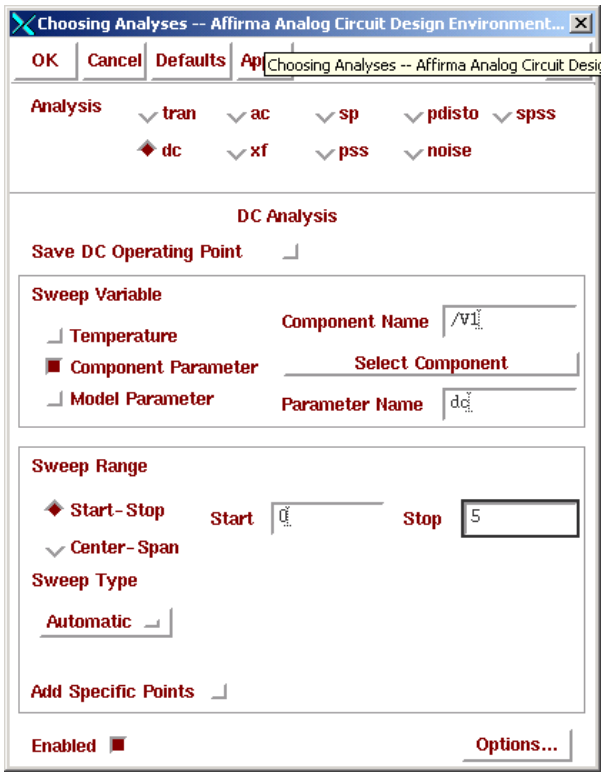

Figure 42: Completed DC Analysis Setup.

Adding Wire Names:

To easily see which signal is the input and which is the output it is advisable to add wire names to the wires connecting A and Y of the inverter. To add:

Goto Add ... Wire names and fill out the pop-up according to Figure 43.

Stamp down each name as in Figure 44.

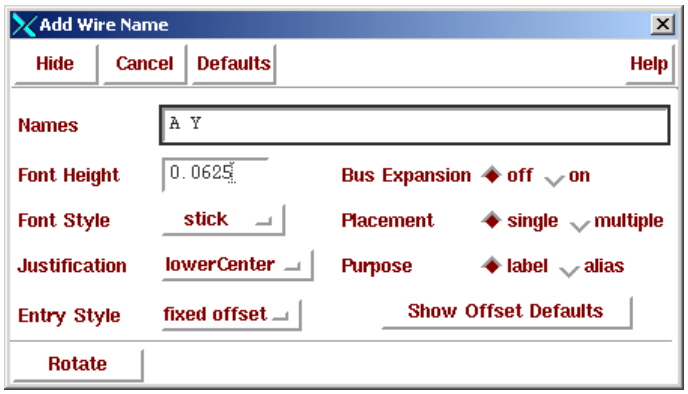

Figure 43: Adding Wire Names.

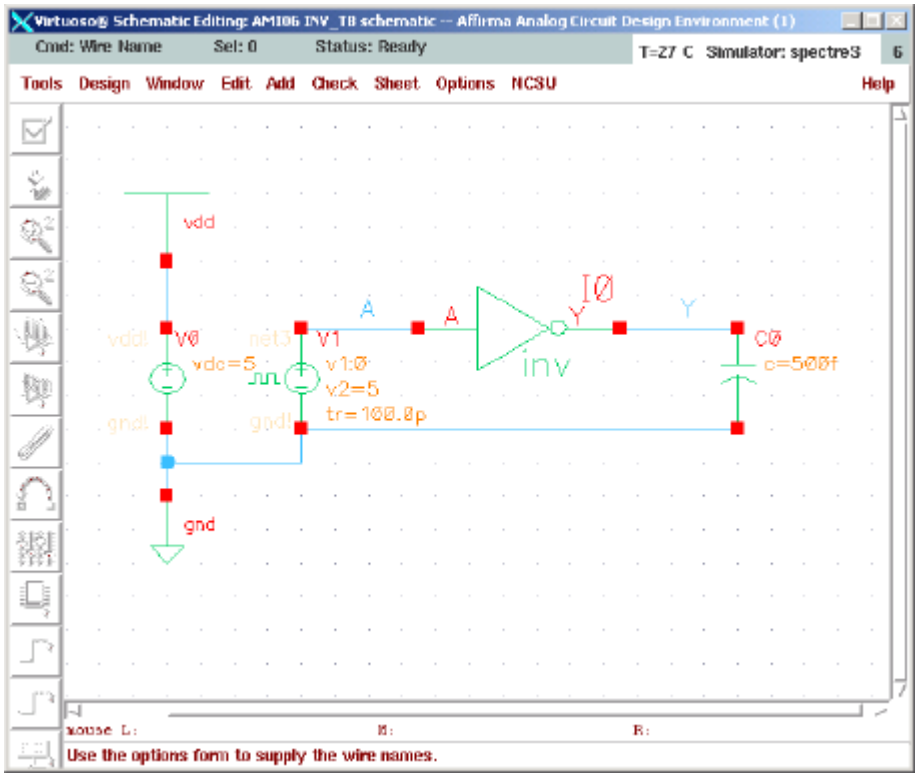

Figure 44: Schematic with Net Names.

Choosing which nets to plot.

Now you need to choose which vectors you want to plot. Goto Outputs... To be plotted... Select... on schematic. Go back to your inverter test bench schematic and click on the wire going into your inverter and the wire between the output of your inverter and the load capacitor. The lines should change color as you select them. If you click on a pin it will probe the current and the terminal of the device you clicked will have circles drown around them. When you are done it should look like Figure 45.

Press the esc key to get out of selection mode. Your Affirma pop-up should look like Figure 46.

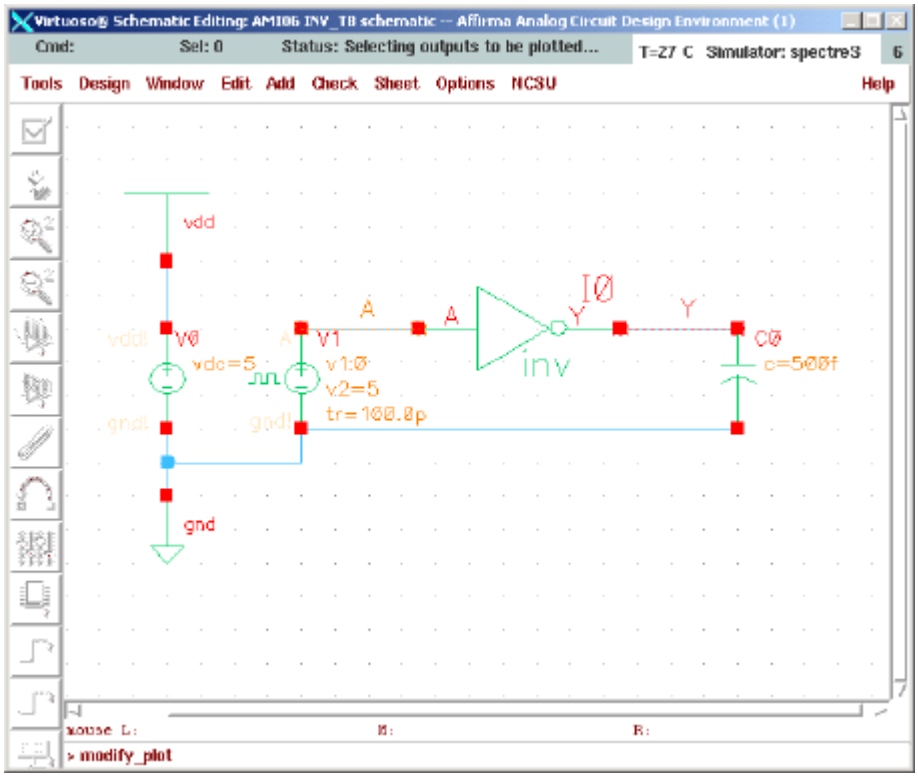

Figure 45: Schematic with Nets selected for Plotting.

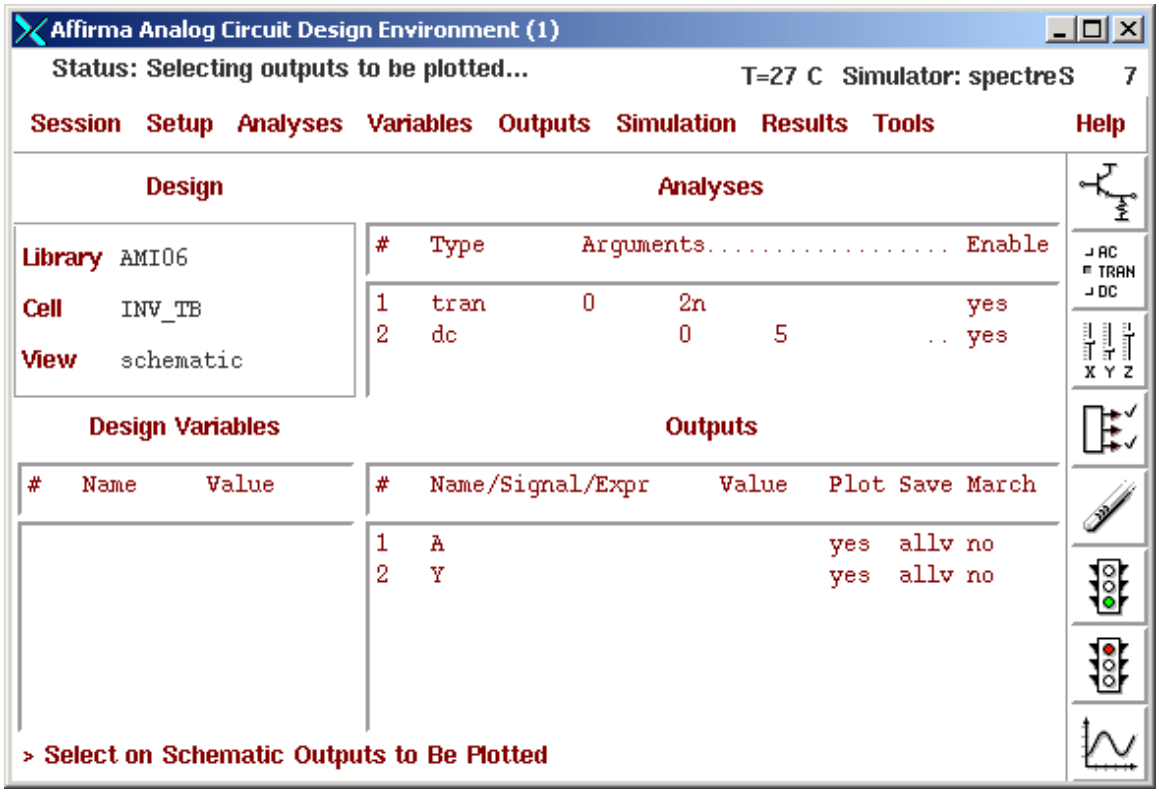

Figure 46: Selecting which vectors to plot.

In Affirma, Goto Simulation... Run. The CIW and the Affirma pop-up will show you the process it goes through to simulate your circuit. If all went well, a plot like Figure 47 should appear.

If the simulations worked, go back to the CIW and goto Options... Save defaults and click OK on the pop-up. This will save your simulator environment, but not your plot settings.

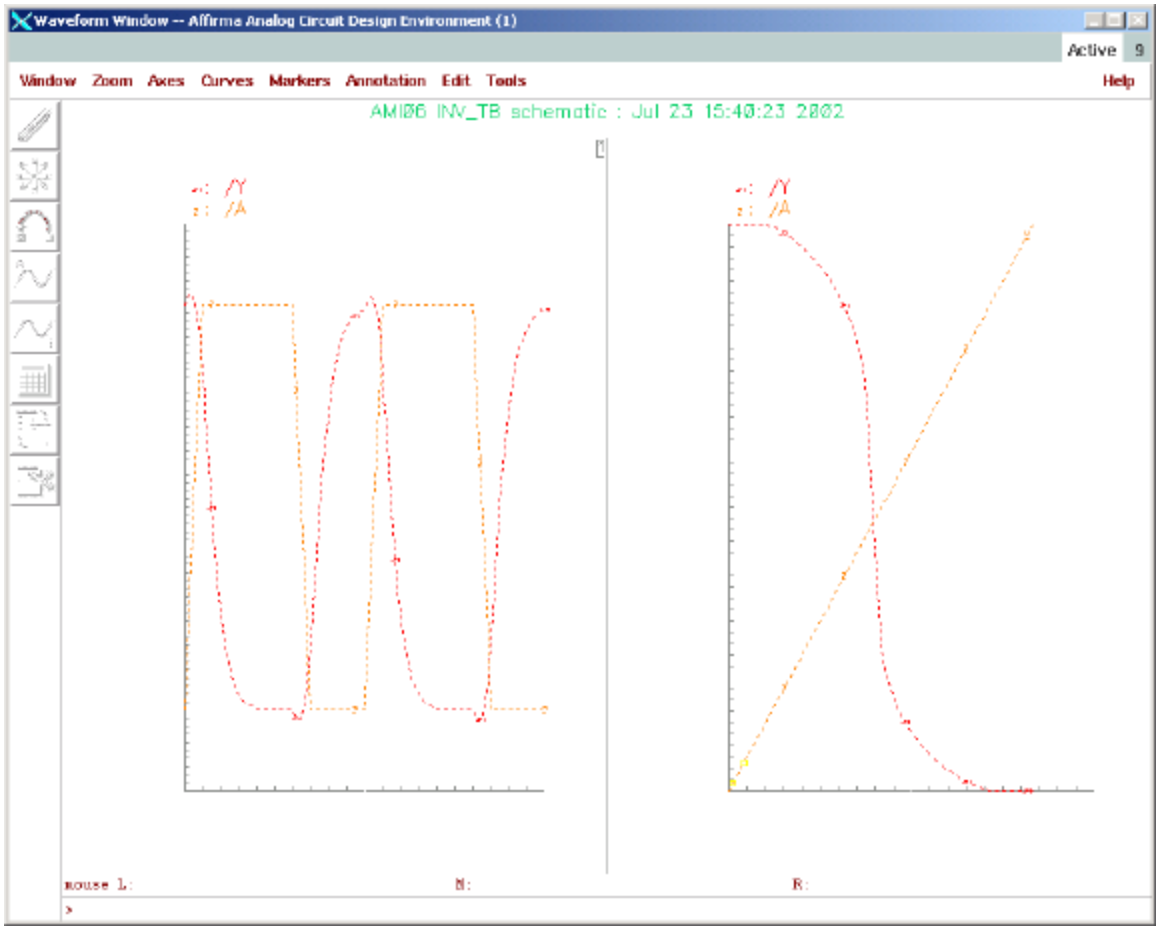

Figure 47: Plots of Transient and DC Analyses.

Simulation Problems:

Oh no, the simulation did not work!

If you get error messages like “model XXX not found” you did not set the path correctly.

If you get a spice error message, you probably did not set your vpulse, or vdd correctly.

If it takes a long time to run, and or simulator times out, the capacitor could be too big, or the width or length of a MOSFET could have the wrong units (for example, instead of 4 microns it is 4 meters).

If the simulation run but the output does not look like Figure 47, the capacitor could be too big or the DC voltage is not set correctly.

To fix or change properties of the inverter goto Design... Hierarchy... descend edit and click on your inverter. Change the pop-up to schematic and click ok. To edit a mosfet or anything else press q and click on what you want to edit. To return to the top, goto design... hierarchy... return to the top.

Measuring Propagation Delays and V_{INVTH}:

It is very hard to read off propagation delays from the plot. To easily read propagation delays:

Make sure the Transient Plot is selected by clicking on the number 1 in the waveform window. In the Waveform Window Goto Markers... Horizontal Markers and fill out the pop-up according to Figure 48. Click on Apply and then Display Intercept Data

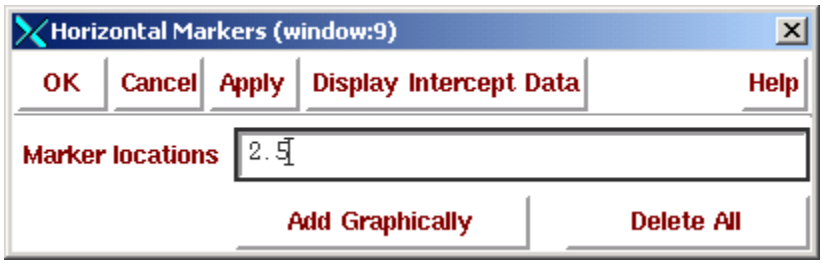

Figure 48: Adding a Horizontal Marker.

You should see results like in Figure 49.

To find τ_{PLH} subtract Curve 1's time form Curve 2's at the first intersection of $V_{DD}/2$ (149.23p – 50p). We get τ_{PHL} to be very close to 100ps.

To find τ_{PLH} subtract Curve 1's time form Curve 2's at the second intersection of $V_{DD}/2$ (748.99p –650p). We get τ_{PLHL} to be very close to 100ps.

We met timing specification!

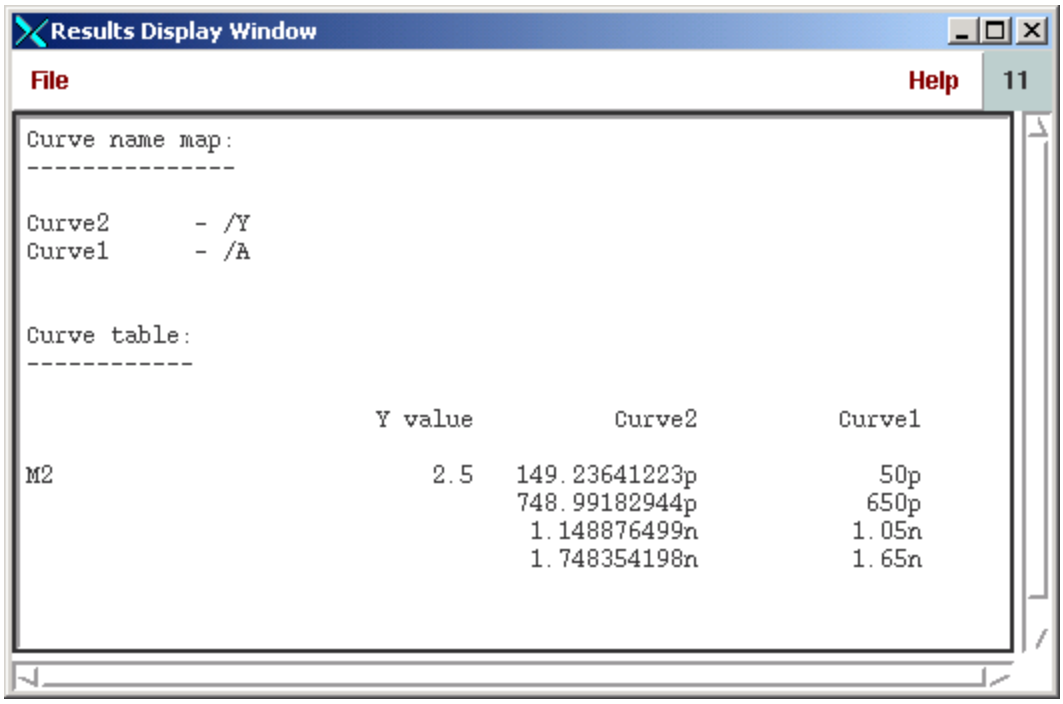

Figure 49: Results from a Horizontal Marker.

To check for V_{INVTH} click on plot 2 DC analysis and press A for a cross hair marker. Place it on the intersection of the two lines and read the value off the bottom as in Figure 50. We get V_{INVTH} to be 2.4 Volts, which is good enough to met specification.

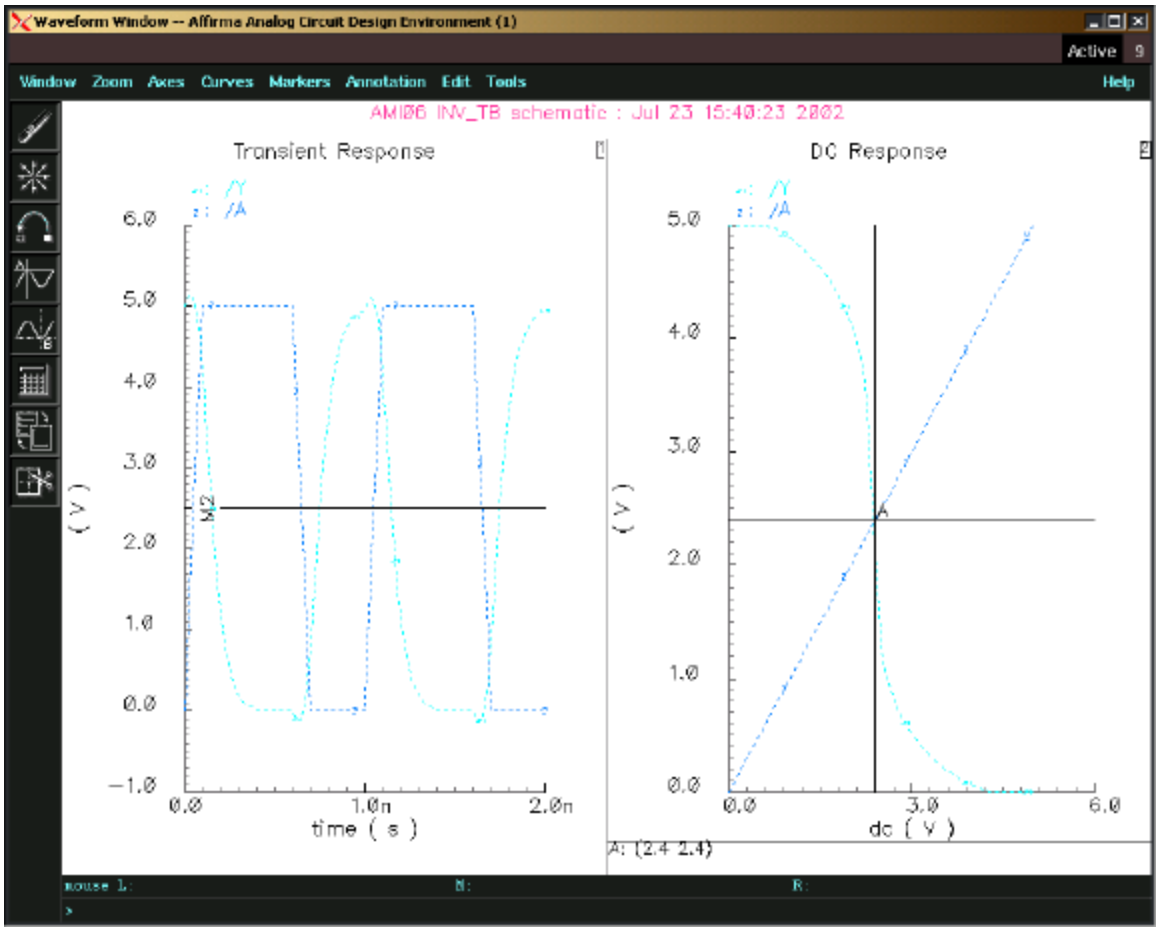

Figure 50: Final Results.

Measuring Power:

We have met our timing specification, but are we within our power budget? We can add a power meter to the schematic (Kang and Leblebici page 245). Too add a power meter to the schematic:

1. Press i to bring up the add component pop-up and click on browse. Set the library to NCSU_Analog_Parts and then click on Uncategorized. You should see a pop-up like Figure 51. Figure 52.

2. Stamp it down in your schematic and wire it up like Figure 52.

3. Add a capacitor and ground to the circuit, and wire it up like in Figure 52.

4. Edit the capacitor's value to be T/V_{DD} (1e-9/5, do not use units!). This is important so that the power is scaled properly. If you change V_{DD} or the period of the clock, you have to edit the capacitor value. (See Figure 53.)

5. Add the wire Name P to the net connecting the capacitor and the power meter as in Figure 52.

6. Add the net P to the plot list and run the simulation.

7. In the waveform window, goto Axes... Strip and you should see results like in Figure 54.

8. Add a cross hair marker A and B like in Figure 54. The delta will give the power for 1 clock cycle. Each clock cycle the power that is used will be added to the power already used. (Note even the axis is label in volts, read off the values in Watts.)

We used 19.78mW of power, which means we met specification! (Just barely, due to statistical variation it might be possible for some manufactured circuits to use more power.)

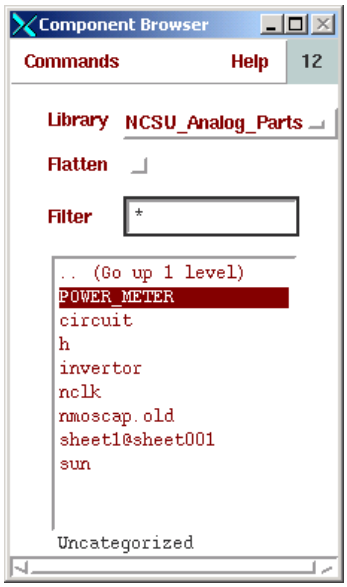

Figure 51: Adding a Power Meter.

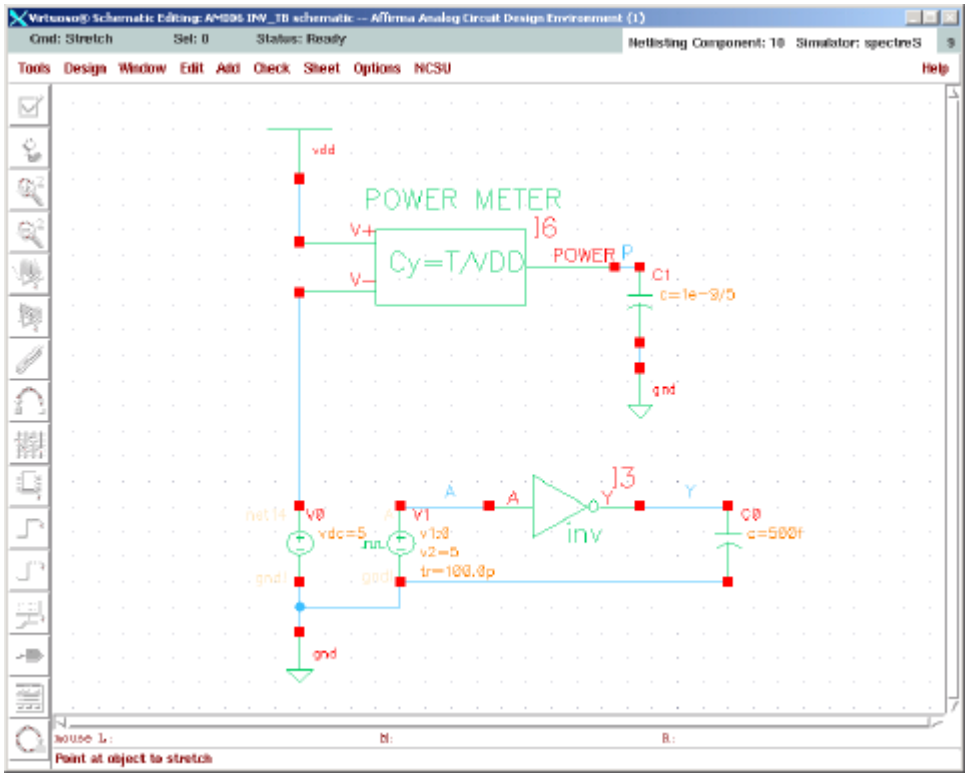

Figure 52: Schematic with Power Meter Added.

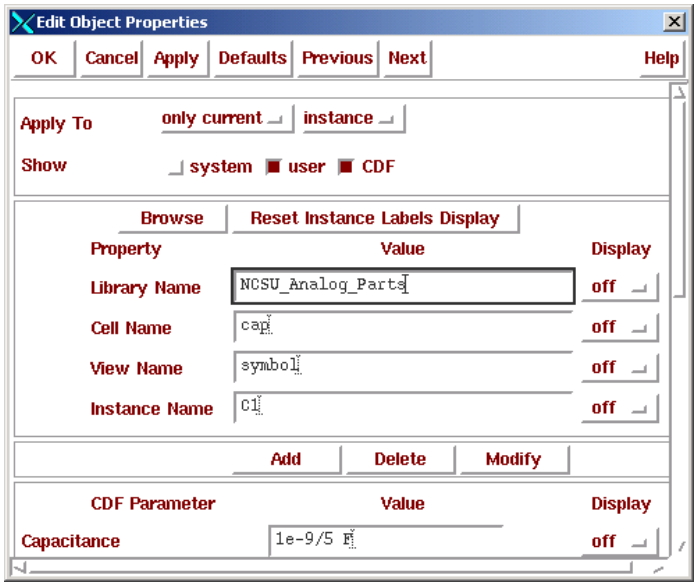

Figure 53: Editing the Capacitor's Value.

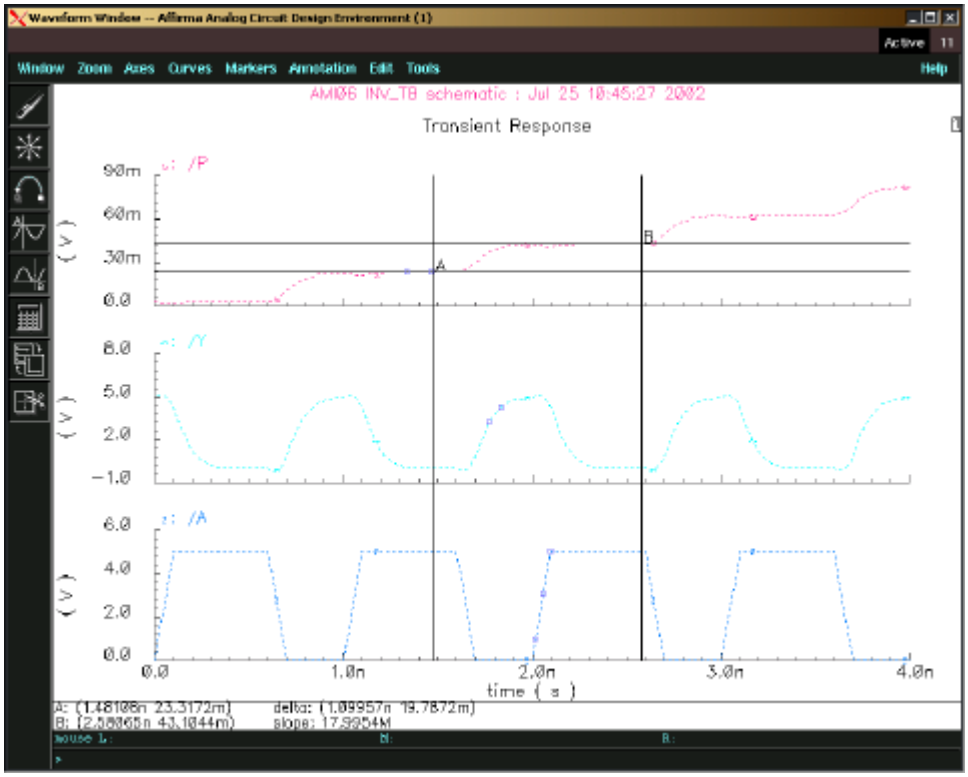

Figure 54: Waveform with Power (19.78mW).

Saving your State:

To make sure you do not have to enter all the information into Affirma every time you want to run a simulation, save the state. To save the state goto Session... Save the state in the Affirma Window and fill it out according to Figure 55.

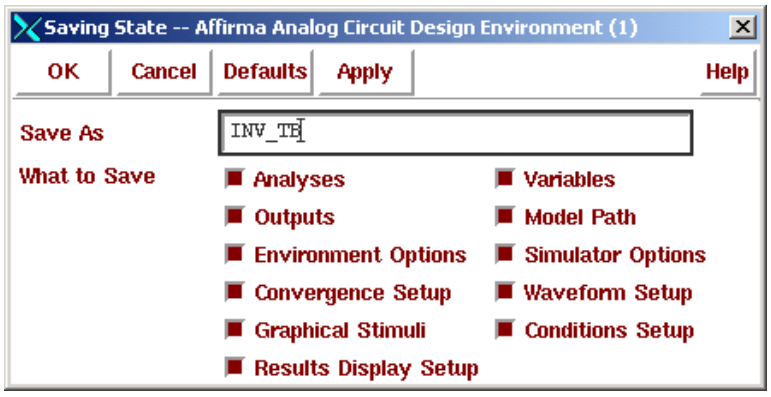

Figure 55: Saving the State.

Summary:

We have simulated our model with a more accurate tool and have met the logic, timing, noise and power specifications. We have set up a test bench to verify our inverter that will be used again. We are not at the point where we can have our inverter fabricated. We need to create a layout view of our inverter, which is essential, a picture that shows the shape of a CMOS inverter, as it would look like under the microscope. The layout tools is used to create a representation of the circuit that can be used by the fabrication house's mask maker to make the phototypographic plates that will be used in production. In chapter, three we will learn how to use the layout tool by following the Cell Design tutorial developed by Cadence, and then we will create the layout view of our inverter.

Chapter

4

Chapter 4: Layout

We will take a break from our inverter and complete the Cell Design Tutorial developed by Cadence so learn how to use the layout tool, perform Design Rule Checking (DRC) and Layout vs. Schematic (LVS). After completing chapter's 1-4 of the Cell design Tutorial we will then layout our inverter.

Cell Design Tutorial

CDS has developed a cell design tutorial, which takes the user through the layout, DRC, and LVS parts of the design flow. It does not use a library that can be fabricated through MOSIS, but it is very well done. **If you complete the cell design tutorial of CDS it will be much easier to complete this tutorial., but you do not have to. Also, the LVS check will not work for this tutorial. It is probably fine just to read the tutorial.** The CDS tutorial is started in the following manner:

1. Make sure you are using a c-shell (Type csh at terminal)
2. Type in the command **cdsdoc &** at the % prompt.
3. A pop-up will appear like Figure 56.
4. Click in the upper right of the pop-up and set to docs by family.
5. Go down in the window, select Virtuoso, and select cell design tutorial (Figure 57.)
6. You might have to click several times for it to work. A netscape browser will appear like in Figure 58. All you have to do is follow the instructions.

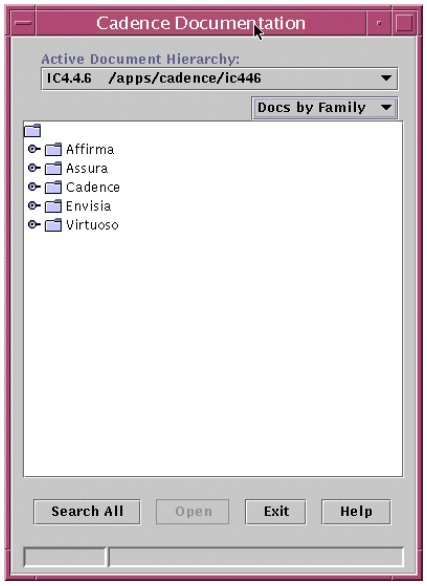

Figure 56: Documentation Browser

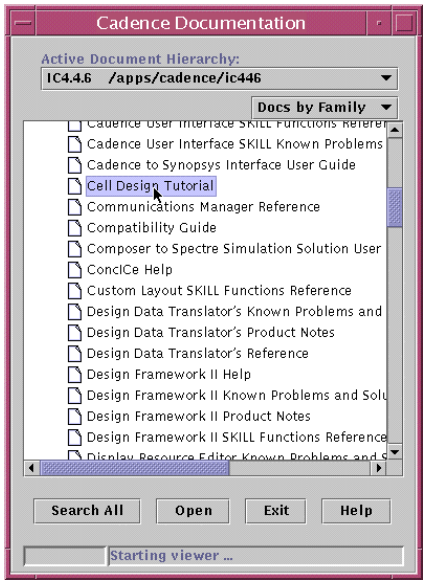

Figure 57: Cell design tutorial

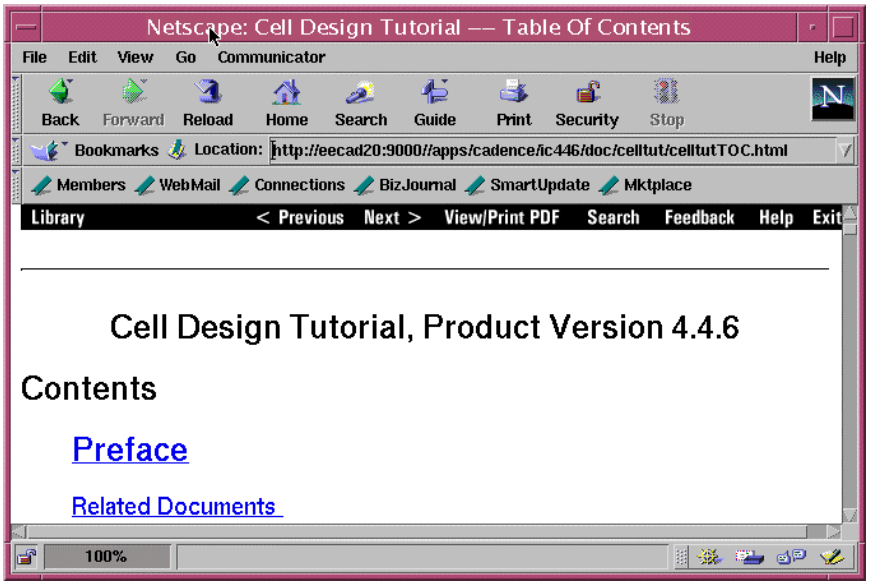

Figure 58: Netscape Browser with Tutorial

Complete chapters 1-4 of the cell design tutorial.

Preliminary Steps to Layout of the Inverter:

Now that you know how to use the layout tool, run a DRC Check, extract the spice model from your layout and run an LVS check, it is time for you to do it on the inverter design. We have good values for W_N , L_N, and how to connect the transistors, we just do not have a layout view so it can be ultimately fabricated.

We have sized the lengths and the widths of the PMOS and NMOS transistors and we relatively certain that we will meet specification. Now we will create the layout view of the inverter

We need to do some planning before we start the layout. We need to come up with a rough floor plan, make sure our metal1 power line widths are not prone to electromigration failure and make sure that our transistors fit within the cell height.

Floor Planning:

First, we need to make some assumptions about where everything will go. Our specification requires that the cell height needs to be 40µm, but our transistor widths (W_N and W_P) total to over 140µm! The solution to this is to have smaller transistors in parallel, but how many parallel transistors should we use? We cannot use the whole 40µm because we need room for routing the power lines, input lines and minimum separation requirements. We need to make sure the power lines are wide enough to ensure that there will be no electromigration failures. The general floor plan of our inverter will be standard. The top of the circuit will be the V_{DD} power line. The PMOS transistors will be below the V_{DD} lines. The poly input lines will be in between the PMOS and NMOS transistors, as well as the metal1 output lines. The ground line will be beneath the NMOS transistors. The ntaps(body connection for the PMOS transistors) and the ptaps (body

connection for the NMOS transistors) will be to the left of the transistors. The input will be to the left, while the output will be to the right. For ease of routing, only metal1 and poly will be used in this inverter. To start putting things down we need to know how wide the metal1 V_{DD} and ground lines should be. Then we can design the number of fingers (parallel transistors) we should use.

Sizing the VDD and Ground Lines for Electromigration Failure Prevention:

The way to keep lines from having electromigration failures is too keep the average current density below a critical value. The design equation for the minimum width of the metal1 due to electromigration effects is:

$$W_{M1} := \frac{I_{ave}}{J_C} \quad (8)$$

Where:

$$I_{ave} := \frac{1}{12} \cdot \frac{K_N \cdot \tau \cdot f_{CLK}}{V_{DD}} \cdot \left(V_{DD} - V_{TN}\right)^3 \quad (9)$$

and

$$J_C := .6 \cdot 10^{-3} \cdot \frac{A}{\mu m} \quad (10).$$

K_N can be found by first extracting K_{NP} from:

$$K_{NP} := \frac{1}{\left(V_{DD} - V_{TN}\right)} \cdot \frac{1}{A} \cdot \left[\frac{2 \cdot V_{TN}}{\left(V_{DD} - V_{TN}\right)} + \ln\left[4 \cdot \frac{\left(V_{DD} - V_{TN}\right)}{V_{DD}} - 1 \right] \right] \quad (11)$$

Then solve for K_N:

$$K_N := \frac{W_N}{L_N} \cdot K_{NP} \quad (12).$$

In this case, the minimum width of the metal 1 power lines (W_{M1}) are .456µm. This is almost half of the minimum allowable metal1 width of .9µm! We are well within our specification but we will use 1.8µm because other circuits will share these power lines. The parameters are listed in Table 6.

Determining the Number of Parallel Transistors to use:

Now that we have our metal line widths specified that means there is approximately 4µm of height that we cannot use. There is an additional 4µm of height that is unavailable, due to other spacing requirements. The width of the NMOS transistors is given by:

$$W_{NF} := \frac{H_{CELL} - \text{Spacing}}{1 + \text{Ratio}} \quad (13)$$

The number of fingers is given by:

$$N_F := \frac{W_N}{W_{NF}} \quad (14)$$

In our example, this turns out to be 4.576. We need to use an integer number of fingers so we pick N_F to be 5 and then solve for the width of the NMOS parallel transistors to be 10.46μm. Using the ratio of 1.8, we find the width of the PMOS parallel transistors to be 18.83μm. The parameters are listed in Table 6.

Summary:
We have designed our metal1 power line widths to be 1.8mm, and the number of parallel transistors to be 5. The width of the NMOS transistors will be 10.46μm while the PMOS transistors will be 18.83μm. Next, we will layout our inverter.

Table 6: Parameters Used in Floor planning.

Parameter	Description	Value	Units
W_N	Width of the NMOS	52.3	μm
W_P	Width of the PMOS	93.9	μm
W_{NF}	Width of the parallel NMOS transistors.	Use equation 13.	μm
W_{PF}	Width of the parallel NMOS transistors.	Multiply W_{NF} times Ratio	μm
H_{CELL}	Maximum cell height	40	μm
Spacing	Height not available for transistor widths.	8 in this example.	μm
NF	Number of parallel transistors	See example.	integer
W_{M1}	Minimum width of a metal1 line due to electromigration considerations	Use equation 8	μm
I_{ave}	Average current used in CMOS inverter	Use equation 9	A
J_C	Maximum current density allowed before electromigration becomes a problem.	.6	mA/μm
K_N	Transconductance of NMOS scaled by W/L	Use equation 12	A/V^2
K_{NP}	Transconductance of NMOS scaled by W/L	Use equation 11	A/V^2
τ	Rise Time and Fall Times	100p	S
f_{CLK}	Clock Frequency	1G	Hz
V_{DD}	Supply Voltage	5	V
V_{TN}	NMOS threshold Voltage	.6	V
A	Fitting parameter used to design WN	Use equation (2)	Ω
Ratio	Used to scale WP to give symmetric propagation delays	Use equation (3)	unit less

Layout of the Inverter:

Now that we have a preliminary floor plan we are ready to layout the inverter.

Start the tools in your cell directory by typing ***icfb &*** at the command line if you are not running the tool already.

First we need to create a layout view of our inverter. Goto the library manager and create a new cell view according to Figure 59.

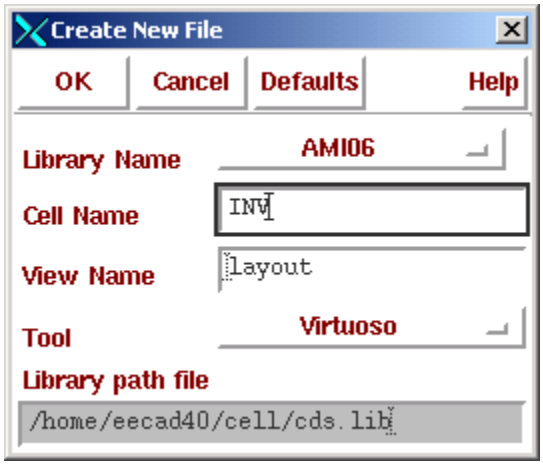

Figure 59: Creating a layout view of the inverter

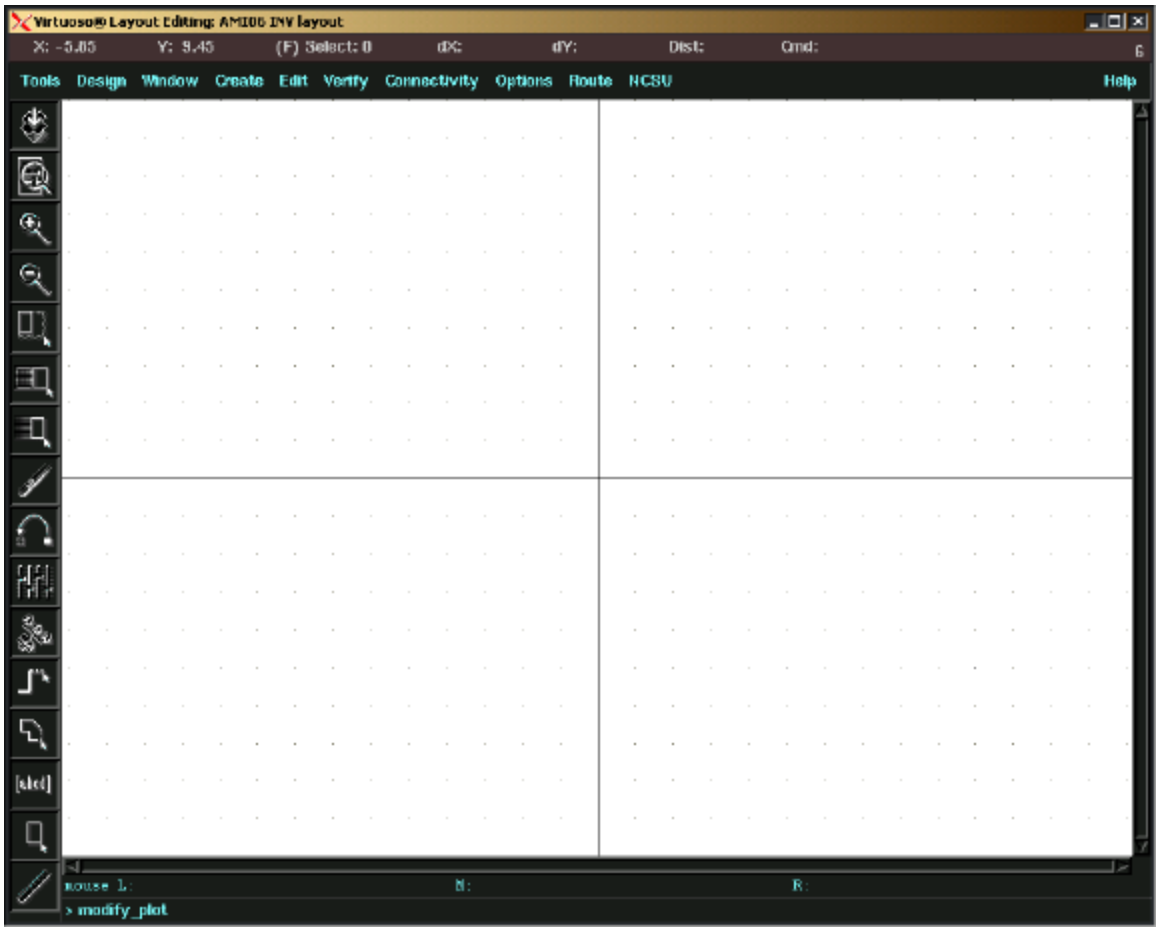

Figure 60: Layout Editor

Figure 60 shows the layout editor that you used in the cell design tutorial.

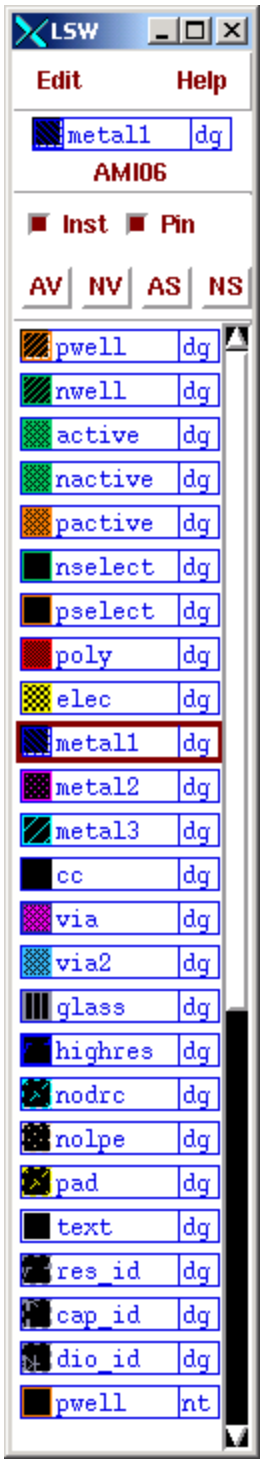

Figure 61: LSW

Notice that the layers that are available in the LSW (Figure 61). This is the AMI06 process.

To set the display so that all the layers will appear, goto Options... Display in the layout editor. Set the pop-up according to Figure 62 and click ok. Make sure the minor, major, x snap spacing and y snap spacing are correct as in Figure 62.

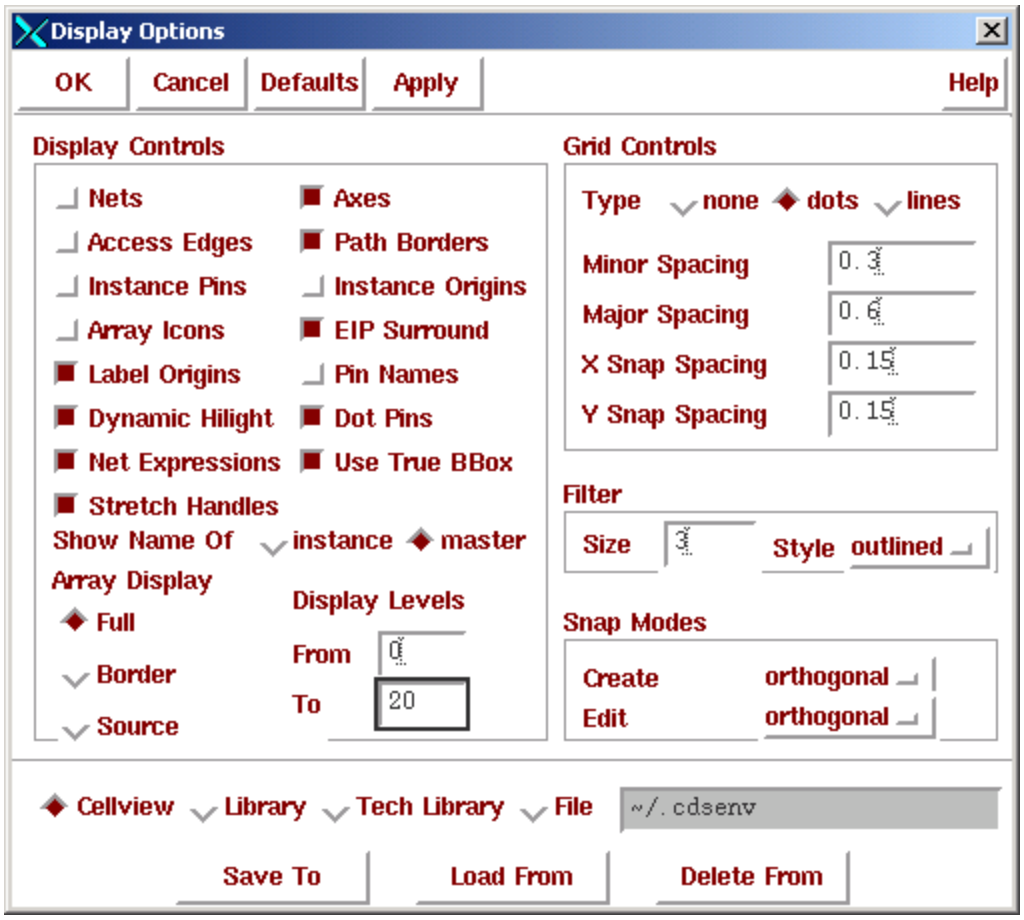

Figure 62: Setting Display Options.

To save these setting, in the CIW, goto options... save defaults. Click OK. It will take a few minutes to save the defaults, but it will save time later.

To make sure we fit in our cell height we will draw the VDD and ground lines first.

To draw the Ground line:

1. Select metal1 in the LSW.

2. Click on the layout window and press p to start a path. A pop-up like Figure 63 should appear. Change the width to 1.8 as in Figure 63 .

3. Start the path by click the left mouse button at the coordinate x=0, y=.9 (Figure 64).

4. End the path by double click at coordinate x=28.05, y=.9 (Figure 64).

5. Press the esc key to get out of path mode.

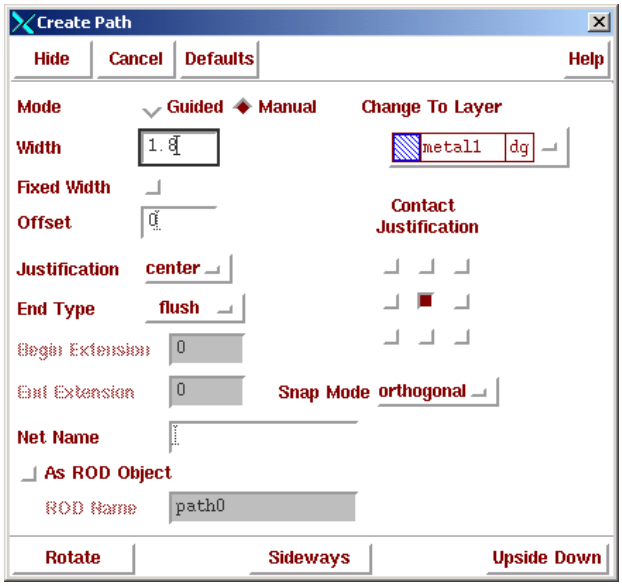

Figure 63: Starting a Path

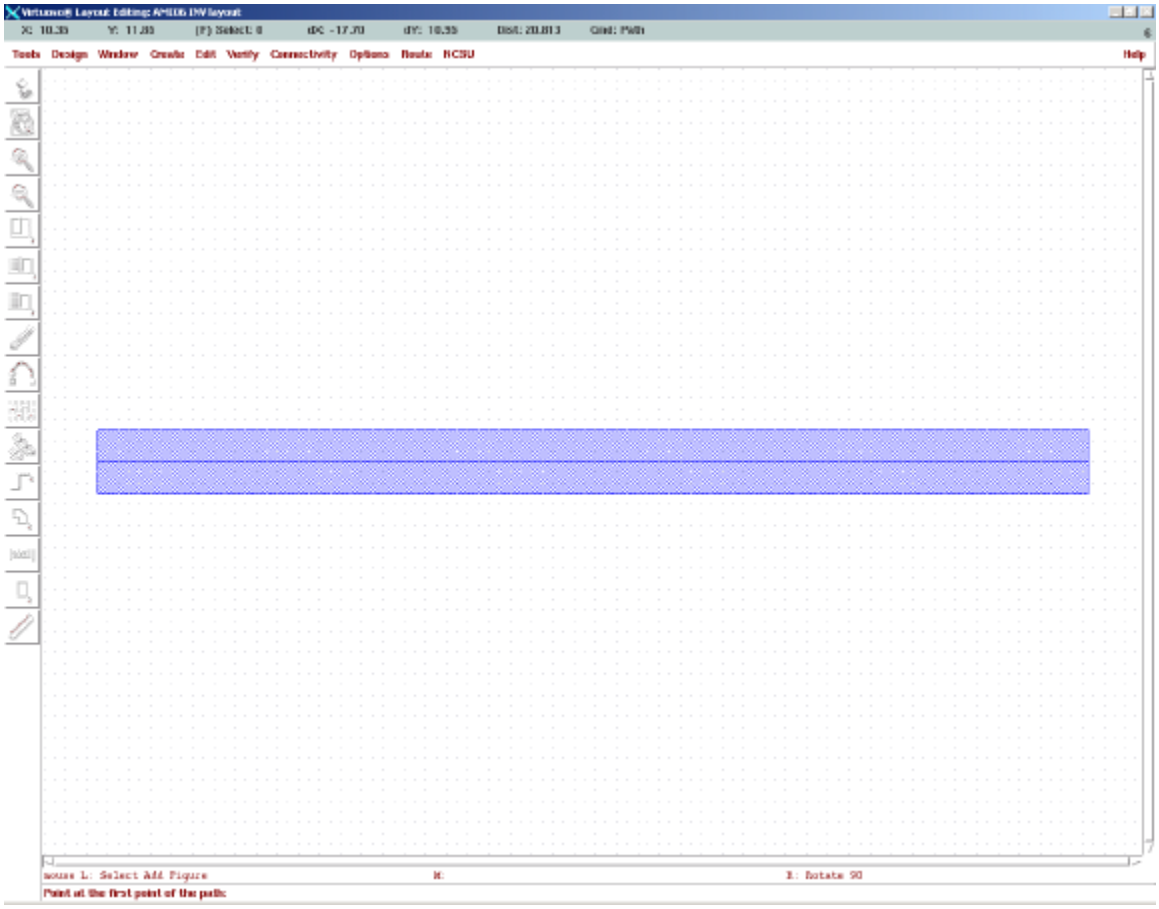

Figure 64: Drawing the Ground Line.

To draw the VDD line:

1. Press c to enter into copy mode.

2. Click on the ground line in the upper left corner and copy it to coordinate x=0, y=39.9 as in Figure 65.

3. Make sure the cell height is correct by measuring from the top left corner of the VDD line to the bottom left corner of the Ground line. You should get a cell height of 39.9 µm.

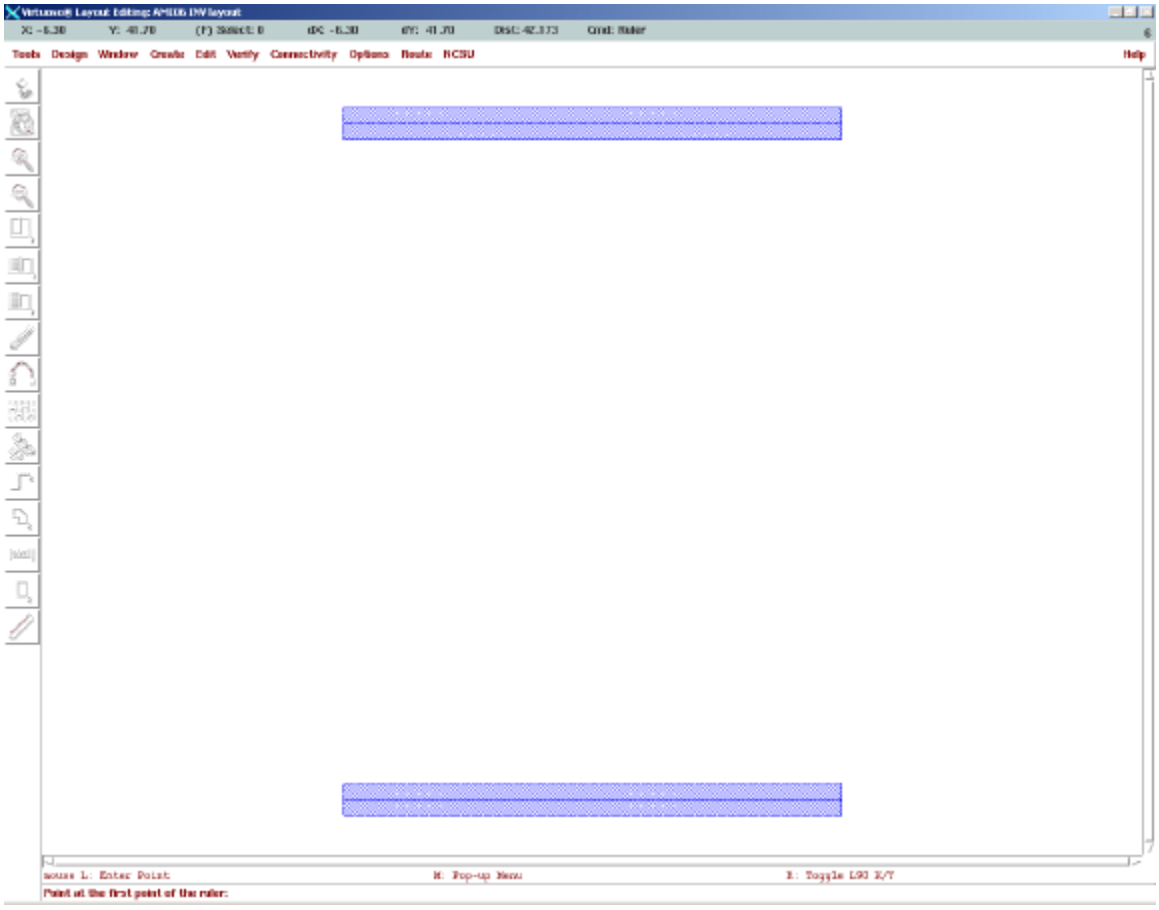

Figure 65: Inverter with VDD and Ground Lines.

Next, we will add the ntaps and ptaps:

1. Press the esc to get out of copy mode

2. Press i to bring up the Create Instance pop-up (Figure 66).

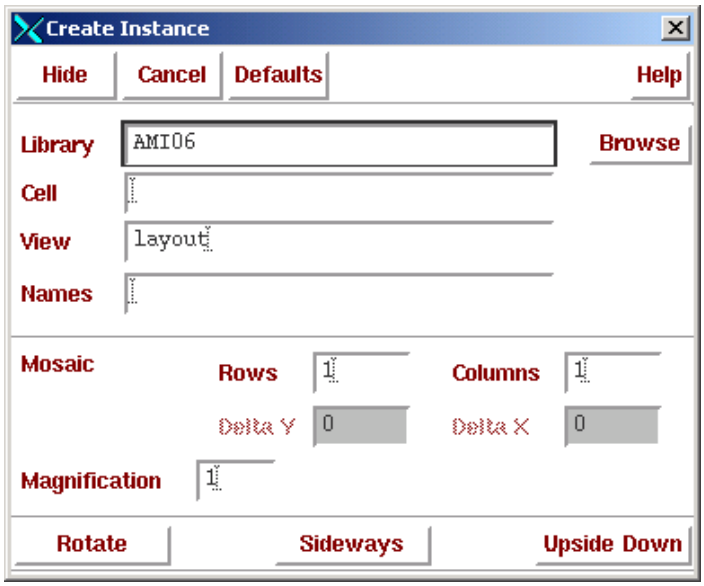

Figure 66: Create Instance Pop-Up.

3. Click on Browse and a pop-up like Figure 67 should appear.

4. Select ntap like in Figure 67.

5. Fill out the pop-up like Figure 68 making sure to select 7 rows of contact. This is done to make sure the NWELL is well contacted. Since it is impossible to route metal1 over this cell it does not matter if we use extra ntaps.

6. Stamp down the ntap so that the upper left corner is aligned with the lower left corner of the VDD line as in Figure 69.

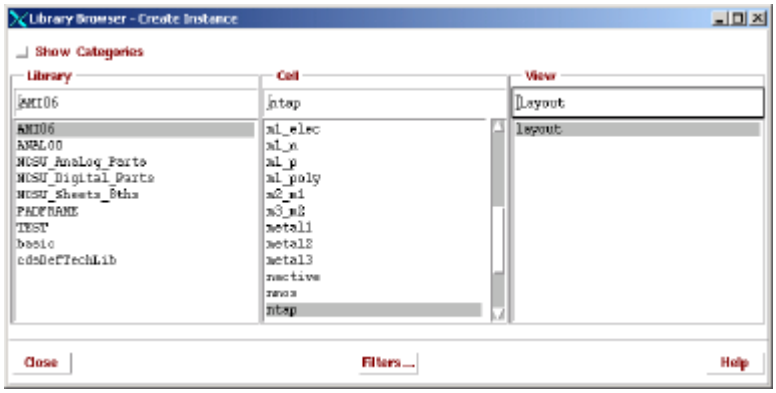

Figure 67: Browsing for the ntap.

7. Go back to the Create Instance pop-up and fill it our according to Figure 70.

8. Stamp it down like in Figure 71.

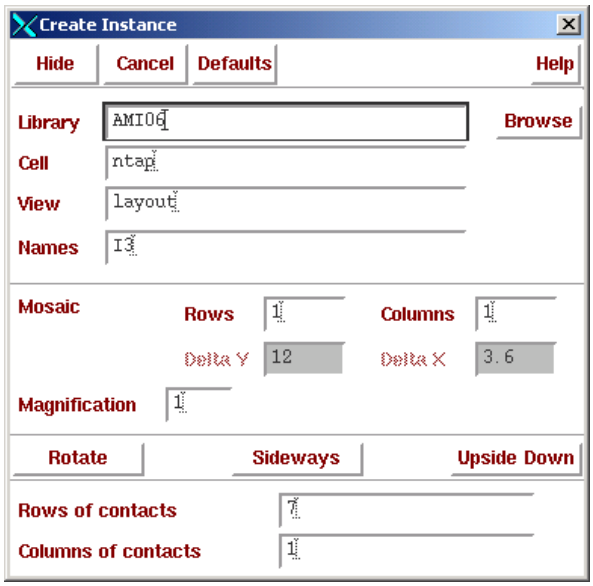

Figure 68: ntap with 7 Rows of Contacts.

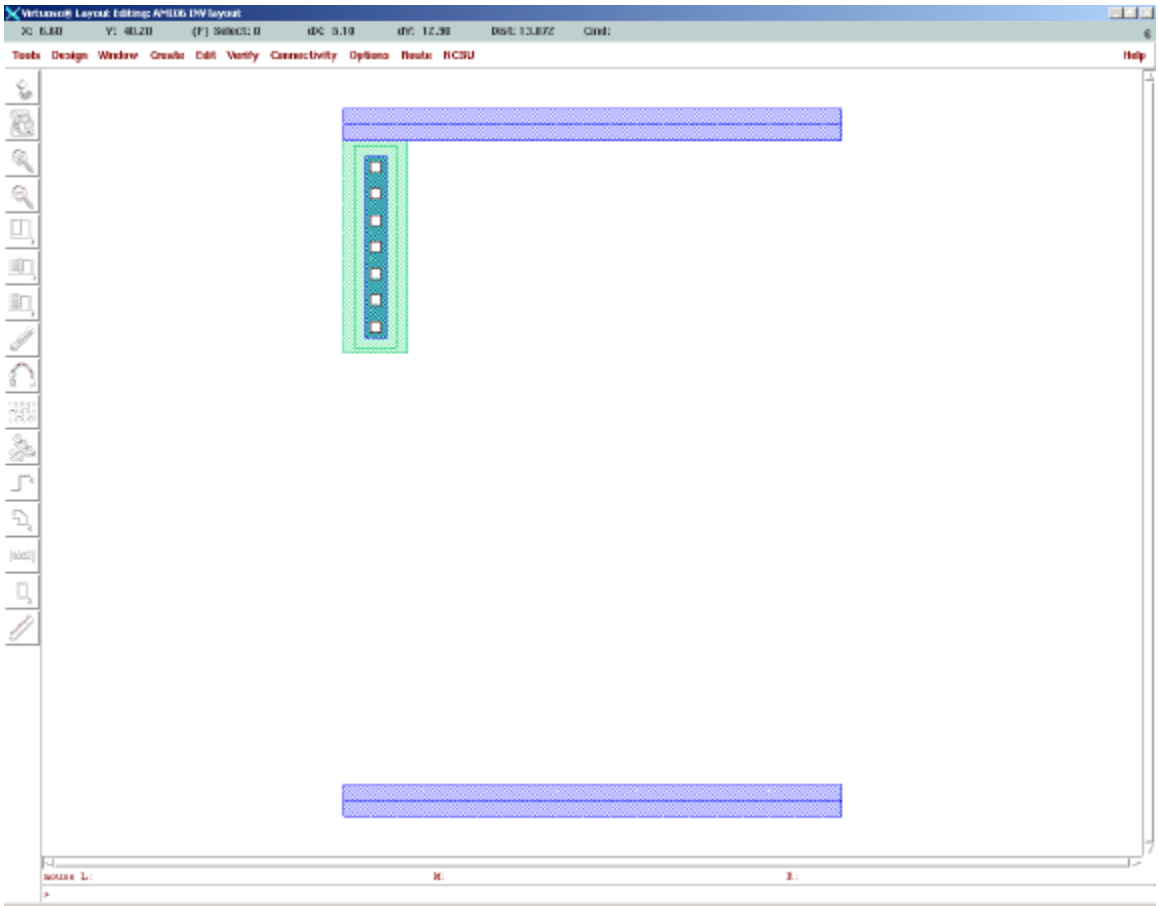

Figure 69: Placement of the ntap.

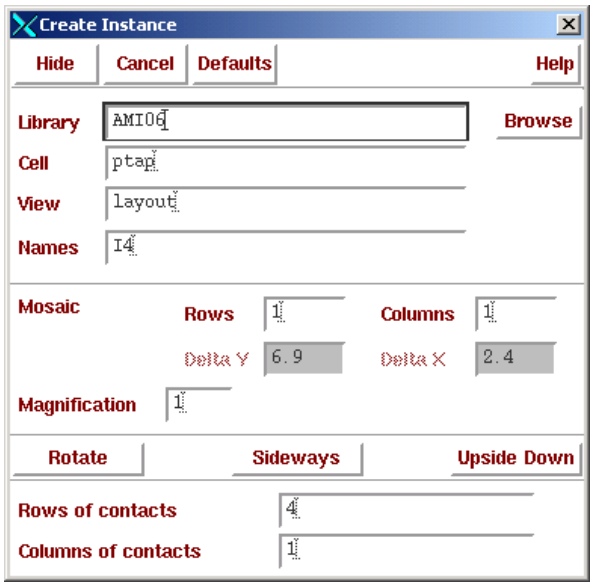

Figure 70: Ptap Pop-up.

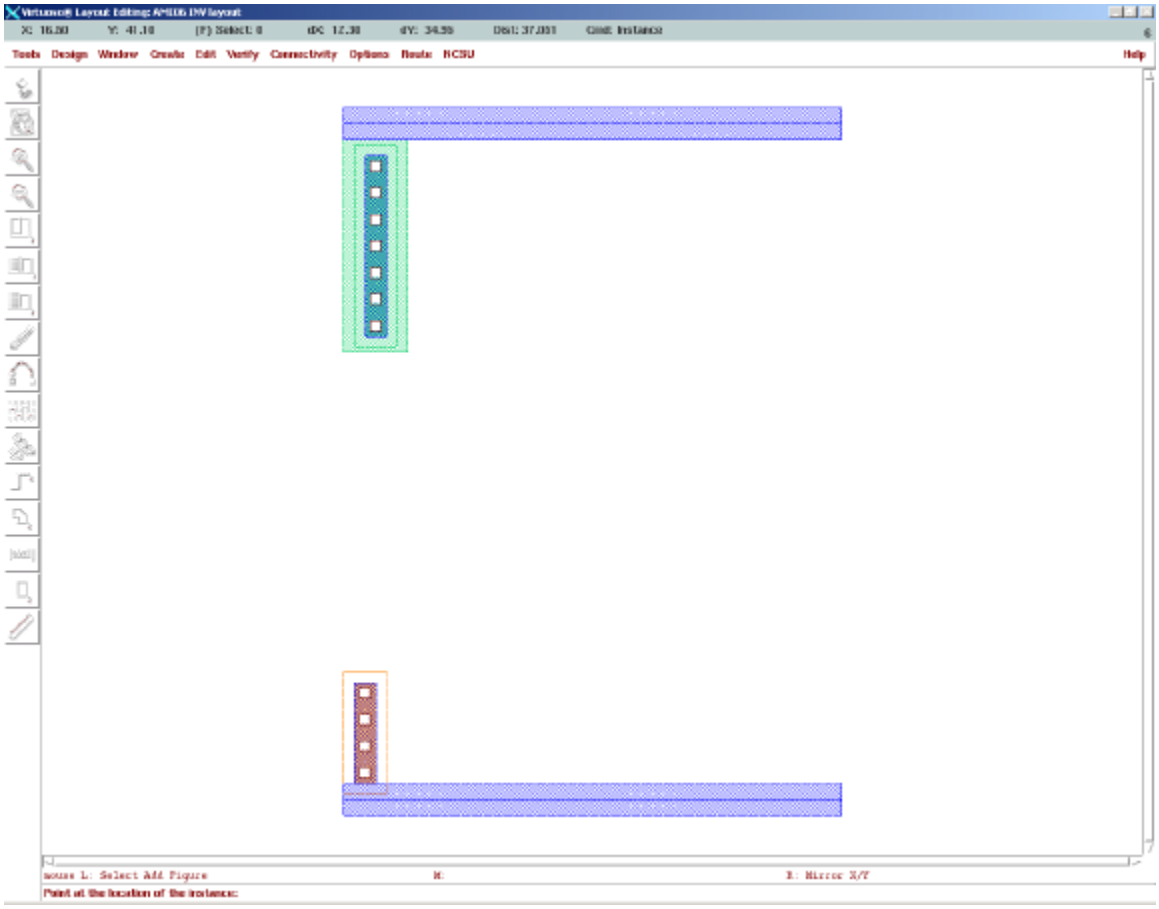

Figure 71: Placement of the ptap.

Viewing cells as an outline only.

Sometimes when doing a lyout you only want to work on certain levels of the design and seeing all the levels at any given time will make the screen refresh too slow. To tunr off the complete drawing of cells:

Go to Options... Display and a pop-up like Figure 72 should appear Set the display levels from 0 to 0 like in Figure 72 and click apply. You layout should look like Figure 73. Go back to the display and change the display levels back to 0 to 20 to finish the rest of the tutorial.

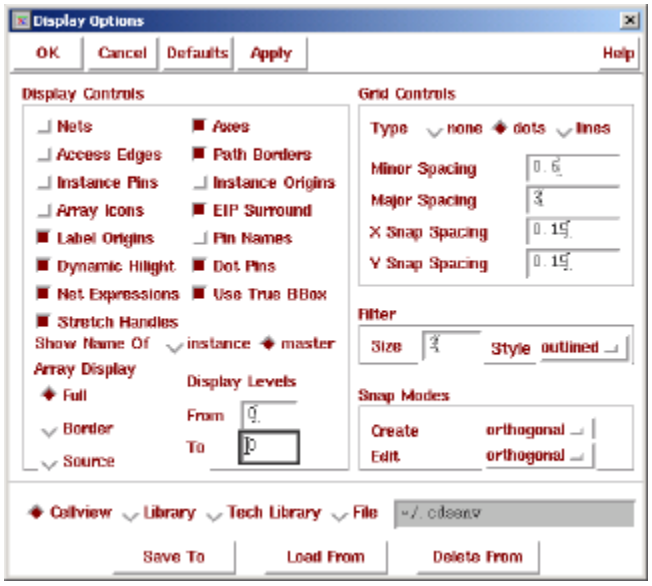

Figure 72: Setting the Display Levels.

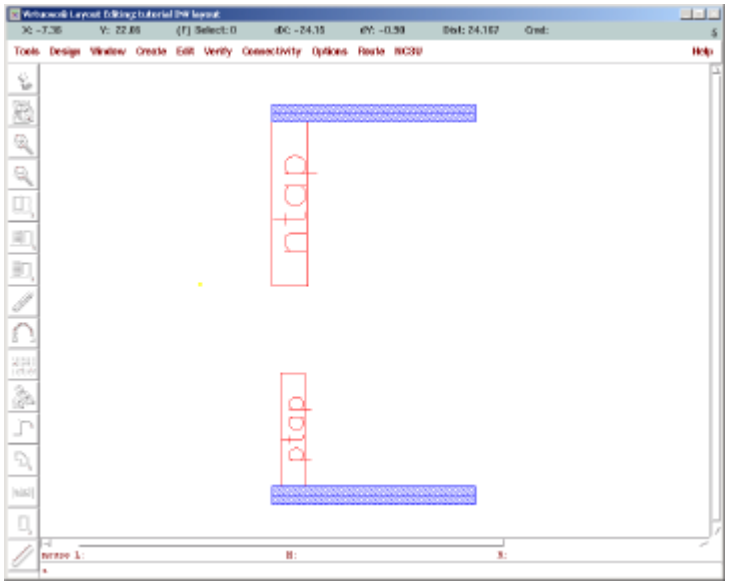

Figure 73: Viewing cells in Outline Only.

To add the PMOS and NMOS Transistors:

1. For the PMOS transistor, in the Create Instance pop-up, fill it out like Figure 74. Note the width is rounded to 18.9u by the software. Make sure you set the multiple to 5.

2. Stamp it down like in Figure 76. (The top left corner of the transistor is at x=3.6 and y=38.10.

3. For the NMOS transistor, in the Create Instance pop-up, fill it out like Figure 75.

4. Stamp it down like in Figure 76. (The top left corner of the transistor is at x=4.86 and y=13.8.

5. Run DRC and fix any errors.

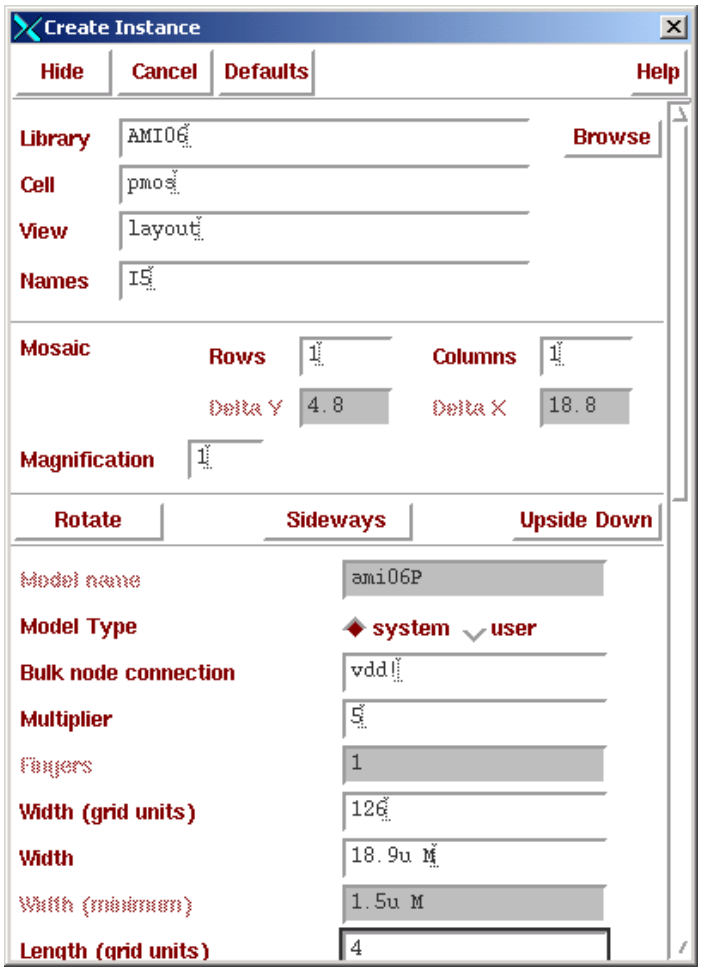

Figure 74: PMOS Transistor with a Multiple of 5.

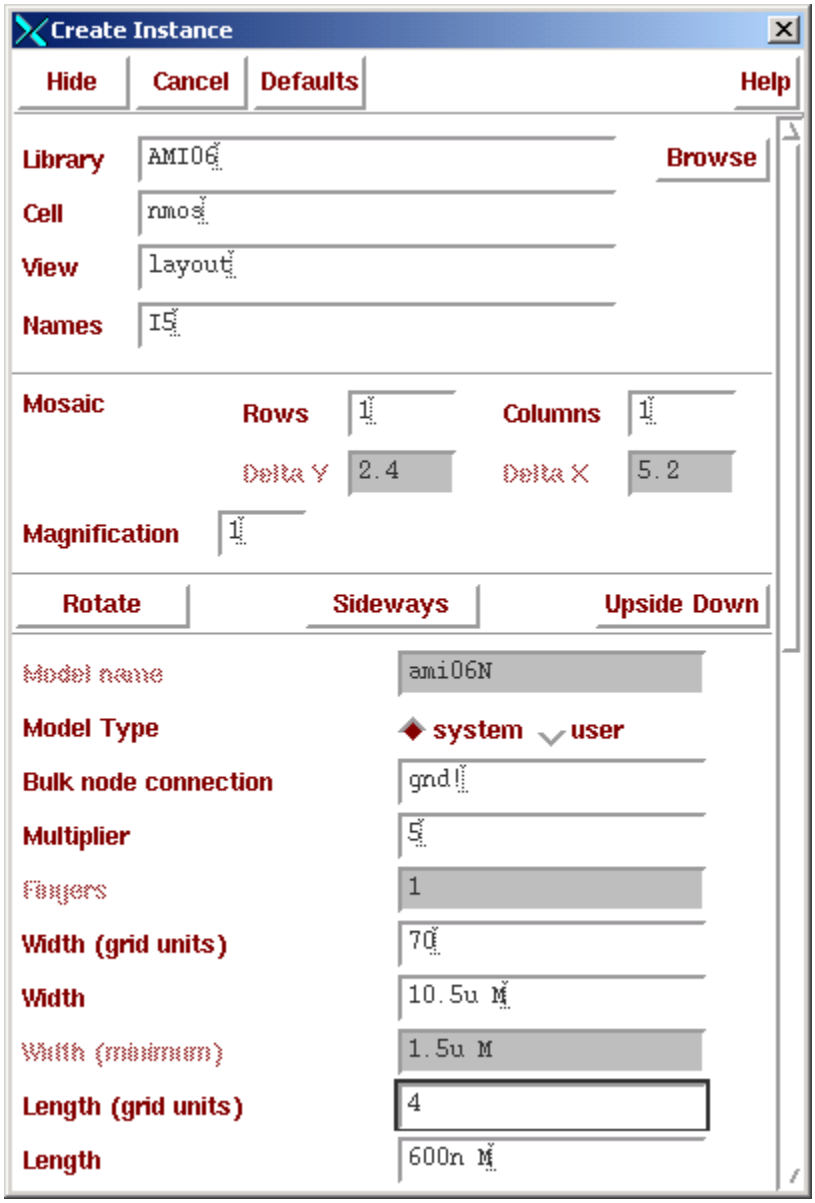

Figure 75: NMOS Transistor with a Multiple of 5.

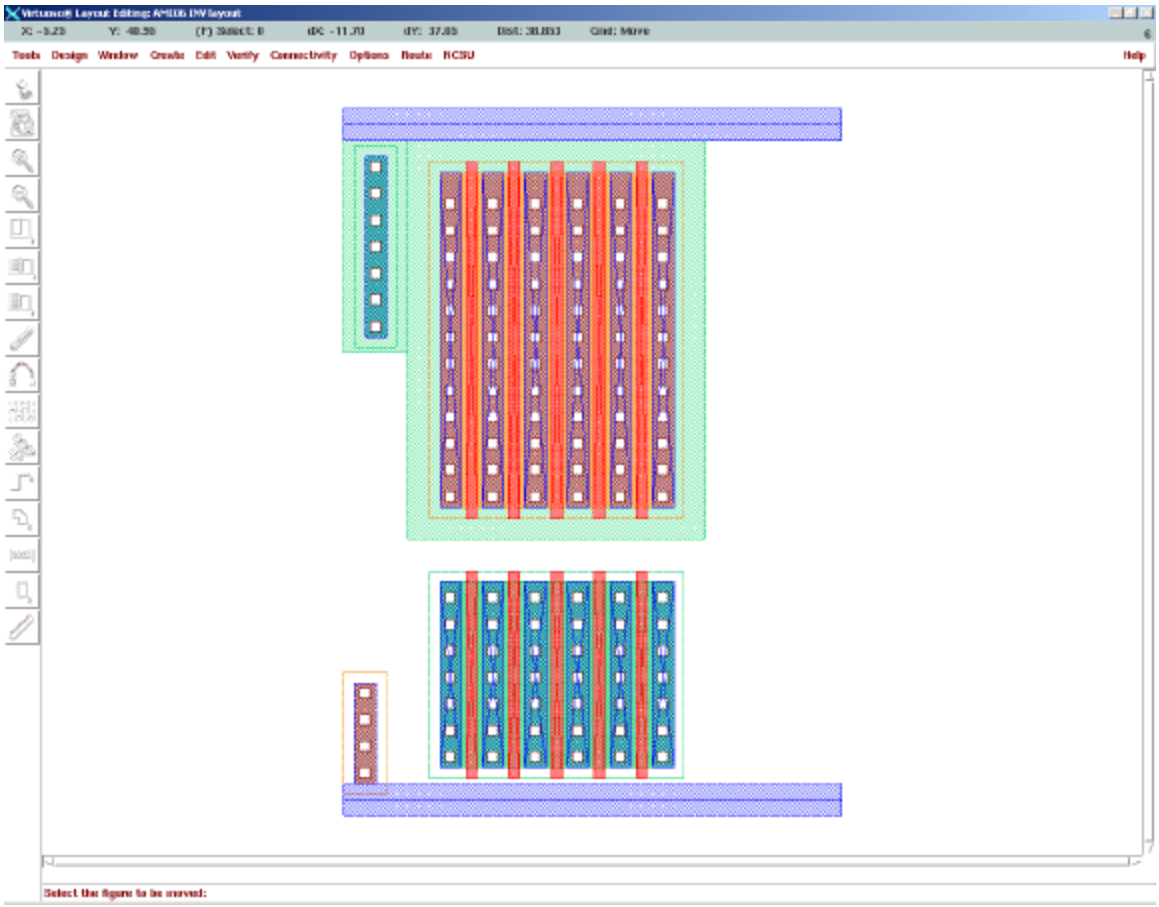

Figure 76: Placement of the NMOS and PMOS Transistors.

Adding the Input Layer:

In this case, the input will come from a metal1 line. We need to form a metal1 to poly contact to define were this input will be.

1. For the metal1 poly contact, in the Create Instance pop-up, fill it out like Figure 77. Note the width is rounded to 18.9u by the software. Make sure you set the rows and columns to 2. This will ensure that even if there is a bad contact the input will still be connected. In addition, the resistance of the contact will be lower than the minimum sized device.

2. Stamp it down like in Figure 78. The top left coordinate is: x=.45, y=16.65.

3. Run DRC and fix any errors.

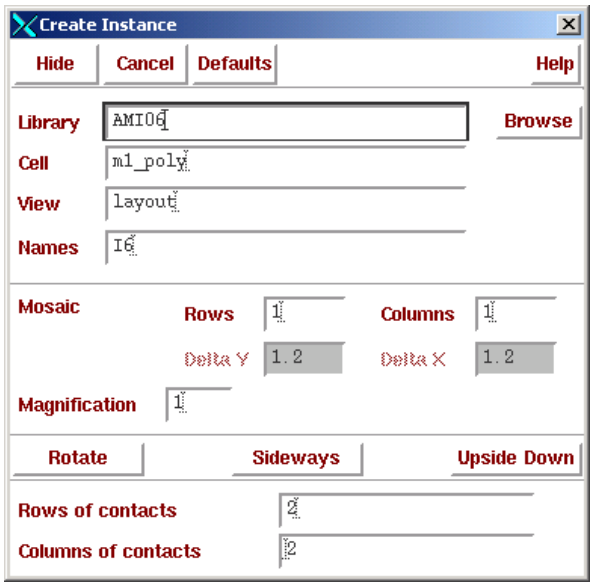

Figure 77: Metal1 to Poly Contact.

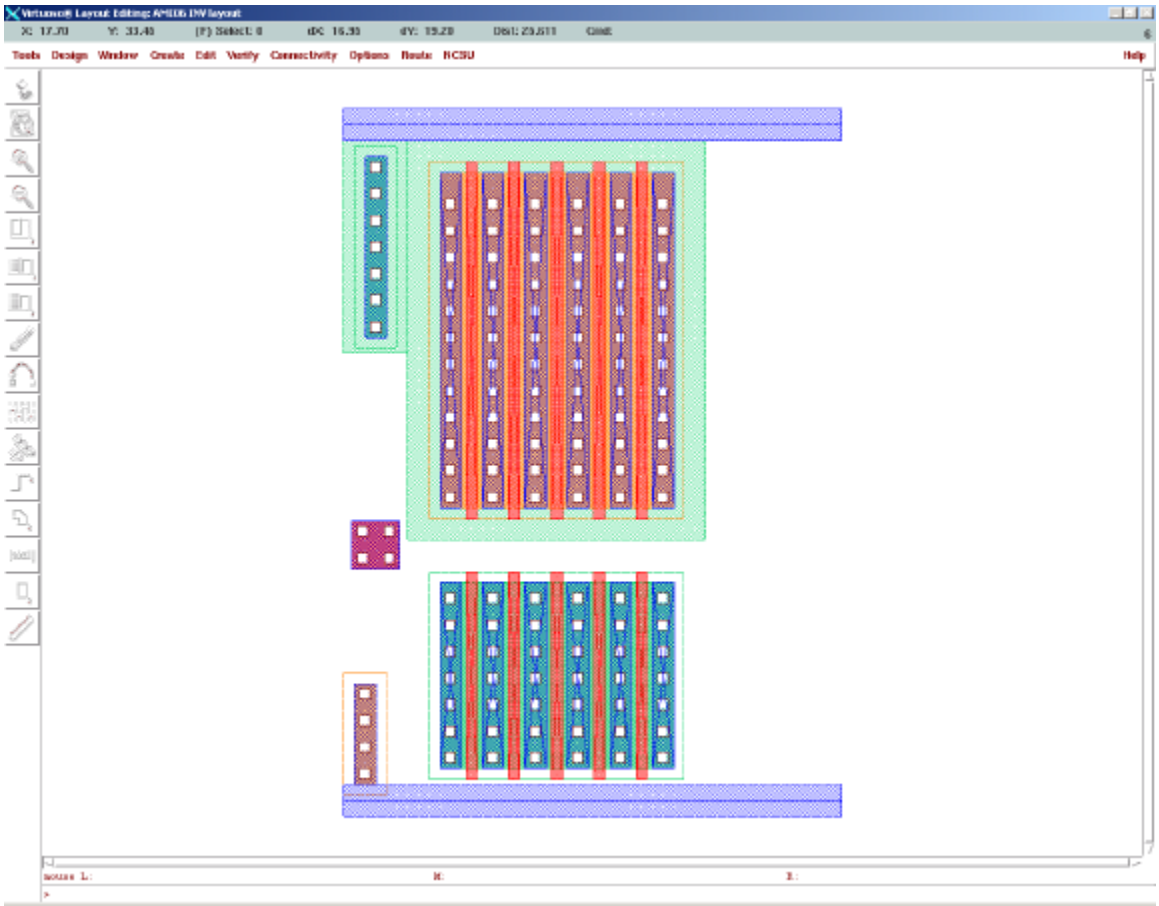

Figure 78: Placement of the Metal1 Poly Contact.

Wiring up the Transistors with Metal1:

1. Zoom in on the top of the layout

2. Create paths of 1.2µm wide connecting the ntap and every other S/D of the PMOS transistor as in Figure 79. Note: For all layers except metal1 have been turned off.

3. Repeat the same process the Sources of the NMOS transistors (Figure 80).

4. Repeat the same process to connect the Drains for the out put (Figure 81).

5. Run DRC and fix any errors.

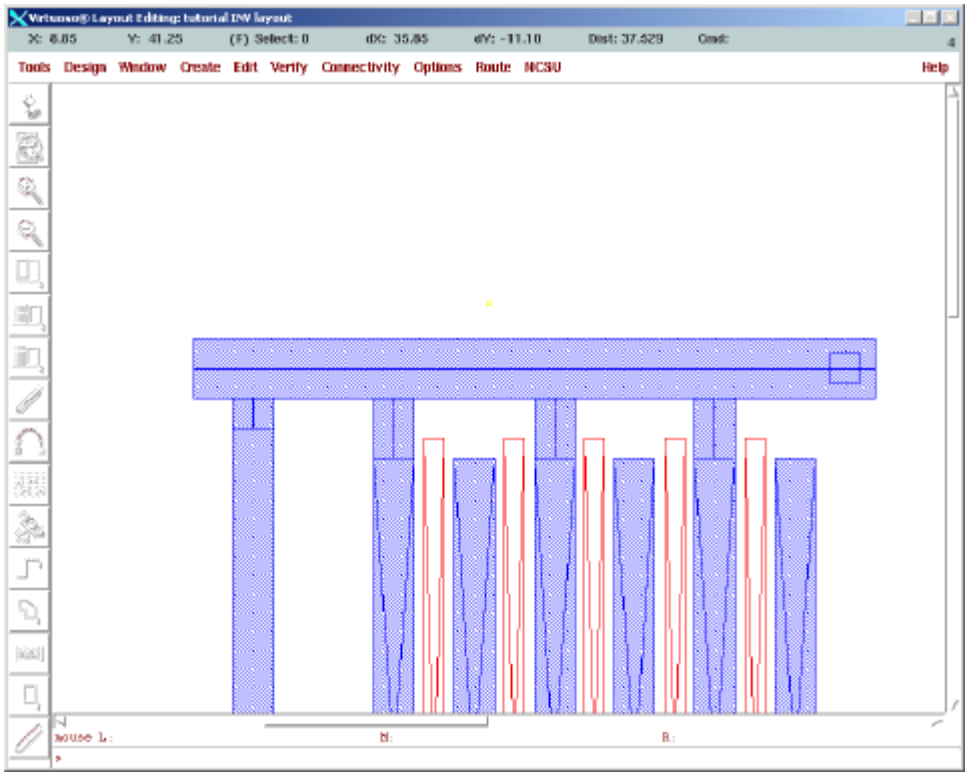

Figure 79: Connect Sources of the PMOS Transistor and the ptap.Figure 79.

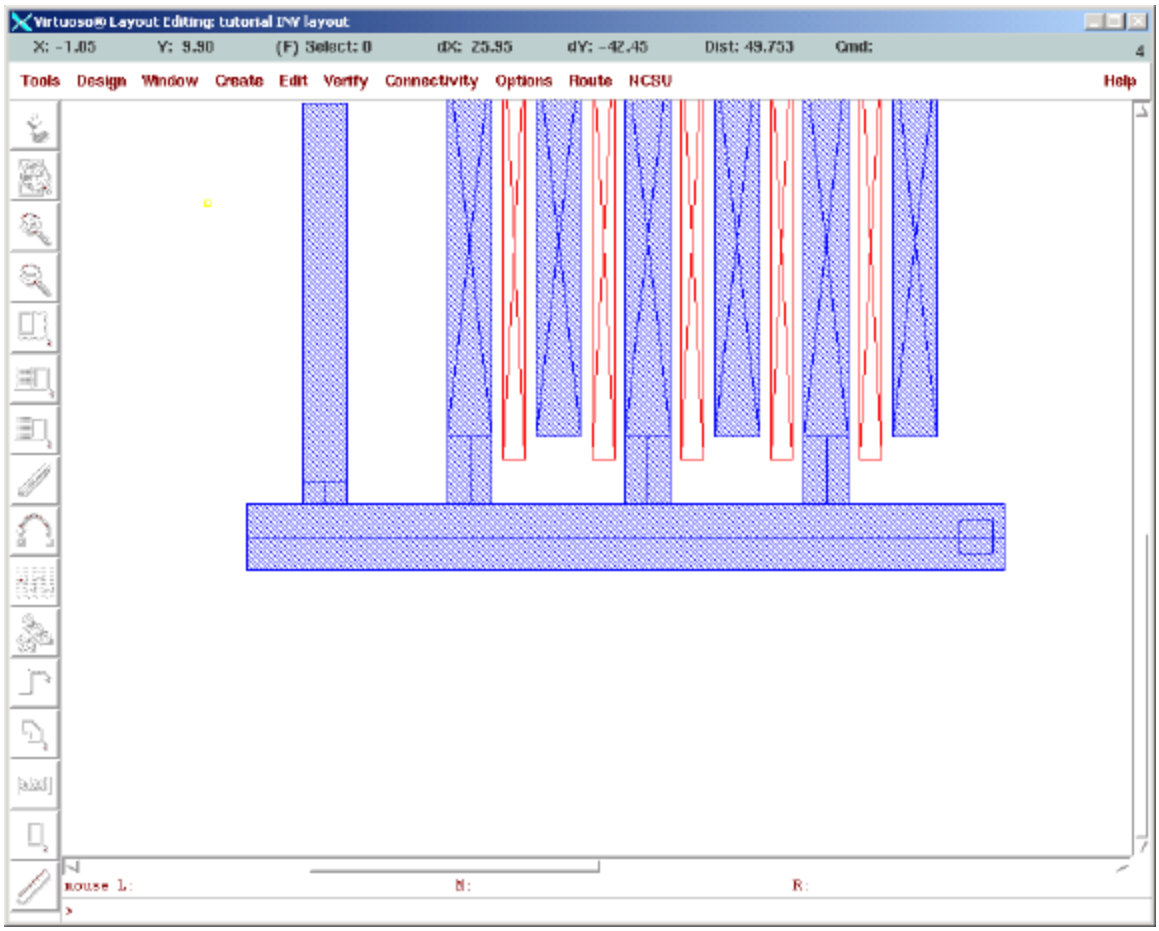

Figure 80: Connect Sources of the NMOS Transistors and ptap.

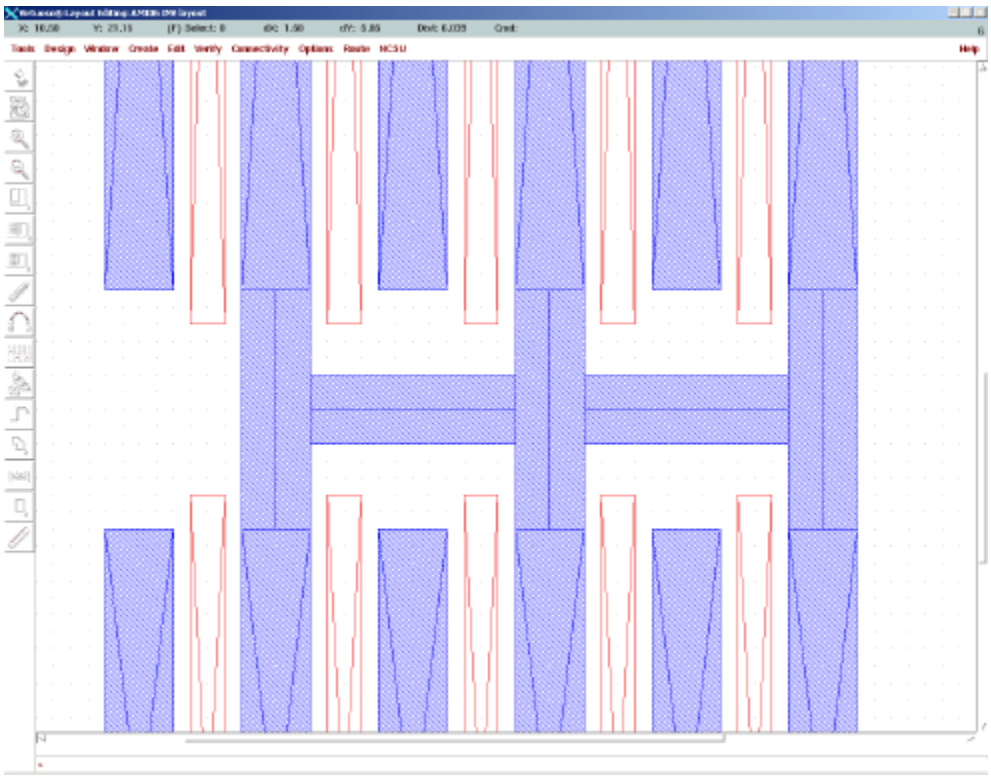

Figure 81: Connecting Drains to form the Output.

Connecting the input to all the gates:

Now we need to wire up the gates with poly so they are all connected.

1. Set the display in the LSW to show only poly.
2. Connect the gates of the NMOS and PMOS Transistors to each other and the input with one large rectangle as in Figure 82.
3. Run DRC and fix any errors.

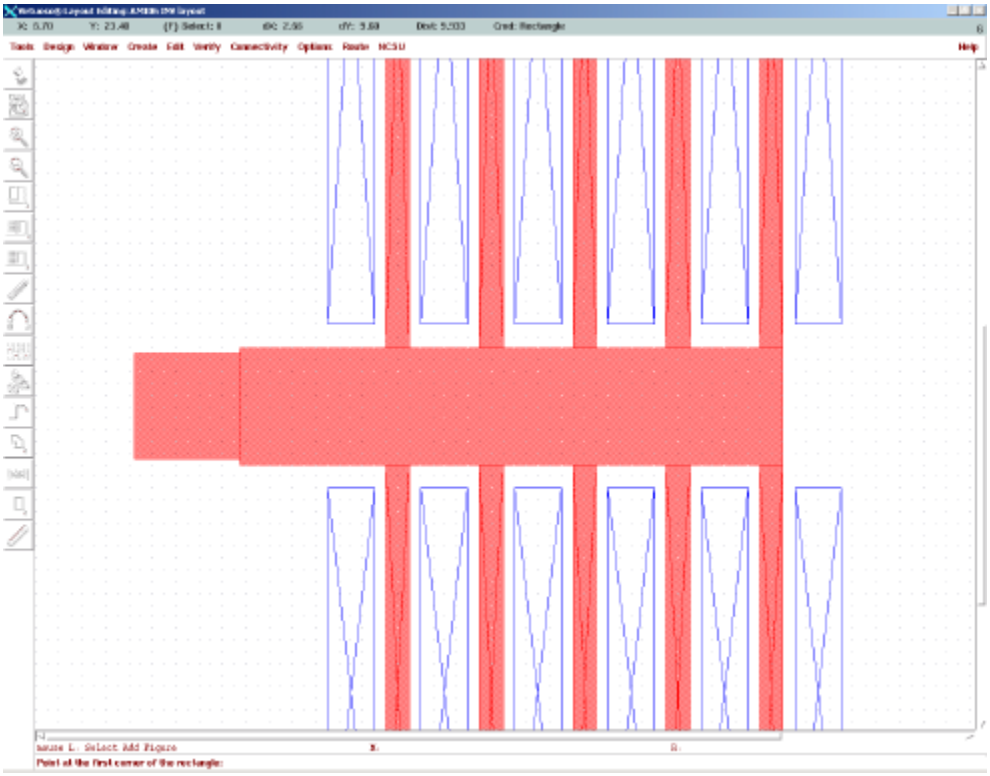

Figure 82: Connecting up the Gates of the Transistors to the Input.

The finished layout (with out any pin information) should look like Figure 83. Note: Stretch the right ends of the metal1 power lines to line up with the right side of the NWELL in Figure 83.

Run DRC one final time to make sure you have found all the errors. You might not see all the errors in the layout if they are small. Read the report in the CIW to make sure no errors were generated (Figure 84).

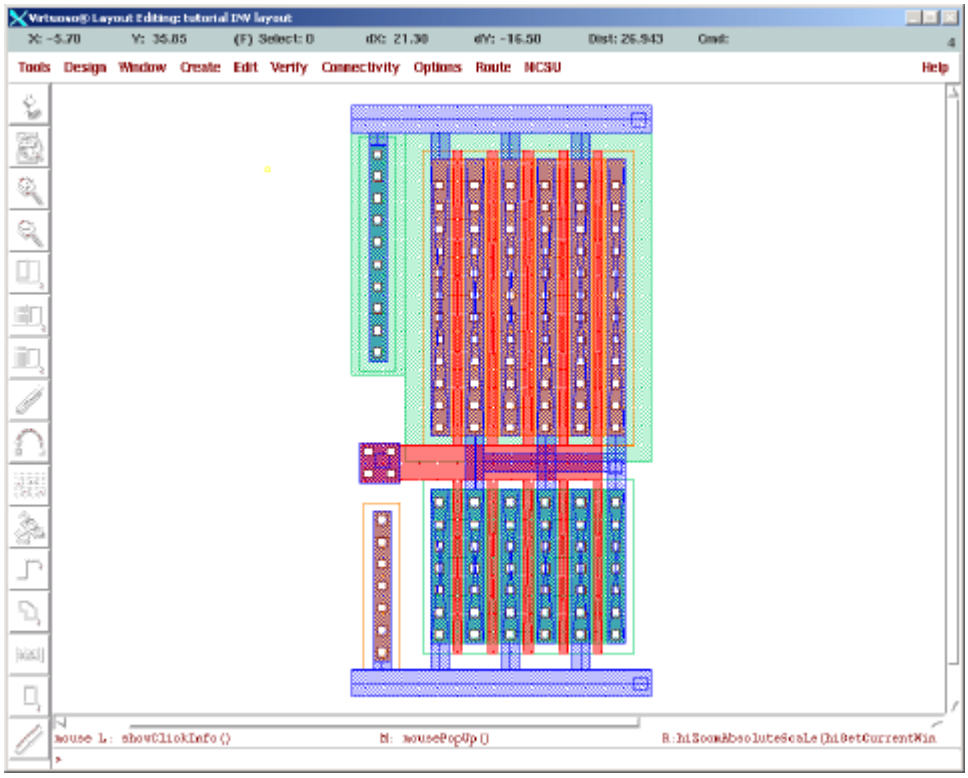

Figure 83: Layout of CMOS Inverter.

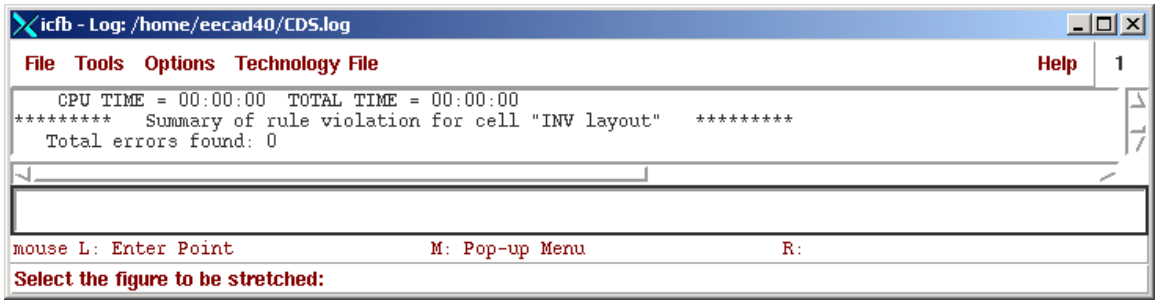

Figure 84: DRC with no Errors.

Adding Pins:

In order to simulate our extracted view of your laid out inverter, you need to add four pins, vdd!, gnd! A and Y. The symbol "!" means that the variable name is global.

Goto Greate... Pin and fill out the pop-up according to Figure 85. You must enter the pin names exactly as in the pop-up or they will not match the schematic and symbolic views of the inverter.

Make sure the layer is set to metal1!

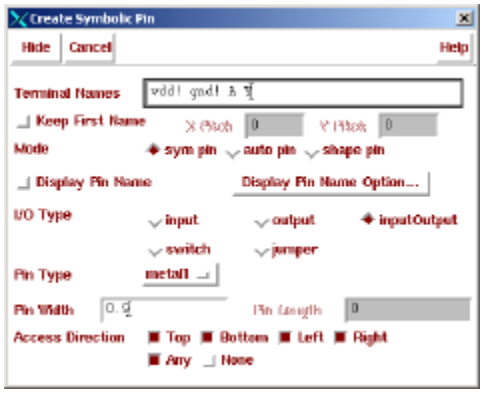

Figure 85: Pin Creation

Stamp down the vdd! pin in the metal power bar on the pmos side of your circuit.

Stamp down the gnd! pin in the metal power bar on the nmos side of your circuit.

In the Create Symbolic Pin pop-up change the I/O type to input like in Figure 86, and stamp it down in the metal1 poly via.

Figure 86: A input pin.

In the Create Symbolic Pin pop-up change the I/O type to output like in Figure 87 and stamp it down in the metal region connecting the two transistor drains.

The completed layout with pins can be seen in Figure 88.

We are now ready for the final DRC and then we can extract the circuit for LVS and post extraction simulation.

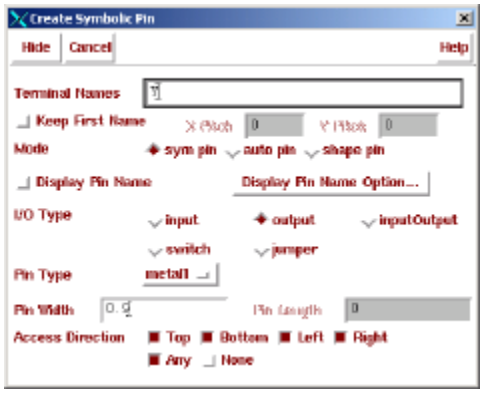

Figure 87: Y Out put pin.

The completed layout should look like Figure 88. Now you are ready for the final design rule checking.

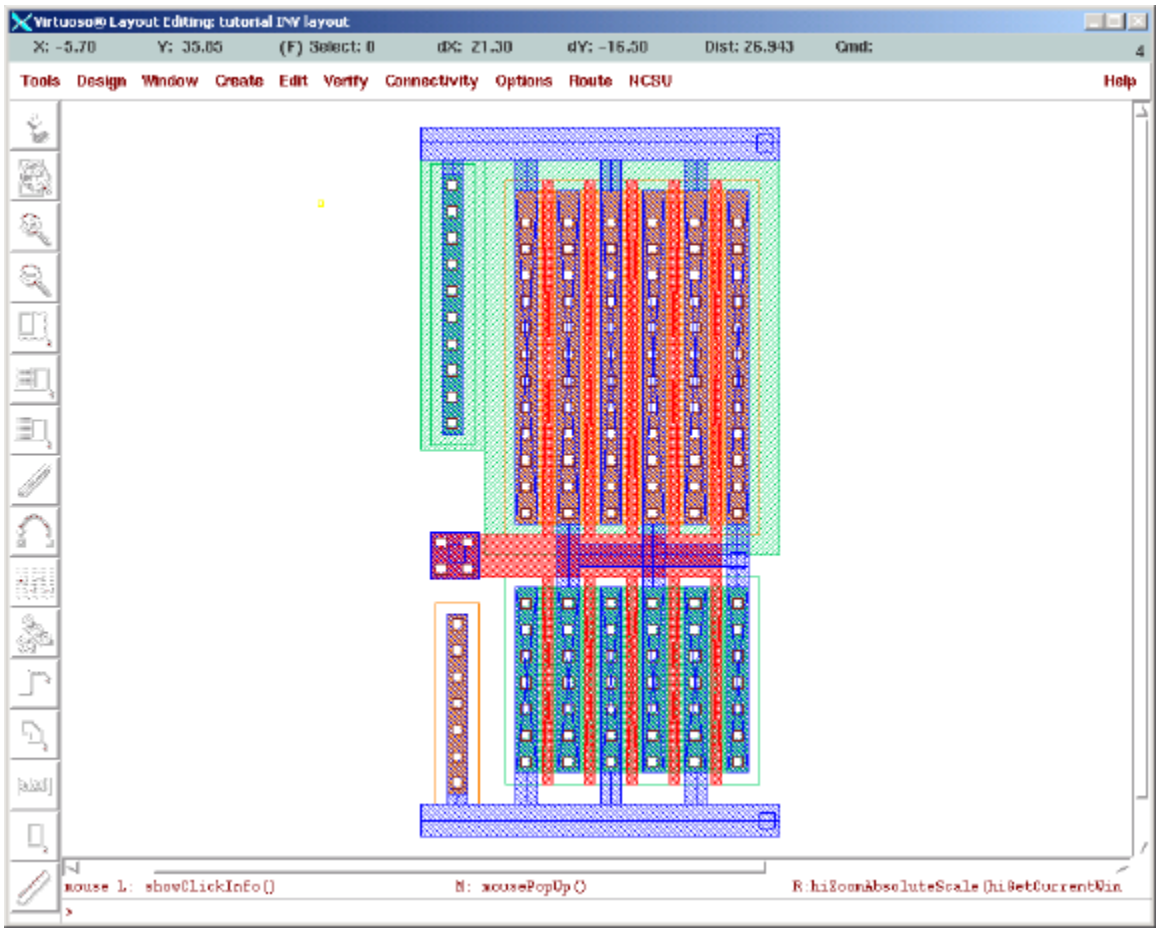

Figure 88: Completed INV Layout.

Design Rule Checking (DRC)

Goto Verify... DRC and a pop-up like Figure 89 should appear. You just have to click ok and it should run with no errors like in Figure 90.

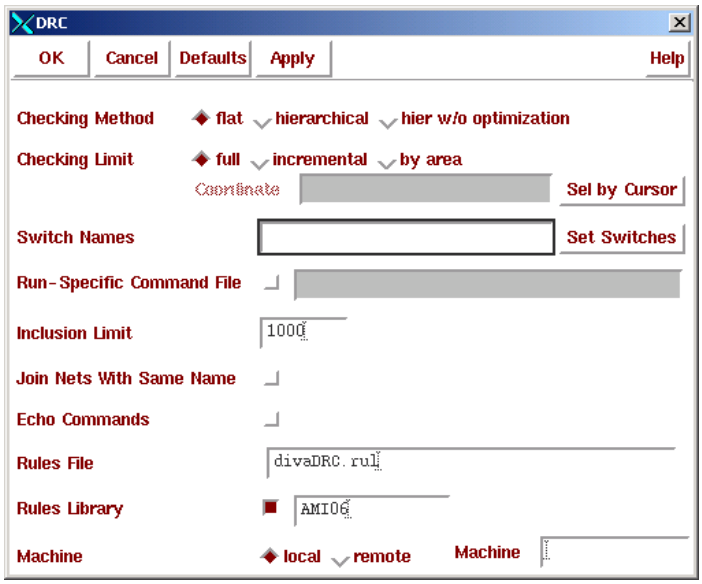

Figure 89: Running DRC

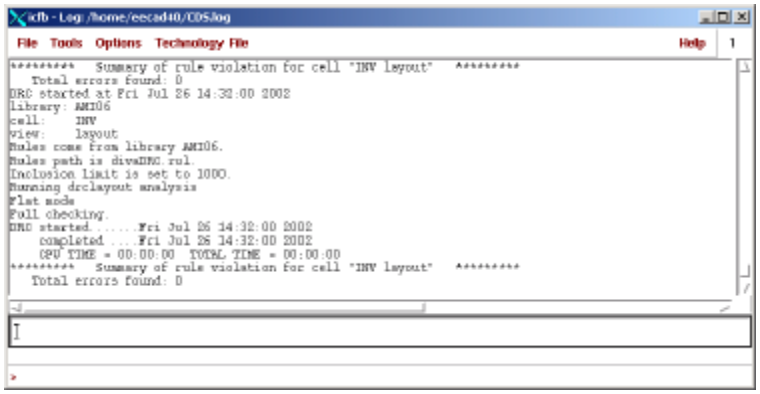

Figure 90: No DRC Errors.

If you have errors, fix them according to the AMI06 design rules with the method you learned in the cell design tutorial.

Once the circuit has no DRC errors save your work.

Summary:
We have drawn a layout view of our inverter and made sure that there were no drawing errors. We still need to know if the as drawn circuit will behave like an inverter according to our timing and power specifications. To do this we will extract the circuit and run an LVS check.

Circuit Extraction:

Now that your circuit is laid out with no DRC errors, it is time to check if it is an electrical equivalent of your inverter schematic. Also, we need to extract any parasitic capacitors from the layout that might affect our circuit's performance.

Goto Verify... Extract and a pop-up like Figure 91 should appear.

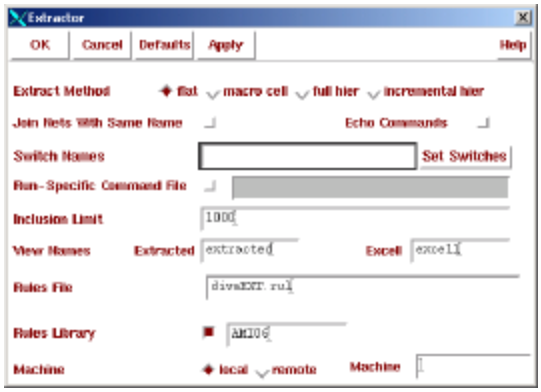

Figure 91: Running the extractor.

Click on the set switches button in the extractor pop-up and a list of choices should appear like in Figure 92.

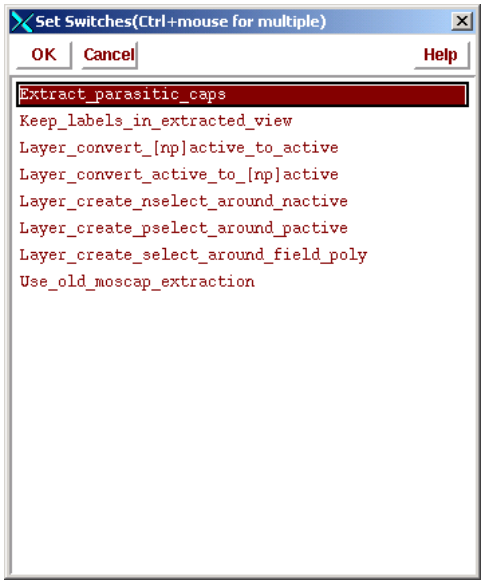

Figure 92: Selecting Extract parasitic capacitances.

Click on Extract_parasitic_caps and press ok.

Your extractor pop-up should look like Figure 93.

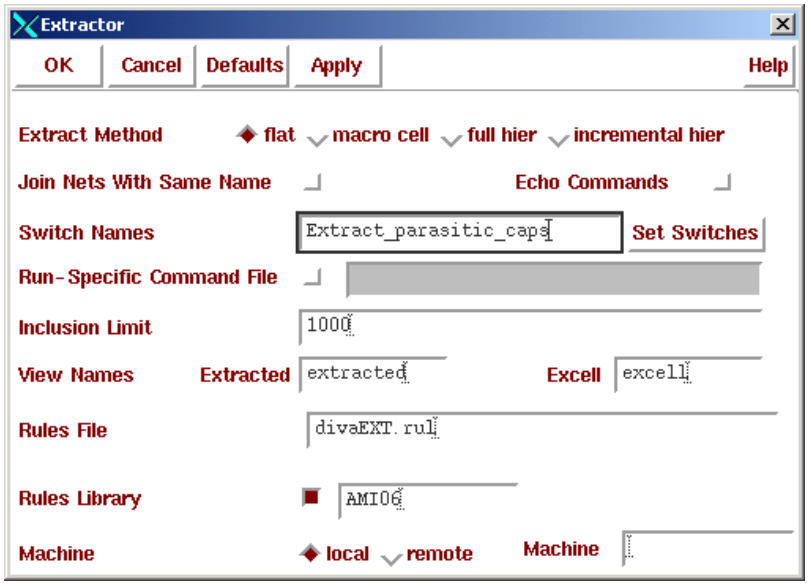

Figure 93: Extractor all set to go.

Click ok to run the extractor. The CIW should give no errors like in Figure 94.

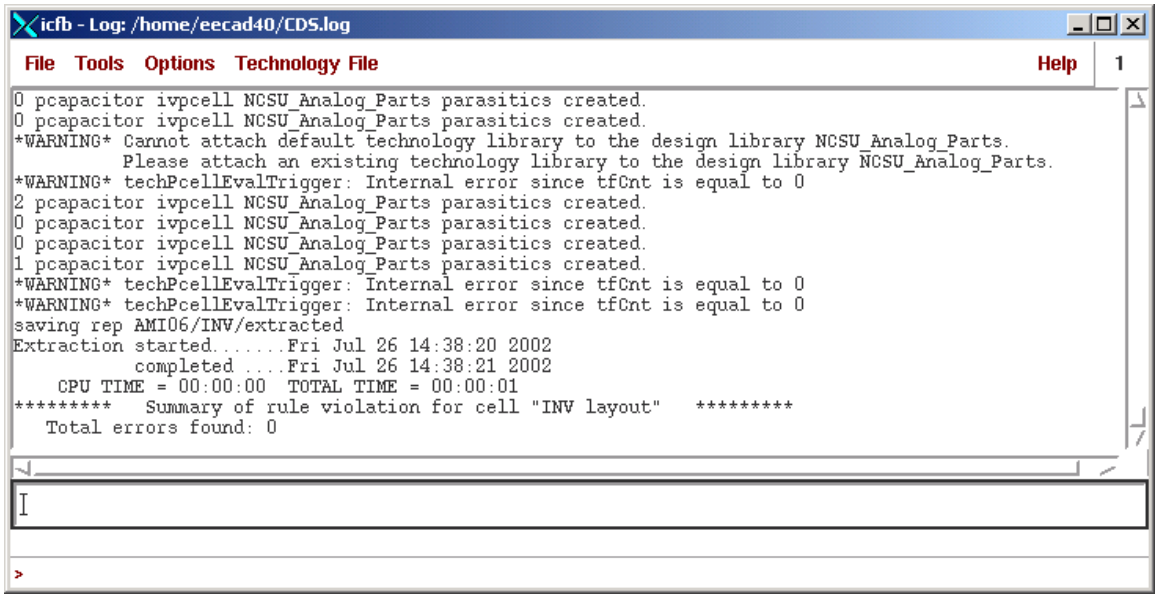

Figure 94: Extracted inverter with no errors.

To see the extracted circuit with parasitic capacitors open up the extracted view from the library manager (Figure 95).

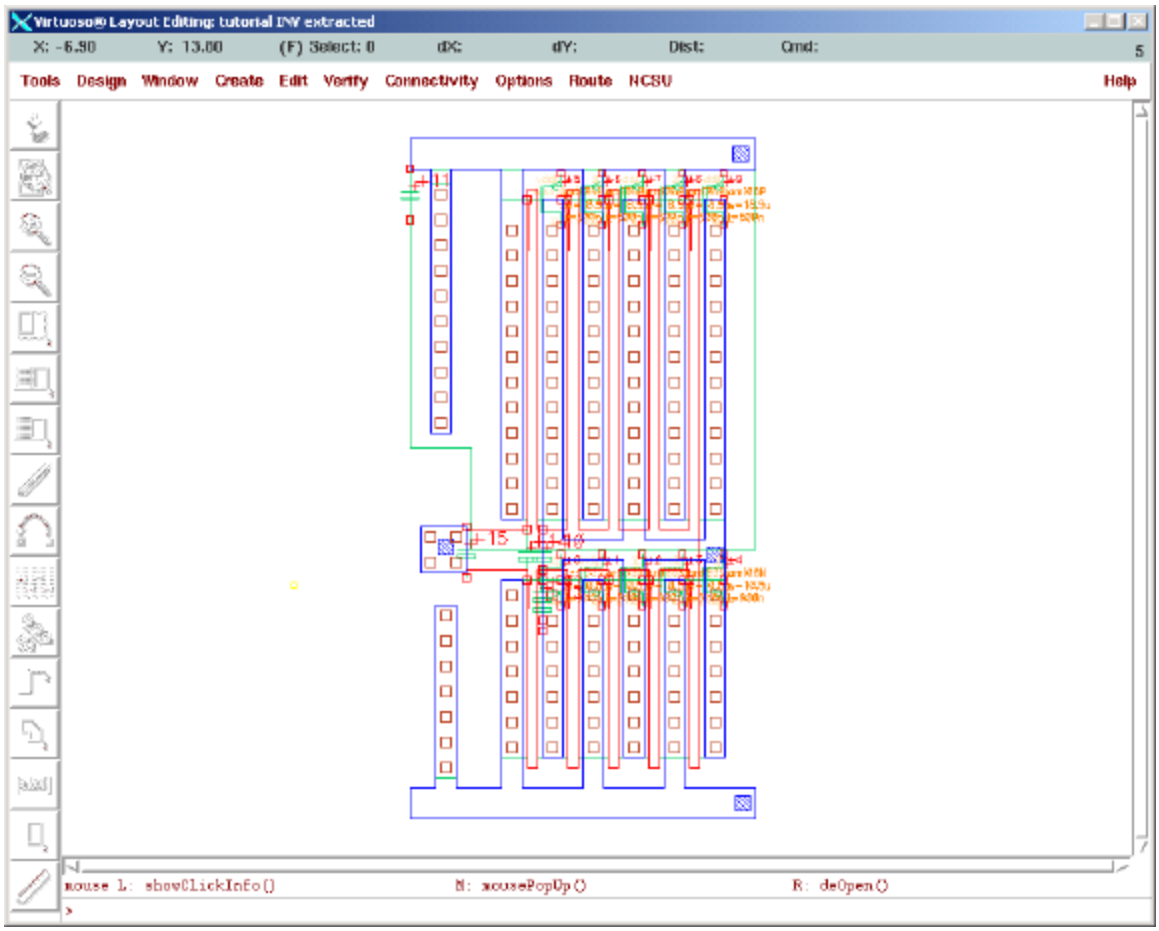

Figure 95: Extracted View of CMOS inverter.

Layout versus Schematic (LVS)

An LVS check makes sure that the circuit you laid out is equivalent to the one you entered into your schematic.

To run an LVS check, Goto Verify... LVS and a pop-up should appear like Figure 96.

Use the Browse button to select which schematic and which extracted view you are going to check for equivalence. You pop-up should be filled out just like Figure 96. Click run to start. This will take several minutes. When the LVS check is successful, a pop-up like Figure 97should appear.

If you have a large circuit to check set the priority to a higher number, so it does not suck up system resources too much. You can leave it 0 (highest priority) for now.

NOTE 1: If you think that having your job running at the highest priority will get your job finished quickly, think again. You could crash the system if too many users use too high a priority on an LVS job!

NOTE 2: Cause for the job to fail are incorrectly entering the data into the LVS pop-up. Sometimes you have to open up your schematic and resave it to get the LVS check to succeed.

NOTE 3: Just because the LVS job succeeded, does not mean your circuit passed an LVS check!

To make sure your circuit passed the LVS check click on output in the LVS pop-up. Looking at Figure 98, we see that all the nets match and that all the devices match.

If you have errors, it can be hard to figure out what they are from the report to see where the errors are:

1. Open up the extracted view of the inverter.
2. Click on Error Display in LVS.
3. A pop-up like Figure 99 should appear. Use it to locate your errors.

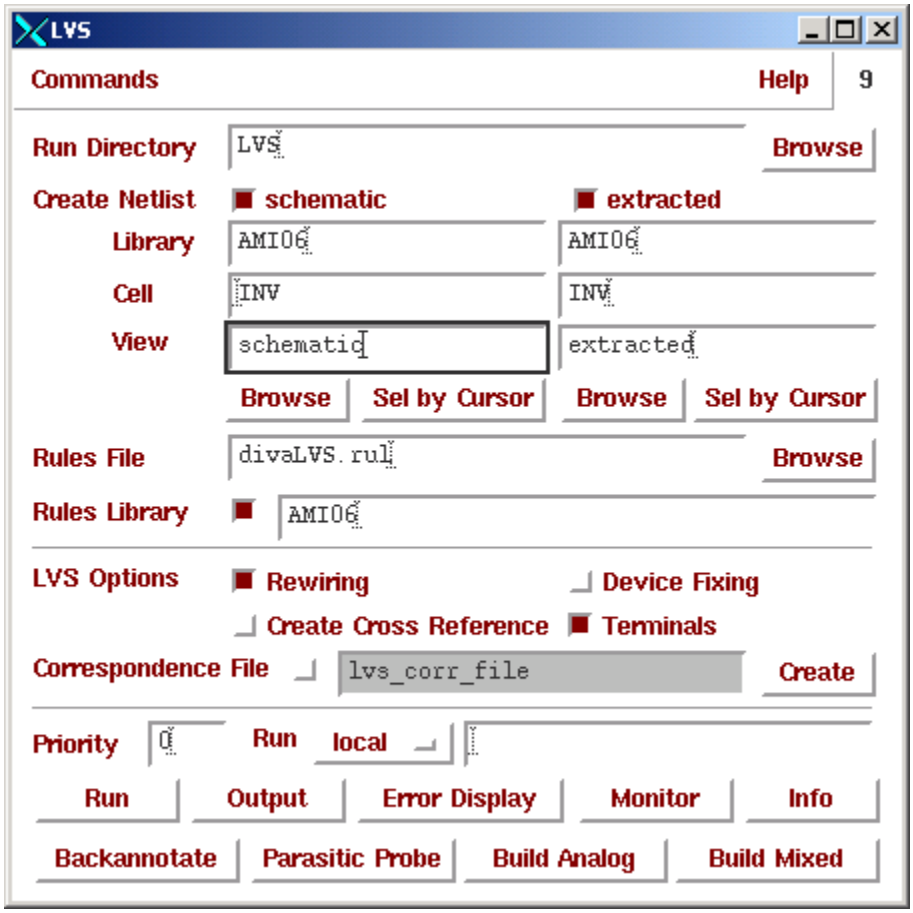

Figure 96: Running an LVS Check

Figure 97: Successful completion of an LVS check.

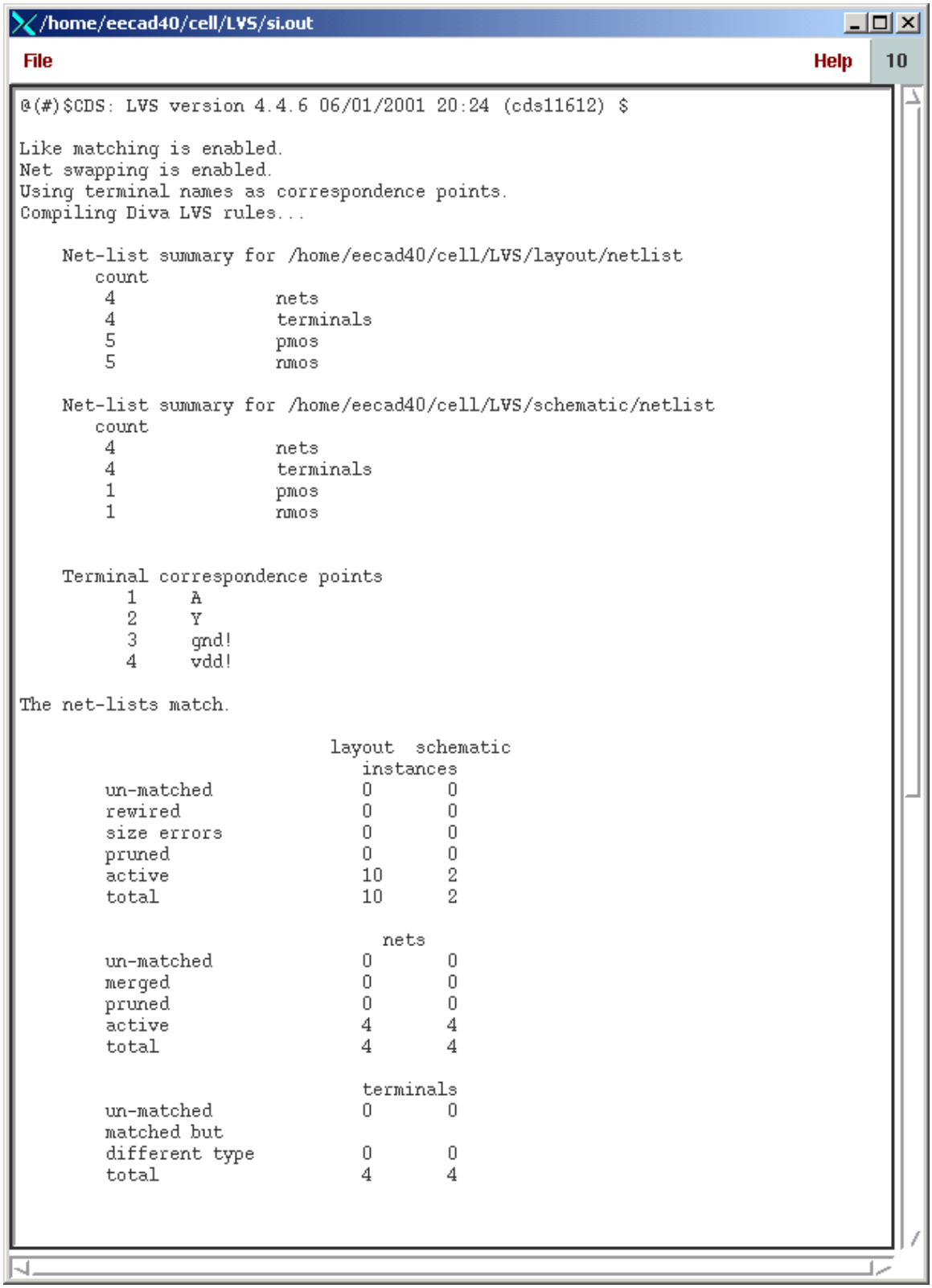

Figure 98: Output of LVS Check.

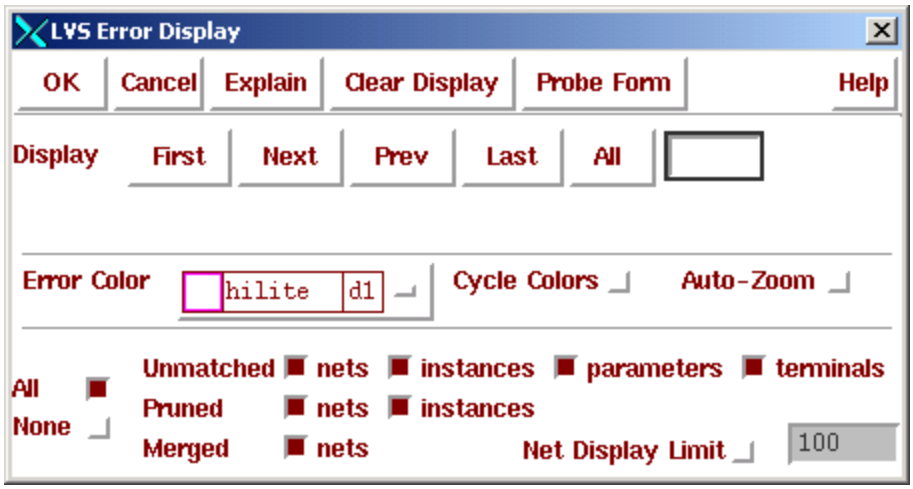

Figure 99: LVS Error Display.

Summary:

We have drawn a layout view of the circuit, and made sure that it is electrically equivalent to the schematic we simulated in chapter 3. All that remains is to re-run the simulation with he extracted view of the circuit to make sure we still meet specification.

Post Extraction Simulation

In post extraction simulation, you verify that your circuit still meets your specification with the additions of parasitic capacitances. For example, in a schematic two wires that are not connected there is no transfer of AC voltage or current. In a layout where two metal lines are close, but not connected there is a capacitance between the two that will cause cross talk. Post extraction simulation will show these errors.

To perform a post extraction simulation on your inverter:

1. Open up your INV_TB schematic Start the Affirma Analog environment.

2. Set up the simulation as before (Figure 100).

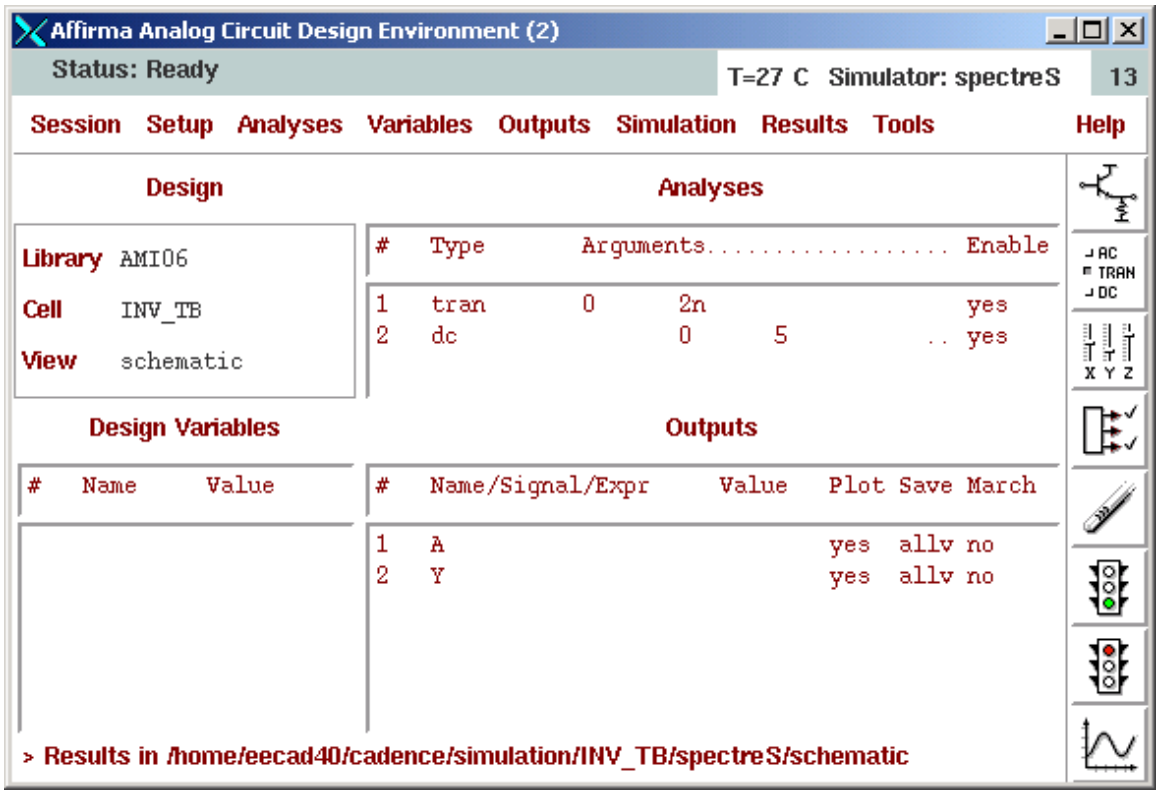

Figure 100: Setting for post extraction simulation.

Goto Setup... Environment. A pop-up like Figure 102should appear.

In the Switch View List insert the view extracted before the spectreS view like in Figure 101. Click ok.

Now run the simulation. Your results should match Figure 102.

Note: If you want to use the schematic view for simulation you have to delete the extracted view from the switch list.

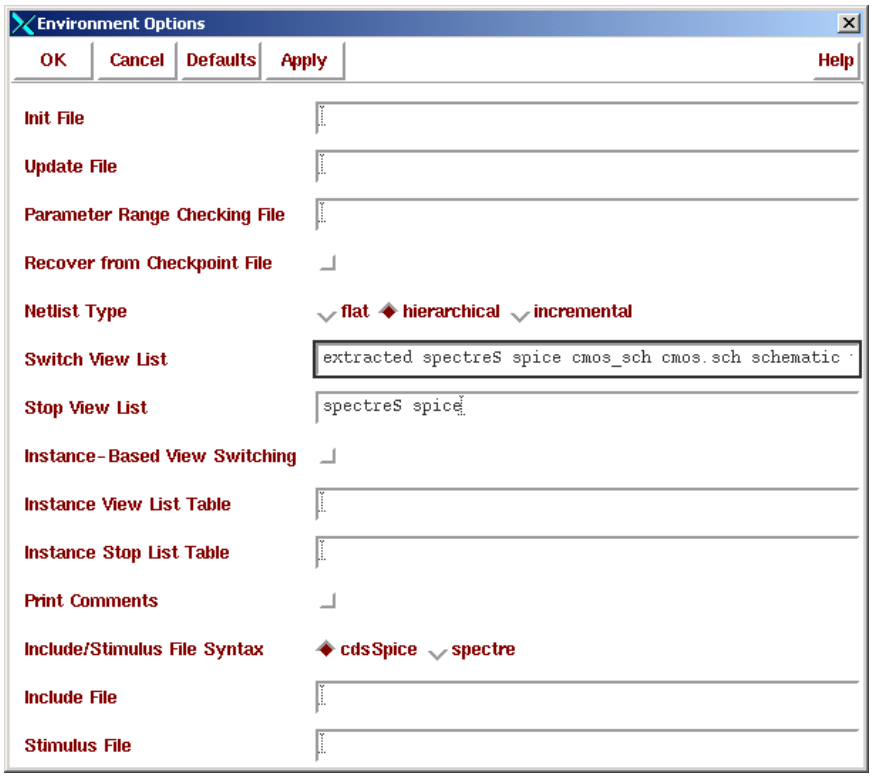

Figure 101: Inserting the extracted view.

We can see that the V_{INTH} values has shifted slightly lower to 2.37 Volts, but it is still within specification.

From Figure 103, we can see that we are within 1% of our timing requirements. This is more than sufficient considering that process parameters can vary as much as 10%.

Re-Run the simulation with the net P selected to check power. We can see in Figure 104 that the power used by this circuit is less than 20mW.

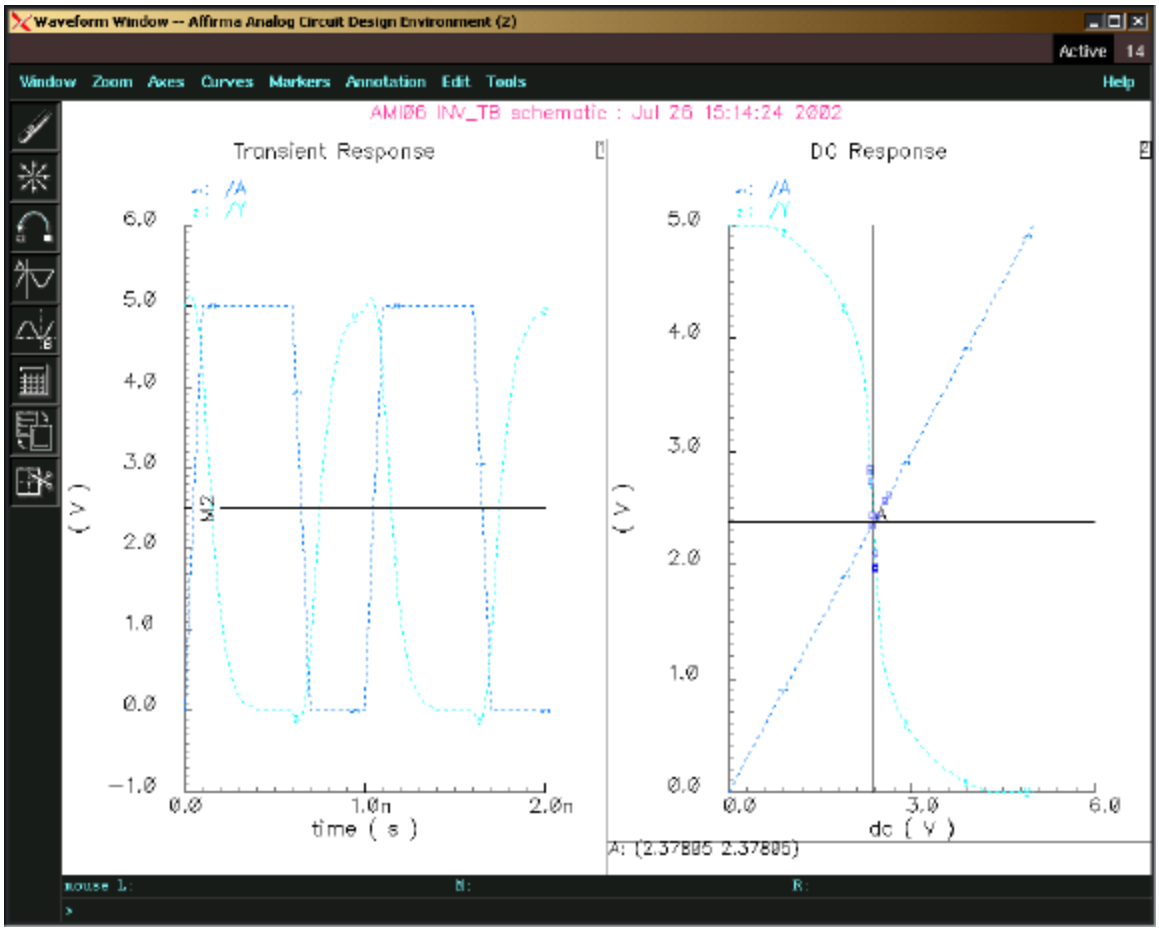

Figure 102: The extracted simulation.

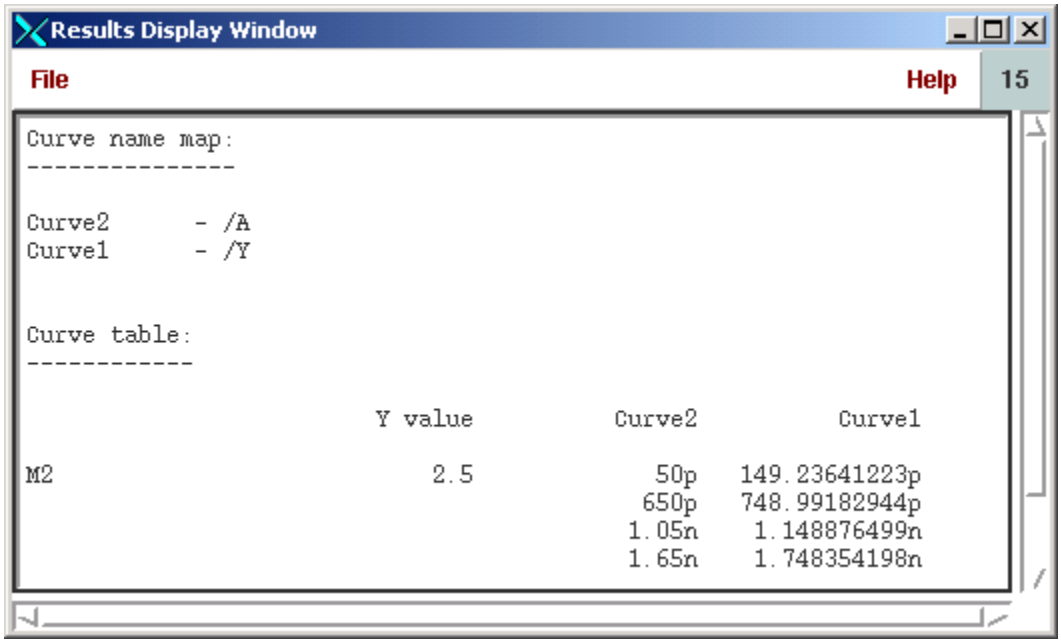

Figure 103: Propagation Delay from Extracted Circuit.

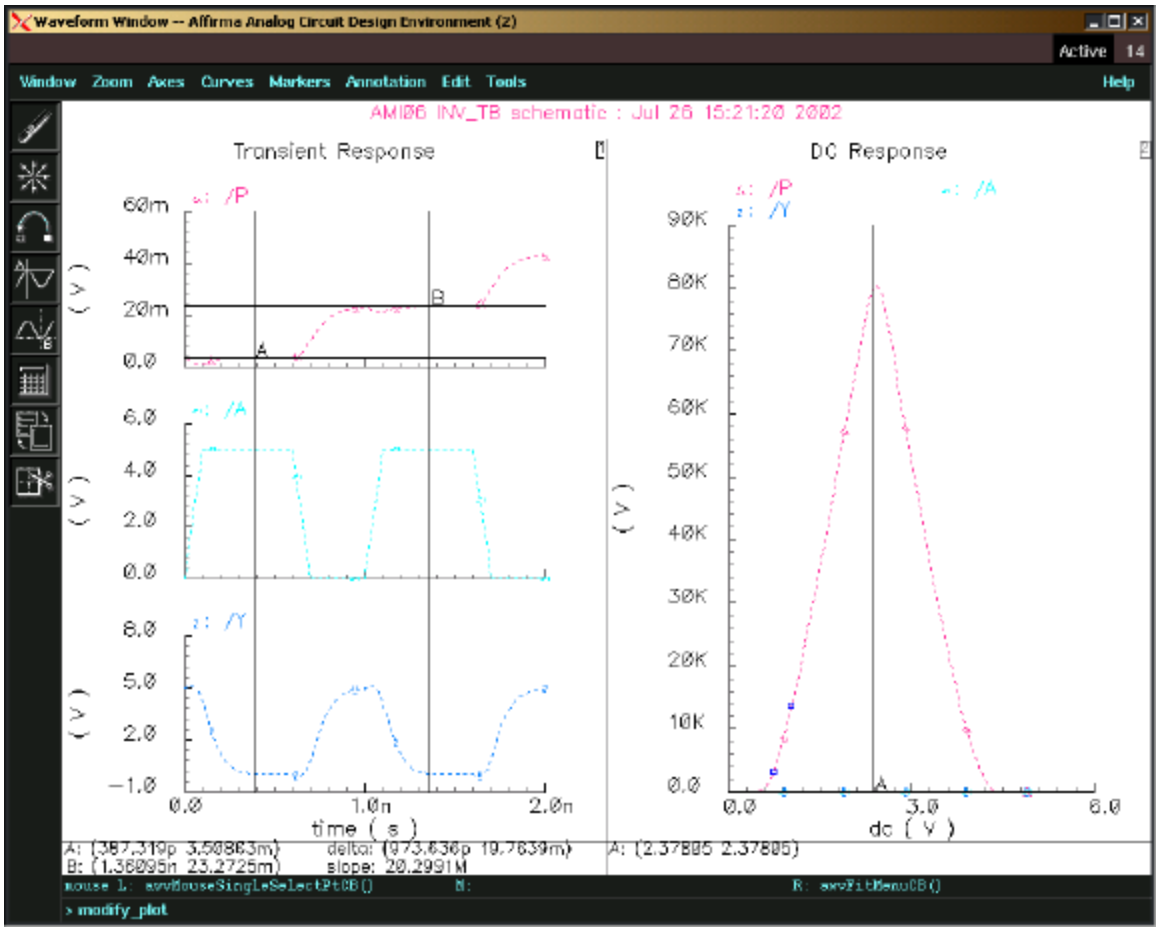

Figure 104: Power used by Extracted Circuit.

Summary:

The designed circuit met every specification! If the layout view did not meet specification we would have to resize the transistors in the layout view, run DRC, Run extraction, run LVS and this simulation all over again until we got it right.

- **The design flow we used is the most efficient way to meet our specification.**
- **If we did not use hand calculations, we would have been totally guessing about the lengths and widths of our transistors.**
- **If we did not use schematic capture and tried to draw the layout view we would have to redraw the circuit every time we wanted to change the size of a transistor.**
- **With out DRC we would have to see if we did not violate any design rules by eye!**
- **With out Extraction we would not know what our parasitic capacitances were.**

Appendix

A

Appendix A: Further Work and AC Analysis

Further Work:

Use the inverter test bench already created:

1. Change the load capacitor to 1000f F, and re-run the simulation, the propagation time should increase. (Press q in the schematic window and click on the capacitor)

2. Change the vpulse to a vsin with an Amplitude of 2.5 V, Frequency of 1GHz, DC offset voltage of 2.5 V, and a delay time of zero seconds. You should get a similar result as in Figure 105 . Questions: (Delete the DC Analysis or you will get a warning.)

a. How can we simulate a sine wave with a digital circuit?

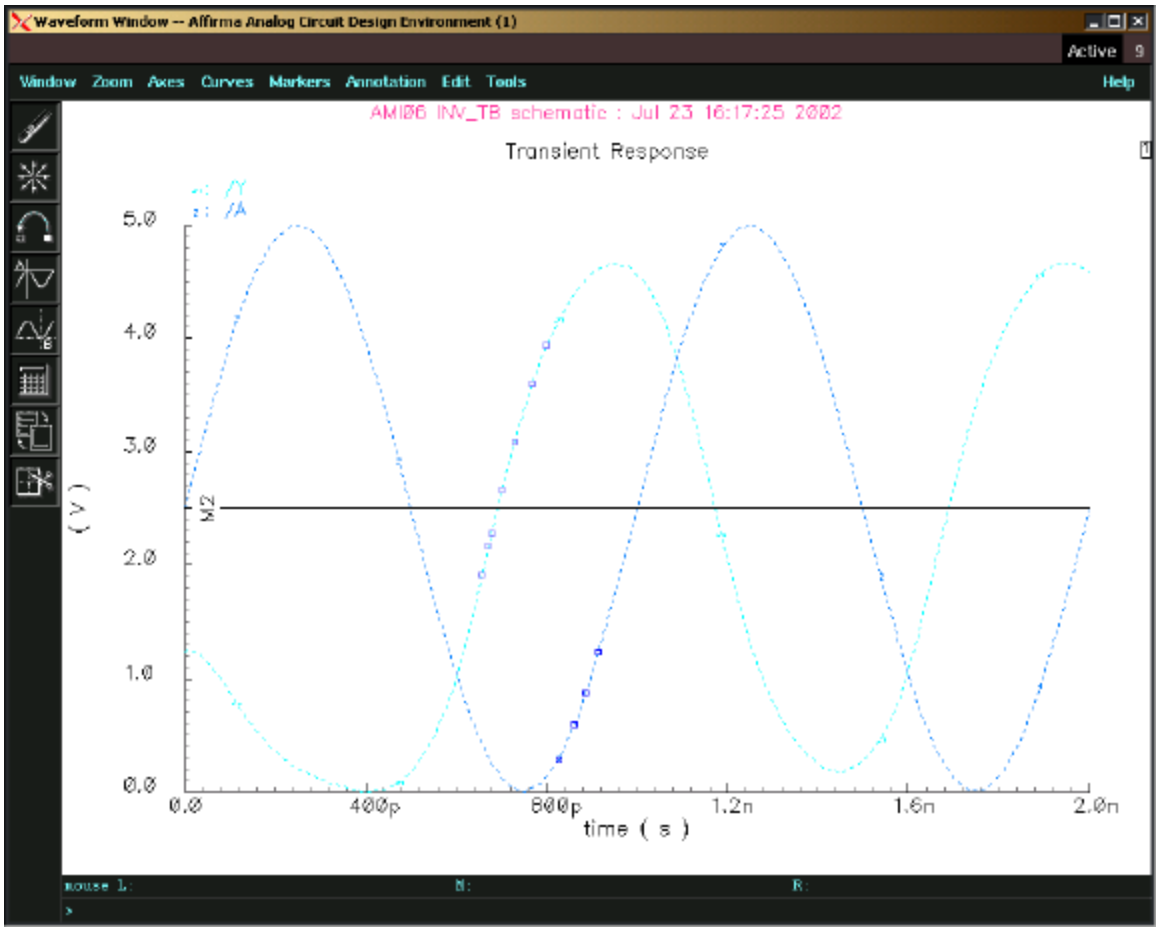

Figure 105: Output of Sine Wave Input

AC Analysis:

This portion does not have anything to do with meeting our specification, but it is useful to help you set up an AC analysis for other circuits. It also demonstrates that the spectre spice environment is an analog simulation environment that can be used for digital circuits.

Start the software and open up the testbench of the inverter created in the previous sections. Delete the vlpulse and add a vsin. Press q, click on the vsin and fill out the pop-up like Figure 106. Setting the AC magnitude to 1 will make it easier to read the gain off of the plot window.

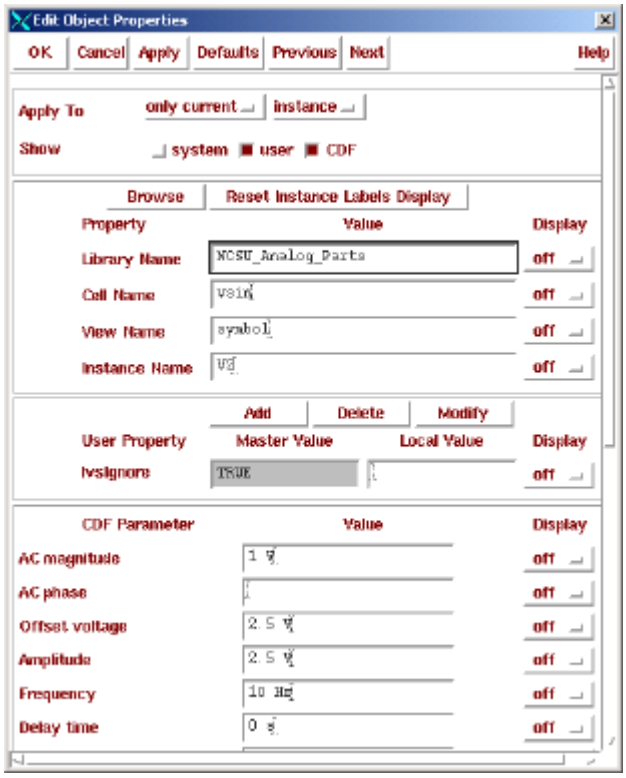

Figure 106: Setting up vsin for an AC Analysis.

To set up an AC Analysis, in the Affirma window go to Analysis... Choose, and click on ac and fill the pop-up like in Figure 107.

Run the simulation and the results window would look like Figure 108.

Question 1: Why don't the voltage gains from the transient and AC analysis match? You can read off the peak voltages with a crosshair marker in the transient response and you can add a vertical marker at 1 GHz and display the intercepted data to get the AC voltage gain.

Question 2: Re-Run the simulation with the amplitude set to 1 V instead of 2.5 V. The simulation results are closer, but why is there still some error?

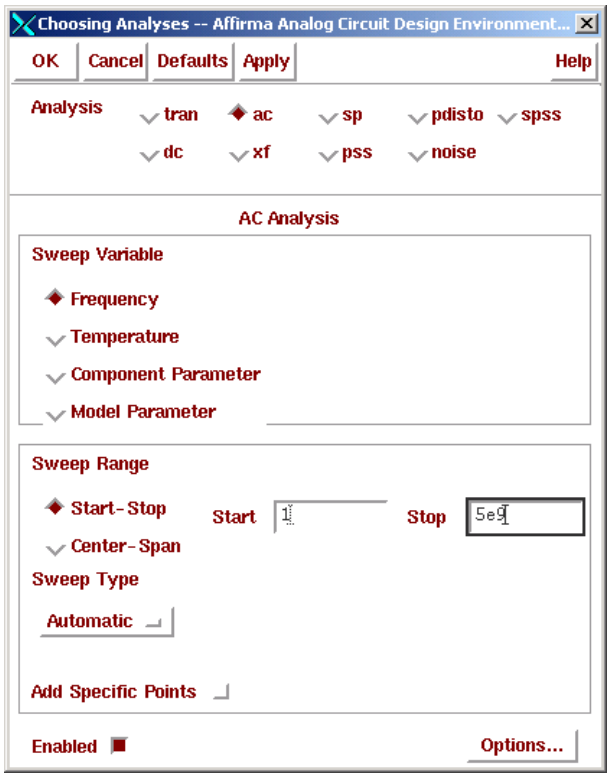

Figure 107: Setting up an AC Analysis from 1Hz to 5GHz.

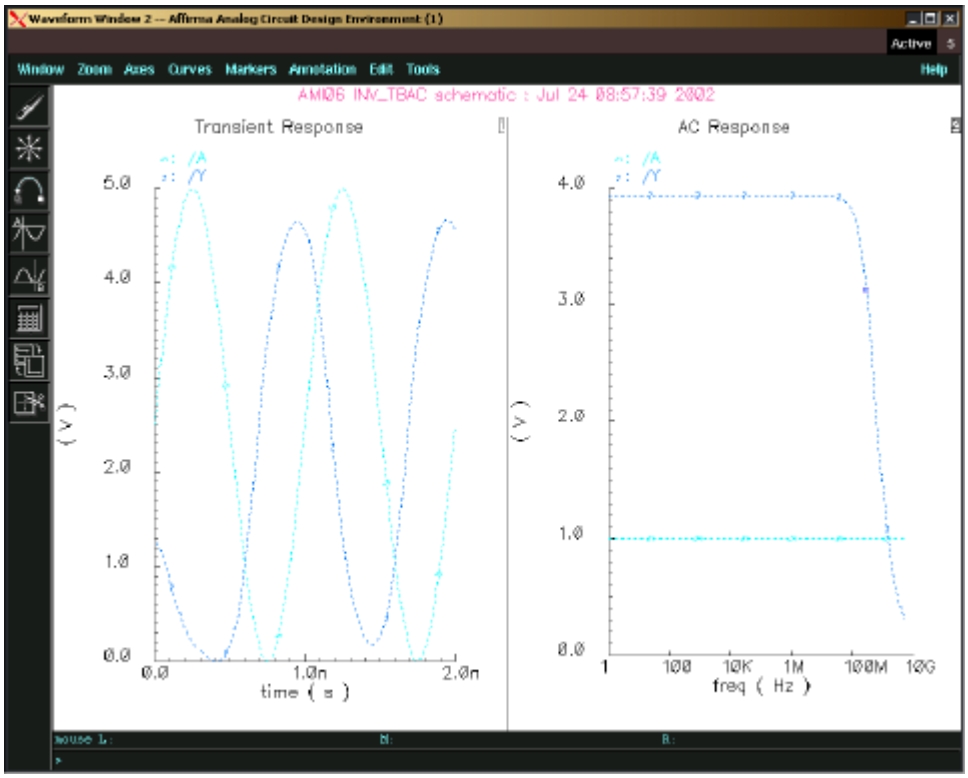

Figure 108: Results of a Transient an AC Analysis.