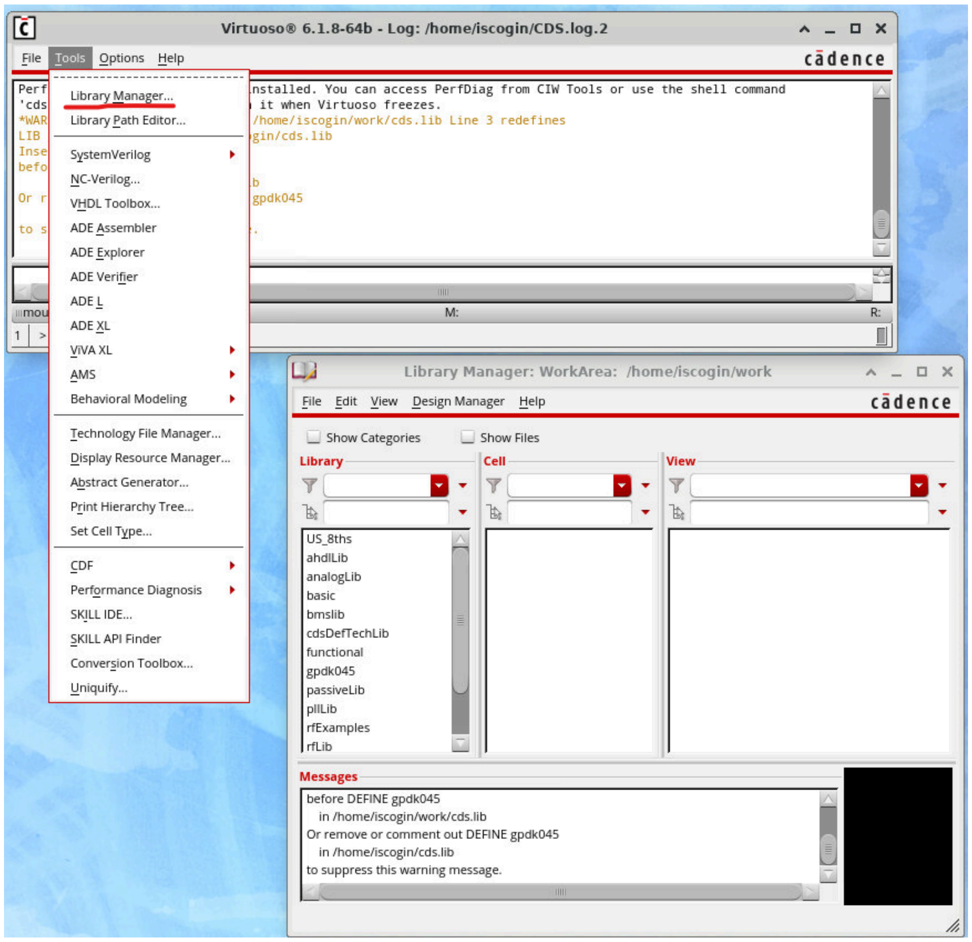
Virtuoso Schematics & Simulation

Introduction

This guide will walk the user through the creation and simulation of schematic-level designs in the Cadence Virtuoso suite. This is preceded by the ‘Virtuoso Palmetto Cluster Setup’ guide and followed by the ‘Virtuoso Layout’ guide. If you have not completed the library setup procedures from the previous guide, please take the opportunity to do so now.

Creating a Library

With initial library setup from the previous guide complete, it is time to start a personal design library. From the CIW, open the Library Manager from Tools > Library Manager…

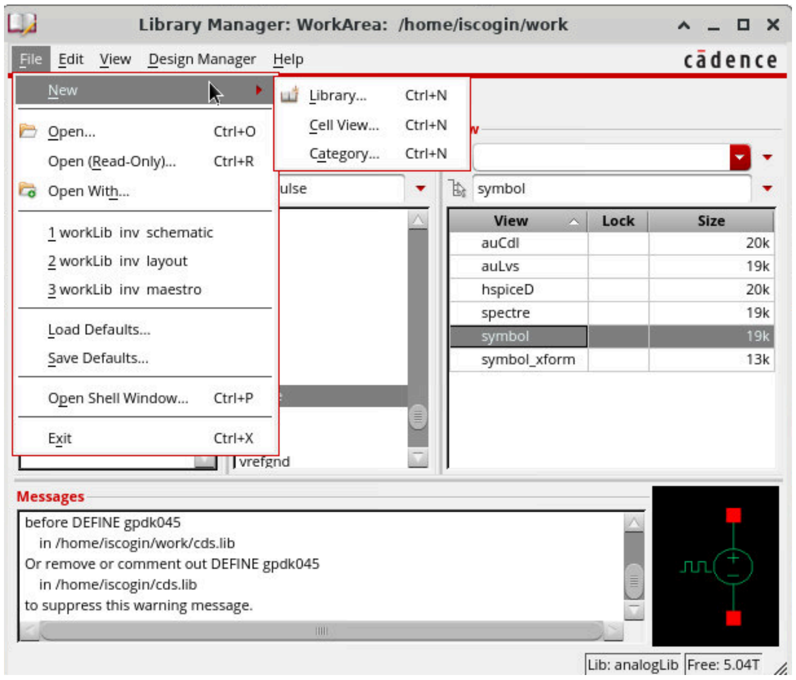


```
Perf    nstalled. You can access PerfDiag from CIW Tools or use the shell command
'cds    it when Virtuoso freezes.
*WAR    /home/iscogin/work/cds.lib Line 3 redefines
LIB     gin/cds.lib
Inse
befo    b
Or r    gpdk045
to s    .
```

```
before DEFINE gpdk045
   in /home/iscogin/work/cds.lib
Or remove or comment out DEFINE gpdk045
   in /home/iscogin/cds.lib
to suppress this warning message.
```

The Library Manager is the main access point for all of your design information. Each library is populated with cells, and each cell has its own views. For example, the layout view for a standard NMOS transistor is found at the L/C/V path gpdk045/nmos1v/layout, while the symbol view for a pulsing voltage source is found at analogLib/vpulse/symbol. As long as there is a direct hierarchy, multiple cell views from any library can be used in a single design.

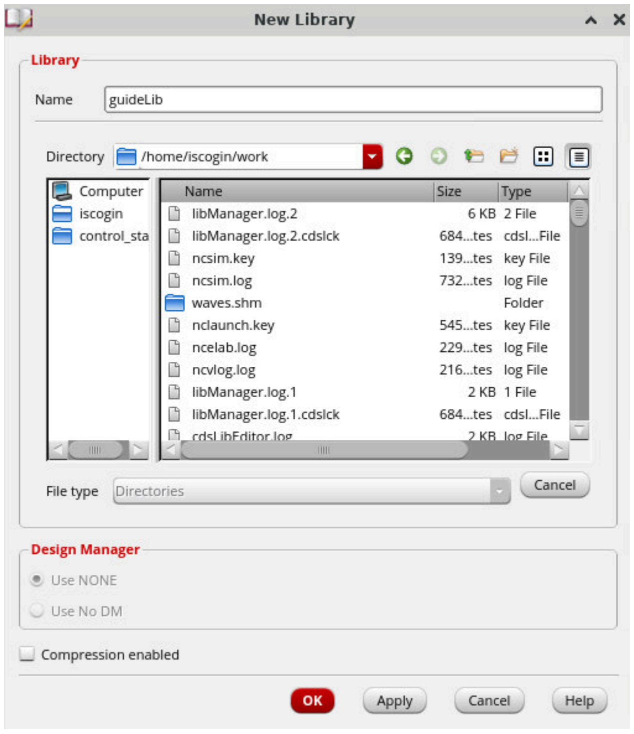
To create a new library, choose File > New > Library… from within the Library Manager. Make sure that the directory is set to your desired working directory and choose a name for the library (I will be using guideLib). Press OK, and when asked about the technology file choose to “Attach to an existing technology library”.



View	Lock	Size
auCdl		20k
auLvs		19k
hspiceD		20k
spectre		19k
symbol		19k
symbol_xform		13k

```
before DEFINE gpdk045
   in /home/iscogin/work/cds.lib
Or remove or comment out DEFINE gpdk045
   in /home/iscogin/cds.lib
to suppress this warning message.
```

Lib: analogLib Free: 5.04T



Name: guideLib

Directory: /home/iscogin/work

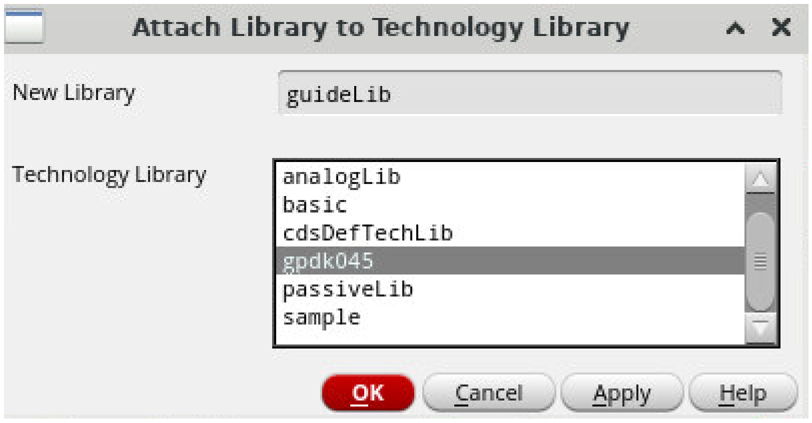
Name	Size	Type
libManager.log.2	6 KB	2 File
libManager.log.2.cdslck	684…tes	cdsl…File
ncsim.key	139…tes	key File
ncsim.log	732…tes	log File
waves.shm		Folder
nclaunch.key	545…tes	key File
ncelab.log	229…tes	log File
ncvlog.log	216…tes	log File
libManager.log.1	2 KB	1 File
libManager.log.1.cdslck	684…tes	cdsl…File
cdsLibEditor.log	2 KB	log File

File type: Directories

[✓ Use NONE] Use No DM

Compression enabled

When given the list of Technology Libraries, choose to bind “gpdk045” to the new library and press OK. This will set the new library to use the technology data (design rules, physical layers, etc..) given by the gpdk045 technology. Mixing up tech files later down the line will cause problems, so it is important to have this set up correctly. You should now see your library in the appropriate column of the Library Manager.



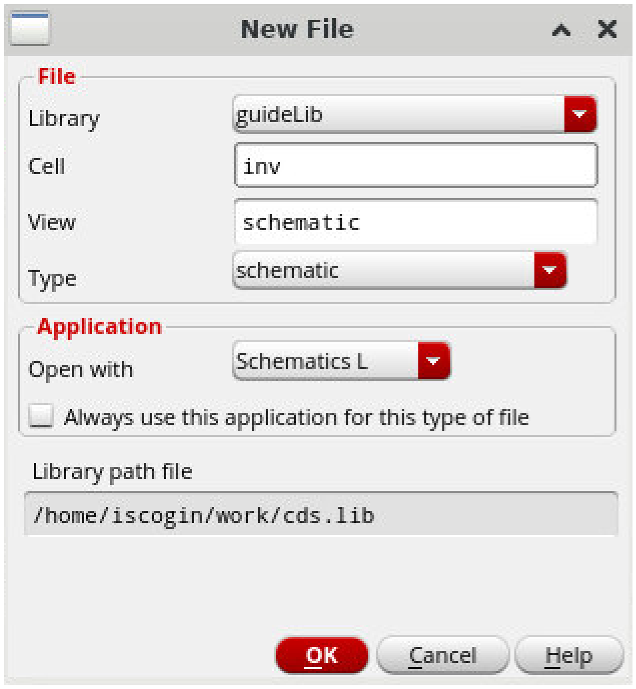
New Library: guideLib

Technology Library:

```
analogLib
basic
cdsDefTechLib
gpdk045
passiveLib
sample
```

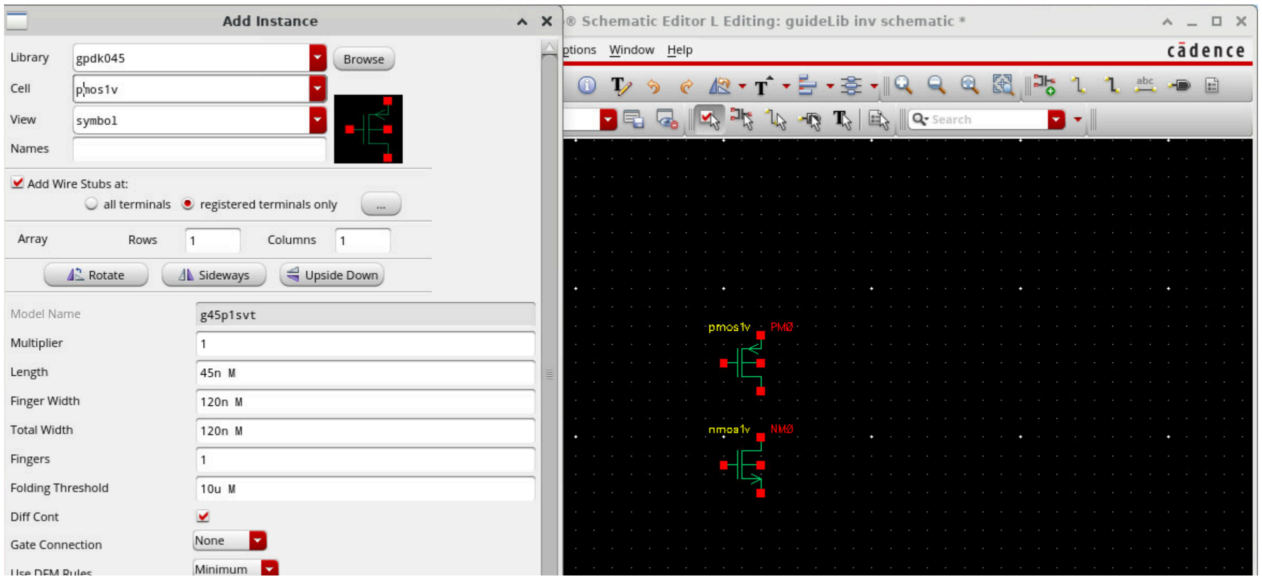
Basic Schematic & Symbol Editing

This section of the guide will walk through the creation of two schematics from the ground up: a CMOS inverter and a simple ring oscillator. Starting with the inverter, create a new cellview by choosing File > New > Cell View… from the Library Manager. Set the Library to the library you created, View and Type to “schematic”, and name the cell “inv”. Pressing OK will create the relevant cells and views in the Library Manager and open the Schematic Editor.



Library: guideLib
Cell: inv
View: schematic
Type: schematic
Open with: Schematics L
☐ Always use this application for this type of file
Library path file: /home/iscogin/work/cds.lib

The inverter design will have two components: an NMOS and a PMOS. To place the NMOS transistor symbol into the schematic, press “i” to open the Add Instance. Choose “gpdk045” as the Library, “nmos1v” as the Cell, and “symbol”, and then click in the schematic window to place the transistor at your cursor location. Do the same for the PMOS transistor, except choose the Cell as “pmos1v” and place it above the NMOS transistor. Note that at any time, you can press the “f” key to perform a zoom fit.


Add Instance

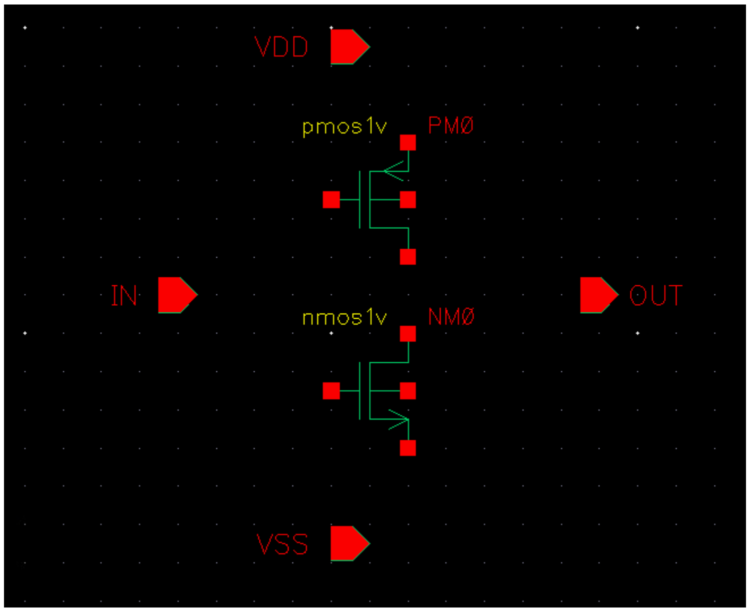
Library	gpdk045
Cell	pmos1v
View	symbol
Names	

[✓] Add Wire Stubs at: all terminals [● registered terminals only]

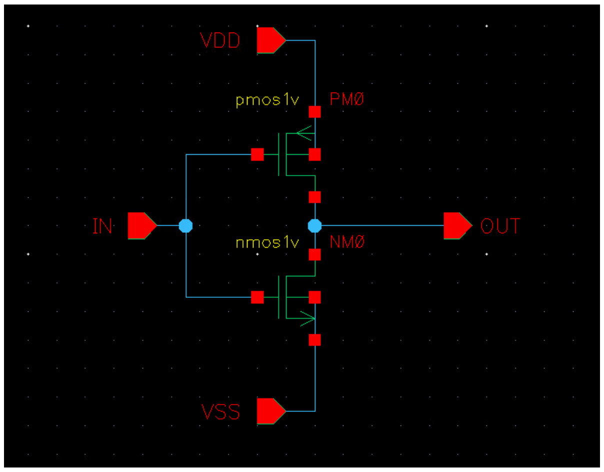
Array Rows 1 Columns 1

Model Name	g45p1svt
Multiplier	1
Length	45n M
Finger Width	120n M
Total Width	120n M
Fingers	1
Folding Threshold	10u M
Diff Cont	✓
Gate Connection	None
Use DFM Rules	Minimum

The inverter schematic will include three input pins (IN, VDD, VSS) and one output pin (OUT). Press P to open the Create Pin menu, and create the four pins. Make sure that the direction pin is set correctly for each pin. It is also good practice to match the signal type (power, ground, signal) to the corresponding pin, though this should be done automatically.

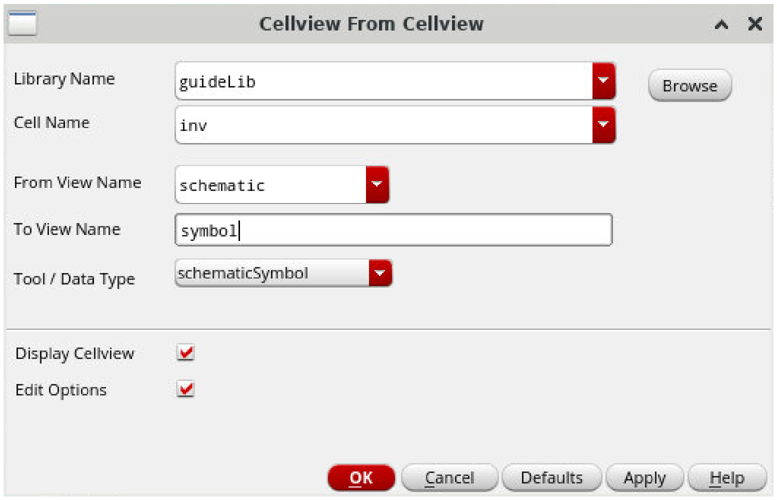


Press “w” to prepare the wiring tool, then connect the IN pin to the gates of each transistor, VDD to the PMOS source, VSS to the NMOS source, and OUT to the drains of each transistor. Note that you can use the “s” key to quick wire between points denoted by yellow diamonds, and the “u” key to undo the previous action, and “Shift+u” to redo. A more comprehensive list of helpful commands can be found here.



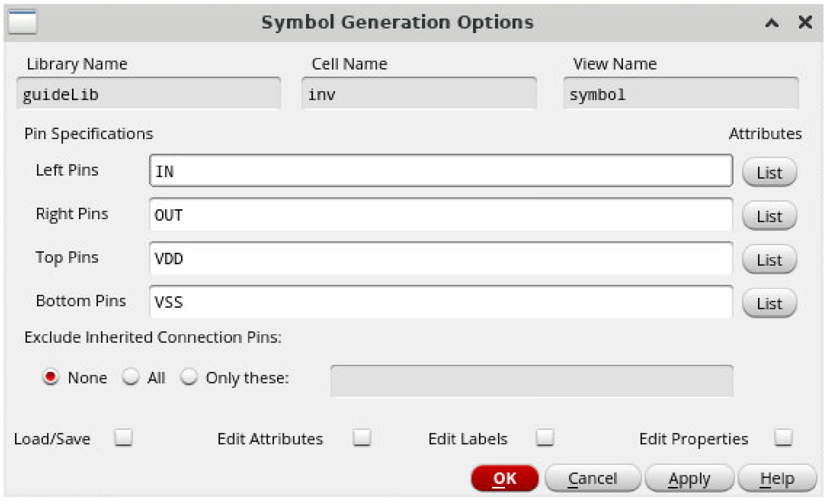
Select the PMOS transistor and press “q” to open the Edit Object Properties menu. Since NMOS and PMOS transistors are not balanced in terms of carrier mobility (with NMOS transistors typically being significantly stronger than PMOS), it is important to match the relative strength of complementary transistors to ensure even rising/falling delays. Change the Finger Width of PM0 to “140n” and press OK for the changes to be applied. To verify the design, choose File > Check and Save or click the save icon with the green check mark. Fix any errors highlighted in the schematic and repeat until all warnings are resolved.

To create a symbol, choose Create > Cellview > From Cellview… from within the Schematic Editor. Make sure the options are the same as those listed below and choose OK. This will use the schematic view you created as the base for a symbol view.



Library Name	guideLib
Cell Name	inv
From View Name	schematic
To View Name	symbol
Tool / Data Type	schematicSymbol
Display Cellview	✓
Edit Options	✓

In the Symbol Generation Options, set IN as the left pin, OUT as the right pin, VDD as the top pin, and VSS as the bottom pin. Leave all other options as their default values and press OK. This will open the Symbol Editor window.


Symbol Generation Options

Library Name	Cell Name	View Name
guideLib	inv	symbol

Pin Specifications

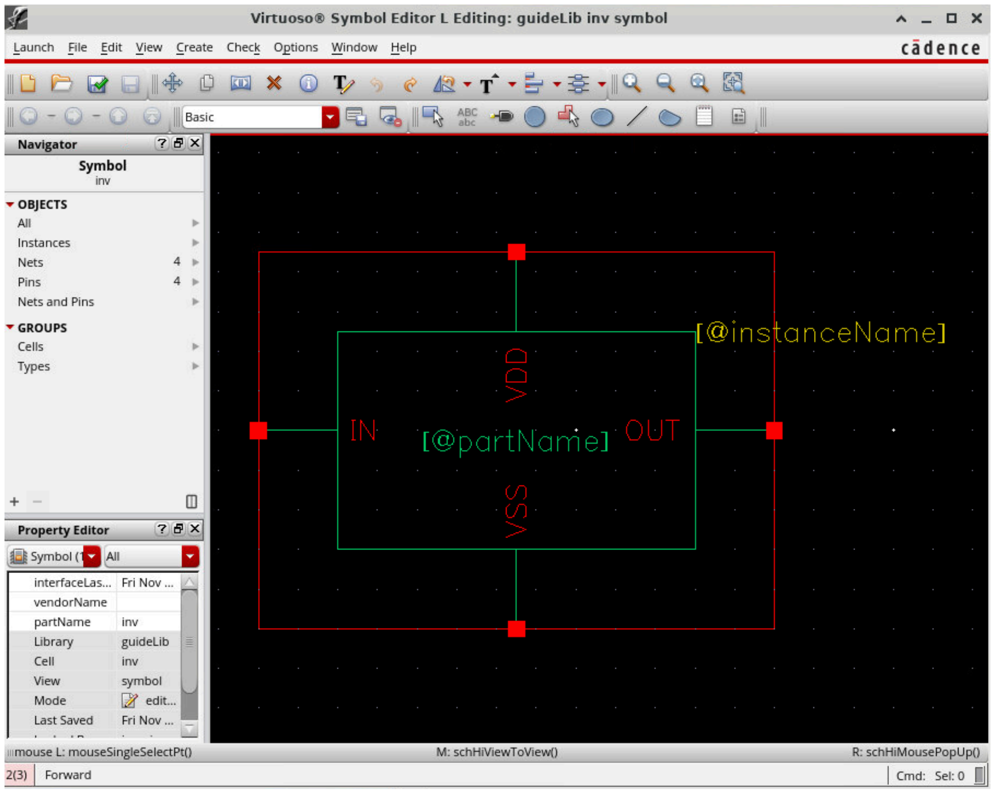
		Attributes
Left Pins	IN	List
Right Pins	OUT	List
Top Pins	VDD	List
Bottom Pins	VSS	List

Exclude Inherited Connection Pins:

[✓ None] All Only these:

Load/Save ☐ Edit Attributes ☐ Edit Labels ☐ Edit Properties ☐

OK Cancel Apply Help

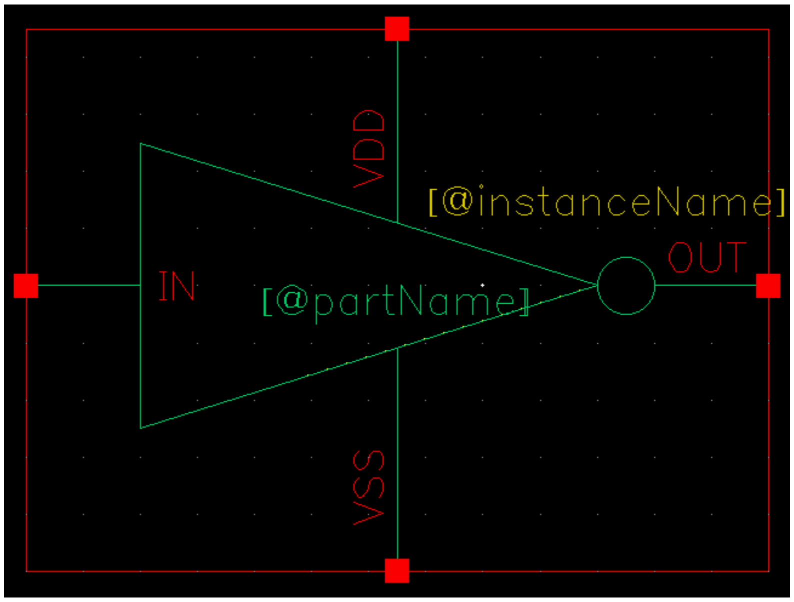

Virtuoso® Symbol Editor L Editing: guideLib inv symbol

interfaceLas...	Fri Nov ...
vendorName	
partName	inv
Library	guideLib
Cell	inv
View	symbol
Mode	edit...
Last Saved	Fri Nov ...

mouse L: mouseSingleSelectPt() M: schHiViewToView() R: schHiMousePopUp()

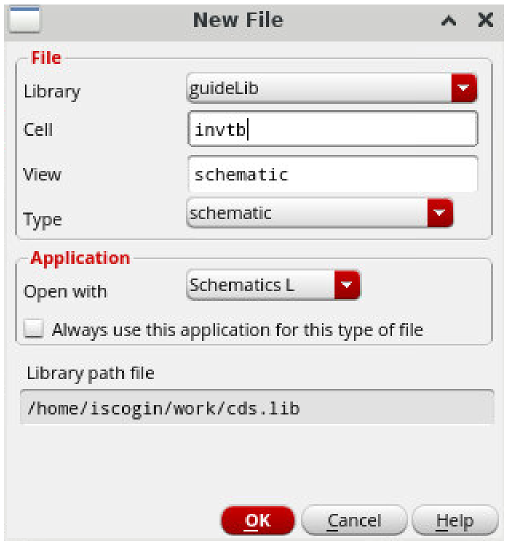
2(3) Forward Cmd: Sel: 0

Press the delete key to activate the deletion tool and click the rectangle to remove it from the symbol. Use the line and circle tools to create the symbol for an inverter. Pressing “esc” will cancel line placement, while double-clicking will terminate the line at the desired position. Note that you can increase or decrease the grid precision by choosing Options > Display… (or pressing “o”) and changing the Snap Spacing option under Grid Controls from its default value of 0.0625. Use the Check and Save function to finalize the symbol, then close the Symbol Editor and Schematic Editor windows. You should see two views (“schematic” and “symbol”) in the “inv” cell.



Testbench Creation

After completing the schematic for a design, simulation requires the creation of a testbench and a simulation view called a “maestro” view. To create an inverter testbench, make another new schematic cell from File > New > Cell View… and choose the title “invtb”.


New File

File

Library: guideLib

Cell: invtb

View: schematic

Type: schematic

Application

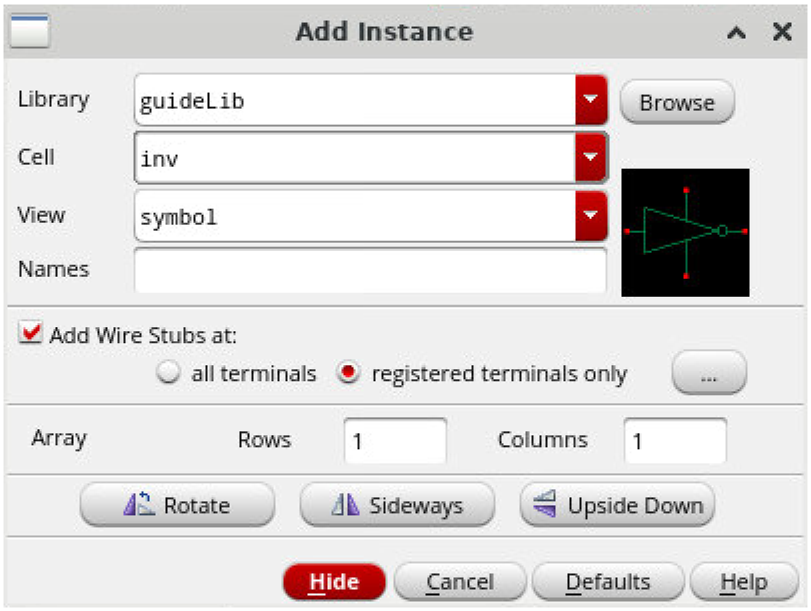
Open with: Schematics L

Always use this application for this type of file

Library path file

/home/iscogin/work/cds.lib

Press “i” in the schematic window to open the Add Instance menu, then choose the symbol view of the “inv” cell from “guideLib” and add it to your schematic.


Add Instance

Library: guideLib

Cell: inv

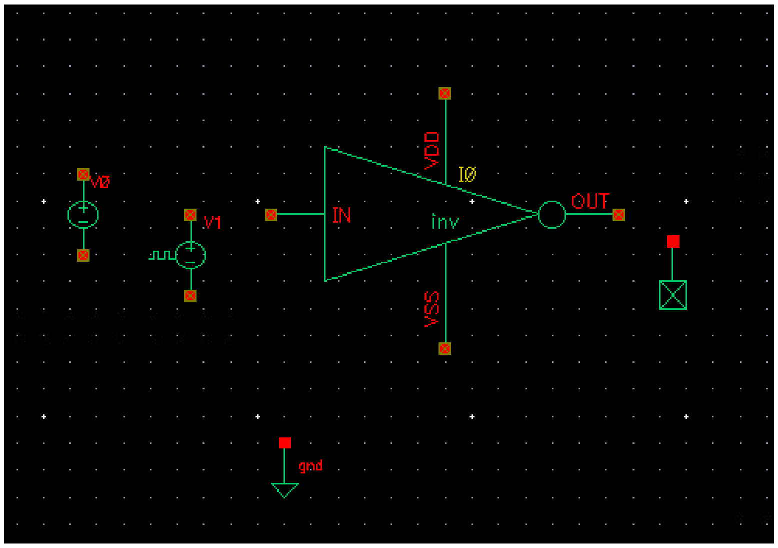
View: symbol

Names:

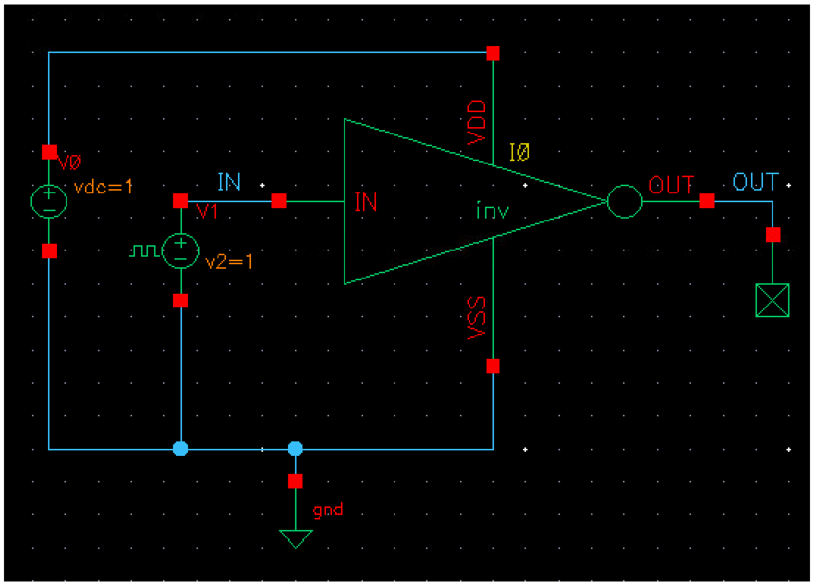
Add Wire Stubs at: [✓]
○ all terminals [● registered terminals only]

Array Rows 1 Columns 1

From the analogLib library, add a DC voltage source (vdc), a pulsing voltage source (vpulse), and a ground symbol (gnd). Also add a noConn symbol from the “basic” library. The DC source will act as the rail voltage VDD for the inverter, while the pulsing source will be the test input. Since the output will not be connected to anything, it is good practice to attach a noConn symbol. A gnd symbol should also be included in any design to properly denote a reference node.

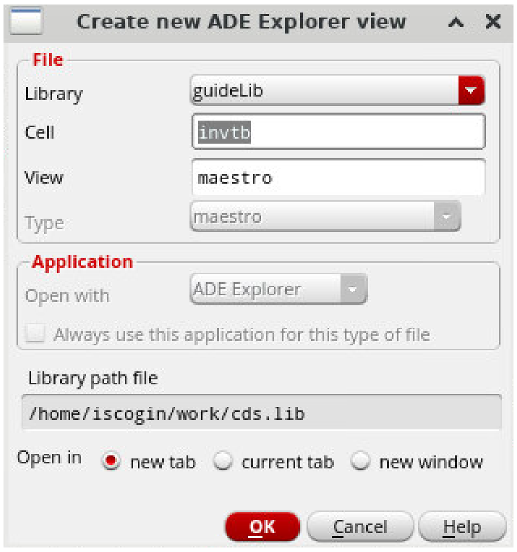


Wire each component with its respective connections (pictured below) and then Check & Save the design to ensure that there are no connectivity errors. Assign net labels “IN” and “OUT” to the input and output wires by using the “L” hotkey and clicking the wires. Press “q” to use the Object Properties Menu to change the “DC voltage” parameter of the DC source to 1, the “Voltage 2” parameter of the pulse source to 1, and the “Period” parameter of the pulse source to 100n. If there are no errors after running a Check & Save, then the testbench is complete.



Virtuoso ADE

From the testbench schematic window, choose Launch > ADE Explorer to open the Virtuoso Analog Design Environment. ADE Explorer and ADE Assembler are the primary simulation environments one might use, the primary difference being that Explorer is a single-simulation environment while Assembler is used to coordinate multiple simulations. When prompted, choose to create a new view and ensure that the default fields match those shown below. Pressing OK will create a maestro view for simulation management.



Create new ADE Explorer view

File

Library	guideLib
Cell	invtb
View	maestro
Type	maestro

Application

Open with: ADE Explorer

Always use this application for this type of file

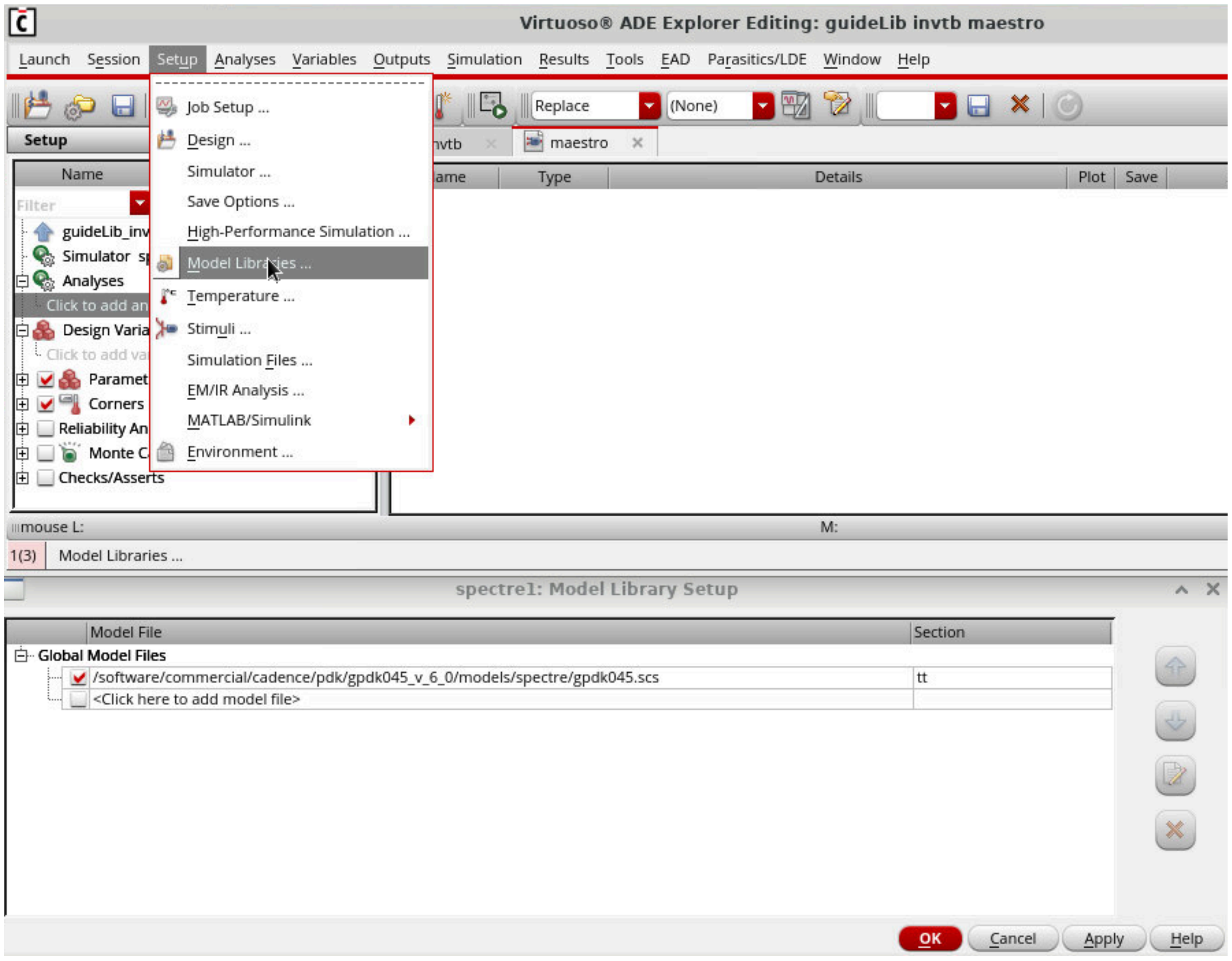
Library path file

/home/iscogin/work/cds.lib

Open in [✓ new tab] current tab new window

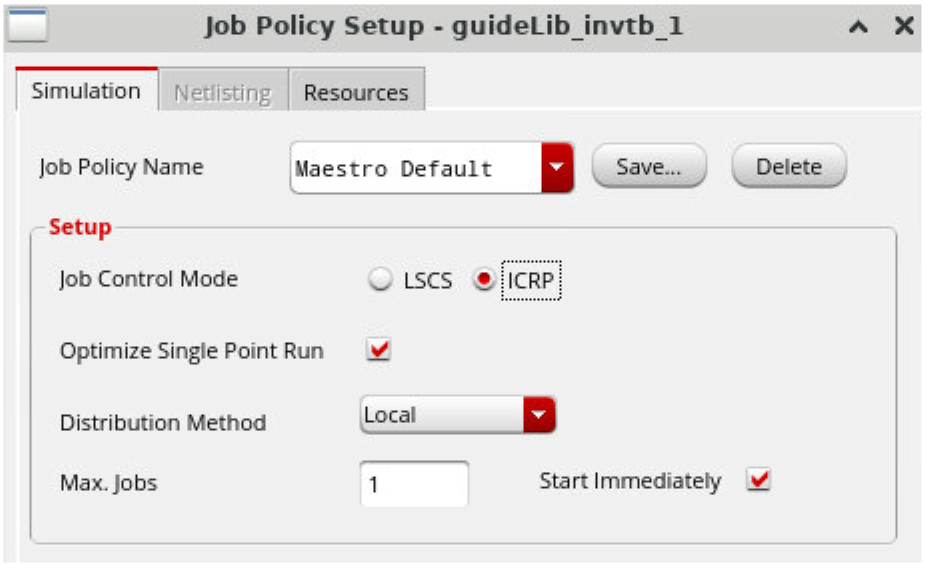
Before you start defining simulations, there are a few setup options that need to be configured. First, go to Setup > Model Libraries… and ensure that the gpdk045 model library is included and enabled in the Global Model Files (the path to which is included below). For general use, the section should be set to “tt” to reflect standard process/temperature operating conditions. This tends to be set to “mc” by default, but the “mc” section is intended for Monte Carlo simulations which fall outside of the scope of this guide. Once these are corrected, click OK.

/software/commercial/cadence/pdk/gpdk045_v_6_0/models/spectre/gpdk045.scs



Model File	Section
Global Model Files	
✓ /software/commercial/cadence/pdk/gpdk045_v_6_0/models/spectre/gpdk045.scs	tt
<Click here to add model file>	

The next step is to correct the Job Policy Setup. By default, Virtuoso ADE will attempt to use the LSCS Job Control Mode to run Spectre simulations. This has caused inconsistent license compatibility issues in the past, so we strongly recommend changing to the ICRP Control Mode. To do this, open the Job Policy menu through Setup > Job Setup… and change the Job Control Mode to ICRP. Leave everything else as its default value and press OK.


Job Policy Setup - guideLib_invtb_1

Simulation | Netlisting | Resources

Job Policy Name: Maestro Default

Setup

Job Control Mode: LSCS [✓ ICRP]

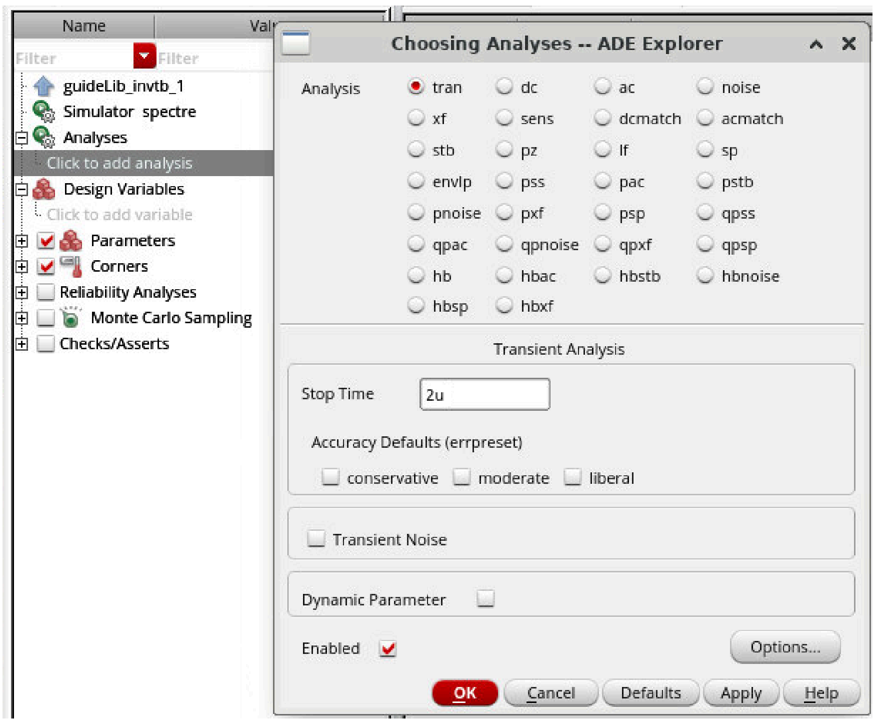
Optimize Single Point Run: ✓

Distribution Method: Local

Max. Jobs: 1

Start Immediately: ✓

With the Model Libraries and the Job Policy corrected, it is time to define the simulation analyses and outputs. In the Setup tab on the left side of the window, click the “Click to add analysis” text under the Analyses tab. The inverter input is defined in the testbench, and we are interested in seeing how the output responds over time. Choose a transient (tran) analysis, set the Stop Time for 2u (2 microseconds), and click OK. This will set the simulation to run from t=0 to t=2μs, with step time dynamically set by the simulator.


guideLib_invtb_1
Simulator spectre
Analyses
Click to add analysis
Design Variables
Click to add variable
Parameters
Corners
Reliability Analyses
Monte Carlo Sampling
Checks/Asserts

Choosing Analyses -- ADE Explorer

Analysis: [✓ tran] dc ac noise xf sens dcmatch acmatch stb pz lf sp envlp pss pac pstb pnoise pxf psp qpss qpac qpnoise qpxf qpsp hb hbac hbstb hbnoise hbsp hbxf

Transient Analysis

Stop Time: 2u

Accuracy Defaults (errpreset): conservative moderate liberal

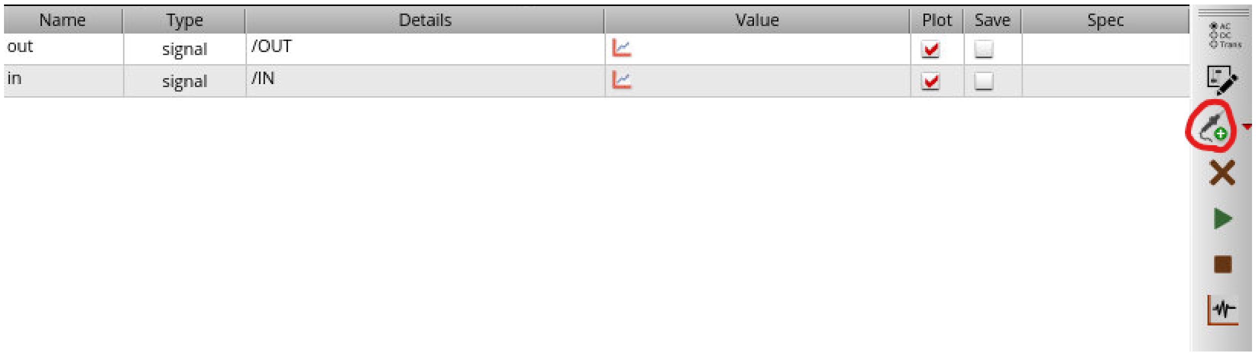
Transient Noise

Dynamic Parameter

Enabled: ✓

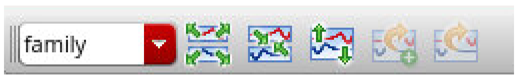
Options... OK Cancel Defaults Apply Help

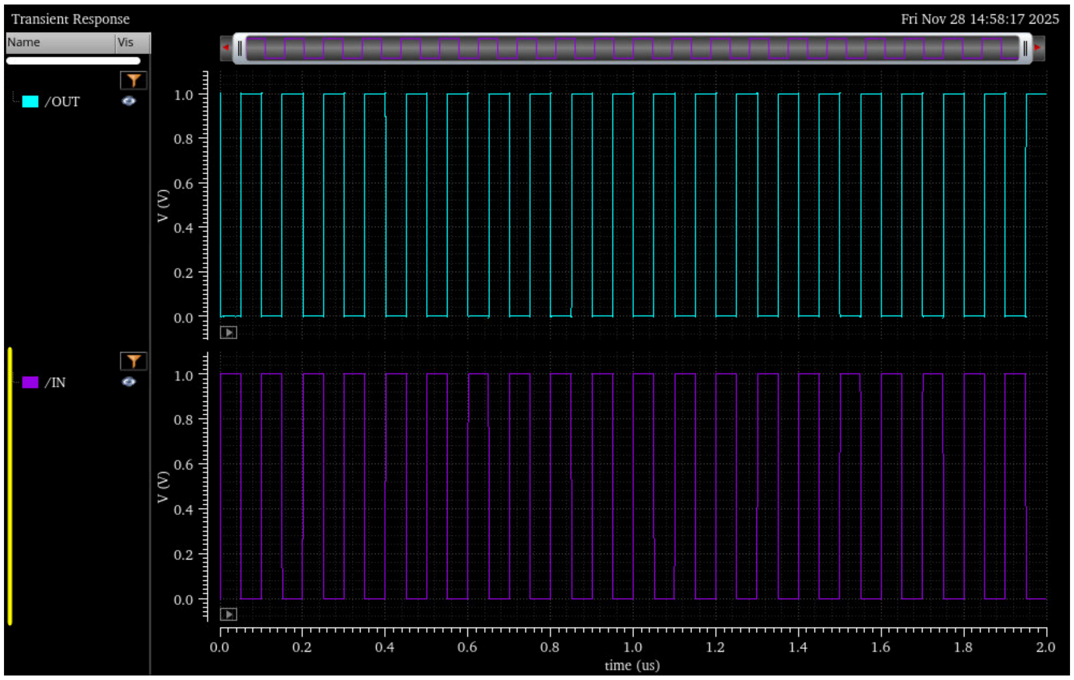
With the simulation defined, the final step before running is to choose the simulation outputs. Click the probe icon on the right side of the window to add an output. Set the name to “out” and the type to “signal”. Double-click the empty “Details” box and click the button with three dots to select a signal directly from the schematic. In this case, click the output wire of the inverter. Repeat the process for the input wire and enable plotting for both outputs.



Name	Type	Details	Value	Plot	Save	Spec
out	signal	/OUT		✓		
in	signal	/IN		✓		

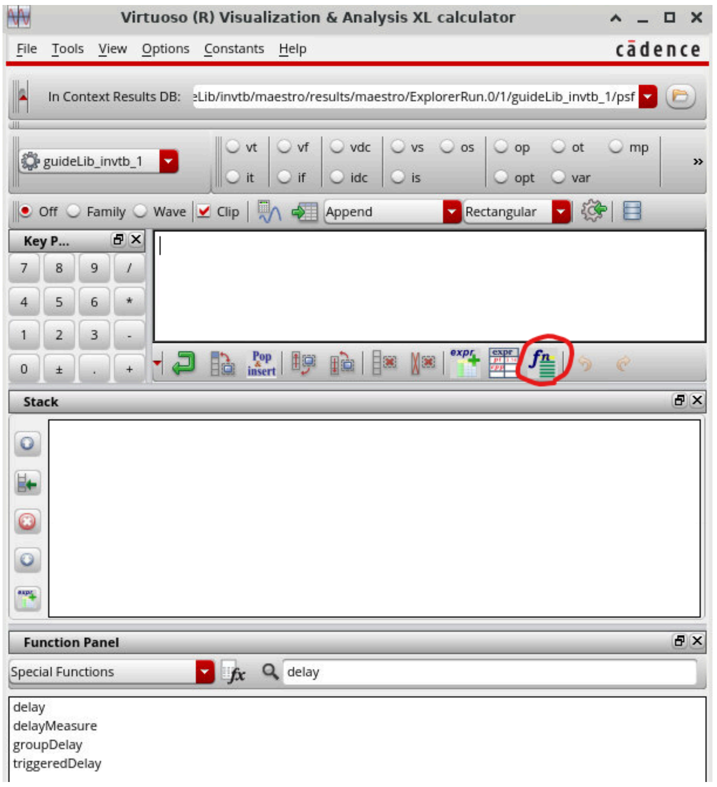
Clicking the green arrow below the probe icon will run the simulation. A simplified ViVA (Virtuoso Visualization & Analysis) window will appear docked in the ADE window, but a more comprehensive analysis window can be found in Tools > Results Browser… from the top bar. Note that the plots can be split or joined using the icons shown below.





Metrics

A key metric for the inverter in the design of a ring oscillator is the rising and falling propagation delay. In general terms, this is a measure of the delay between the input and output response (not to be confused with rise/fall times). From the ADE window, choose Tools > Calculator… to open the ViVA XL calculator. If it is not already selected, click the “Show Special Functions” button (circled) to open the list of special functions, and find the “delay” function.



In Context Results DB: ≥Lib/invtb/maestro/results/maestro/ExplorerRun.0/1/guideLib_invtb_1/psf

guideLib_invtb_1

Append

Rectangular

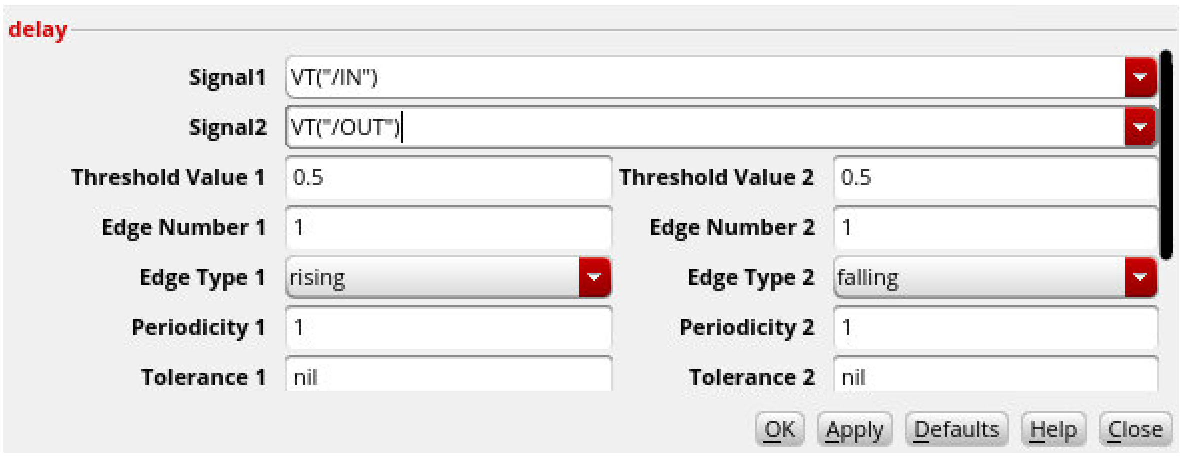
Special Functions

delay

Since we want to find the delay of the output as the input changes, enter VT(“/IN”) for Signal1 and VT(“/OUT”) for Signal2. VDD is 1V, so set the switching threshold values for both signals to 0.5. For the falling output propagation delay, set Edge Type 1 to rising and Edge Type 2 to falling. These will be reversed for the rising output delay. Set Edge Number to 1 for both, as we will use the first rising/falling edge of the input to measure the delay of the first falling/rising edge of the output. Everything else can be left as-is.

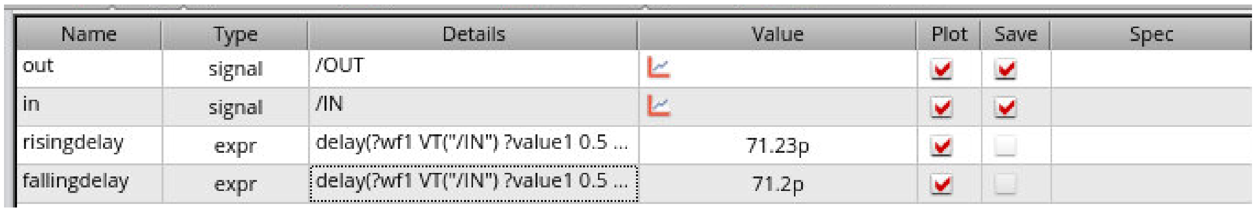
Falling Delay
delay(?wf1 VT("/IN"), ?value1 0.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 VT("/OUT"), ?value2 0.5, ?edge2 "falling", ?nth2 1, ?tol2 nil, ?td2 nil , ?stop nil, ?multiple nil)

Rising Delay
delay(?wf1 VT("/IN"), ?value1 0.5, ?edge1 "falling", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 VT("/OUT"), ?value2 0.5, ?edge2 "rising", ?nth2 1, ?tol2 nil, ?td2 nil , ?stop nil, ?multiple nil)



Signal1	VT("/IN")		
Signal2	VT("/OUT")		
Threshold Value 1	0.5	Threshold Value 2	0.5
Edge Number 1	1	Edge Number 2	1
Edge Type 1	rising	Edge Type 2	falling
Periodicity 1	1	Periodicity 2	1
Tolerance 1	nil	Tolerance 2	nil

Clicking OK or Apply to send the expression to the input window. From here, you can either choose Tools > Plot from the calculator window to receive a direct numerical value, or save it as a simulation output expression to have it calculate every time the simulation is run in case any changes are made.

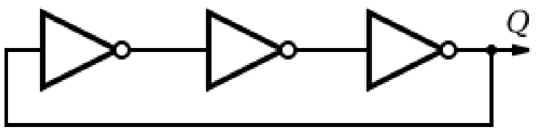


Name	Type	Details	Value	Plot	Save	Spec
out	signal	/OUT		✓	✓	
in	signal	/IN		✓	✓	
risingdelay	expr	delay(?wf1 VT("/IN") ?value1 0.5 ...	71.23p	✓		
fallingdelay	expr	delay(?wf1 VT("/IN") ?value1 0.5 ...	71.2p	✓		

Note that due to the PMOS sizing from the schematic section, the rising and falling delays are nearly identical. Feel free to go back and change the transistor dimensions in the schematic, check & save, and rerun the simulation to see the effects of each parameter. For more information on ViVA calculator functions, see this link.

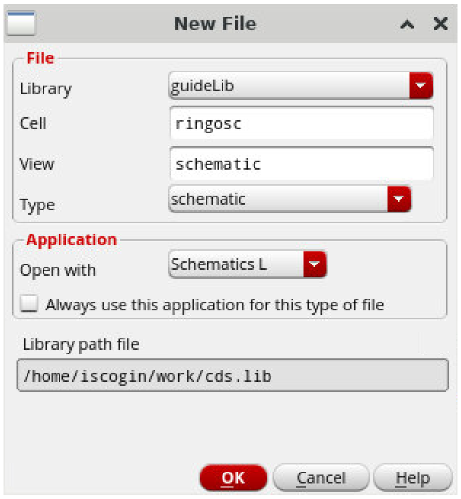
Hierarchical Design - Ring Oscillator

With the inverters simulated and verified, they can now be used in a larger design. A ring oscillator is a series of inverters with an odd number of stages that connects in a circular pattern. Due to the odd number of stages, the system is not able to reach a stable condition and will oscillate according to the propagation delay, number of stages, and loads (parasitic and intended) at each stage. A simple diagram depicting a 3-stage ring oscillator is shown below.



The oscillation frequency f of a ring oscillator is given by 1/(2*N*Tp) where N is the number of stages and Tp is the propagation delay.

Create a new schematic cellview in guideLib titled “ringosc” using the same process as before.


New File

File

Library: guideLib

Cell: ringosc

View: schematic

Type: schematic

Application

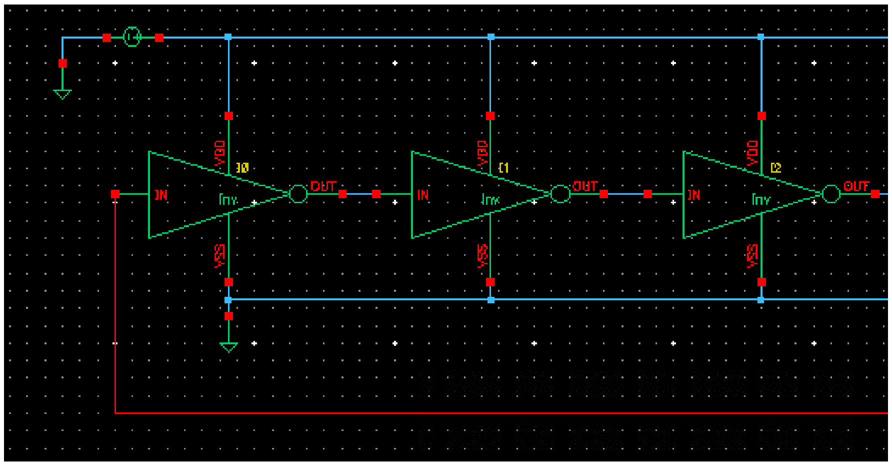
Open with: Schematics L

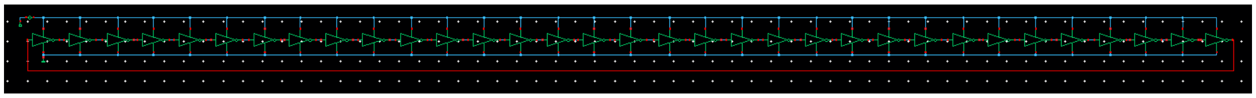
☐ Always use this application for this type of file

Library path file

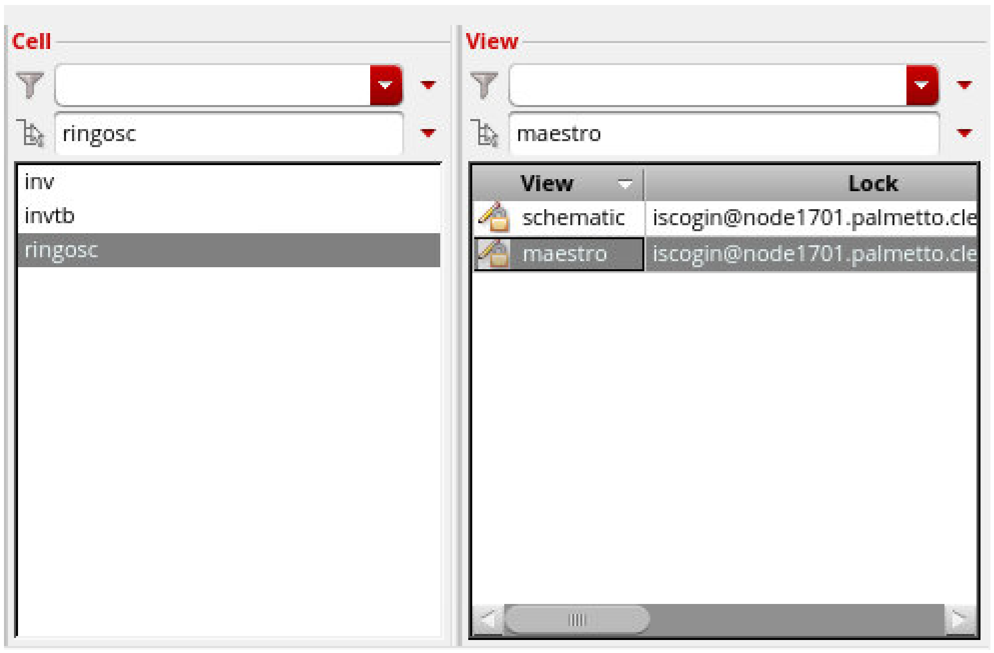
/home/iscogin/work/cds.lib

For this design, I am arbitrarily choosing N = 35 Stages. Place the inverters in series and connect them sequentially, looping the final inverter’s output back to the input of the first inverter. Use a 1V source for VDD and a gnd symbol for ground. Note that the large number of stages is to ensure that the frequency of the oscillator is not inhibited by the rise/fall time of the inverters, and that the actual propagation delay / switching time of each inverter will change due to nonideal factors such as loading, high-frequency parasitics, etc…





Because this ring oscillator design does not have any external I/O, there is no need to make a testbench. Despite this, you could still add pins and make a symbol view similar to before if you wanted to use the whole oscillator as a single block. From the ringosc schematic, launch ADE Explorer and create a maestro view under ringosc.



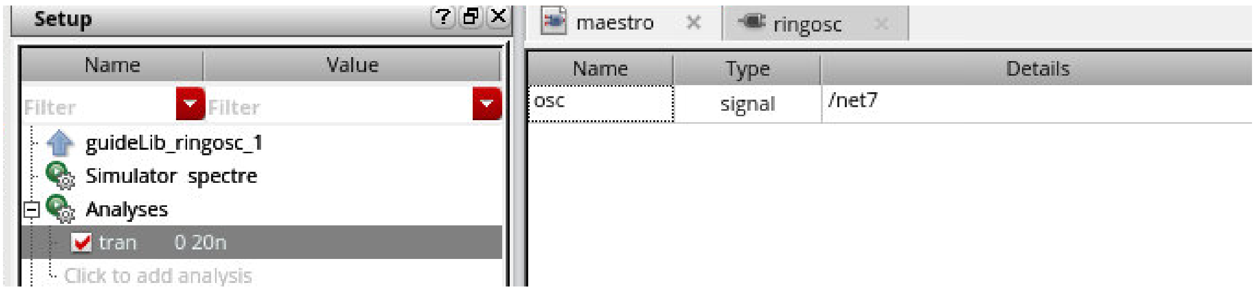
ringosc

Cell
inv
invtb
ringosc

maestro

View	Lock
schematic	iscogin@node1701.palmetto.cle
maestro	iscogin@node1701.palmetto.cle

Similar to before, ensure that the Model Libraries and Job Policy have been correctly defined. Define a transient simulation for 50 nanoseconds (50n) and select any inverter output as the oscillator output for plotting.

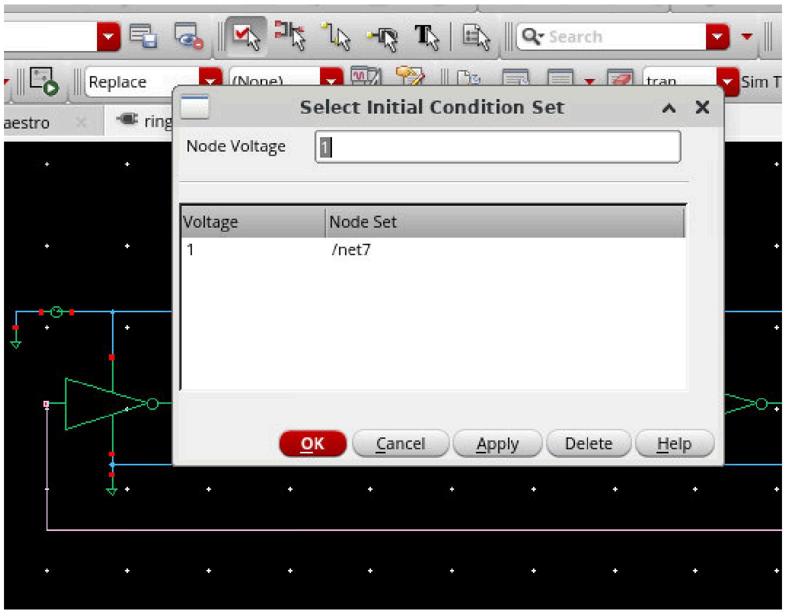


Name	Value
guideLib_ringosc_1	
Simulator	spectre
Analyses	
tran	0 20n

Name	Type	Details
osc	signal	/net7

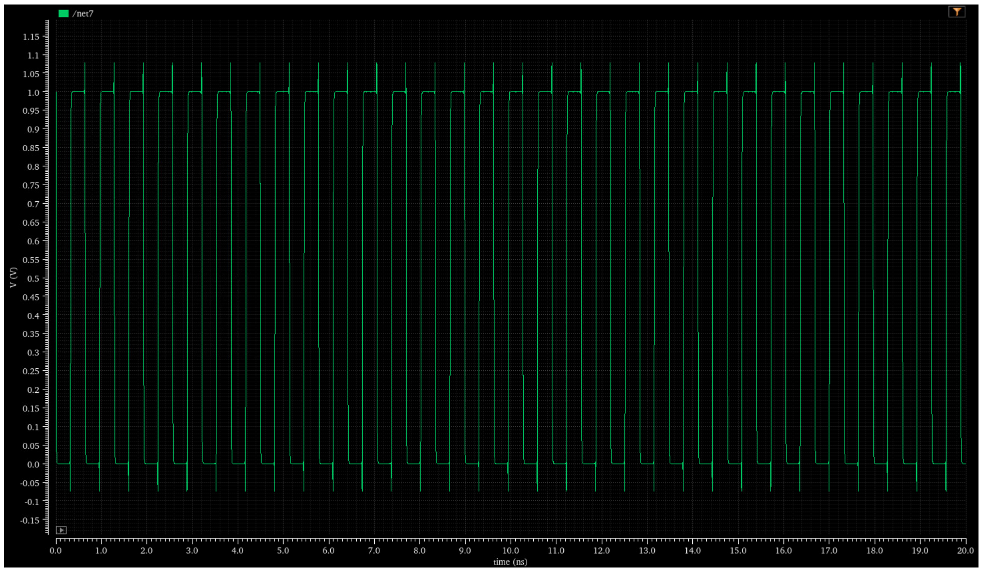
A ring oscillator is an unstable, high-frequency design, so it is good practice to use convergence aids to help the simulator converge on the proper output. The two types of convergence aids are node sets and initial conditions. Nodesets provide an initial guess for a node's output to help the simulator converge, but do not necessarily carry over into the simulation outputs. Initial conditions are much more powerful, as they force the simulation output at a specific node at t=0. However, initial conditions can cause convergence issues if they are used improperly.

Since a ring oscillator relies on thermal noise to get started (something that does not exist in a simulation), a nodeset may not be adequate to kickstart convergence. In this case, we will choose an initial condition by going to Simulation > Convergence Aids > Initial Condition… from within the ADE window. Set the Node Voltage as 1 and click one of the nets between two inverters. Click OK and run the simulation. If you do not see any oscillations, try adjusting the period by adding more stages to your oscillator.



Node Voltage: 1

Voltage	Node Set
1	/net7


/net7
V (V)
time (ns)