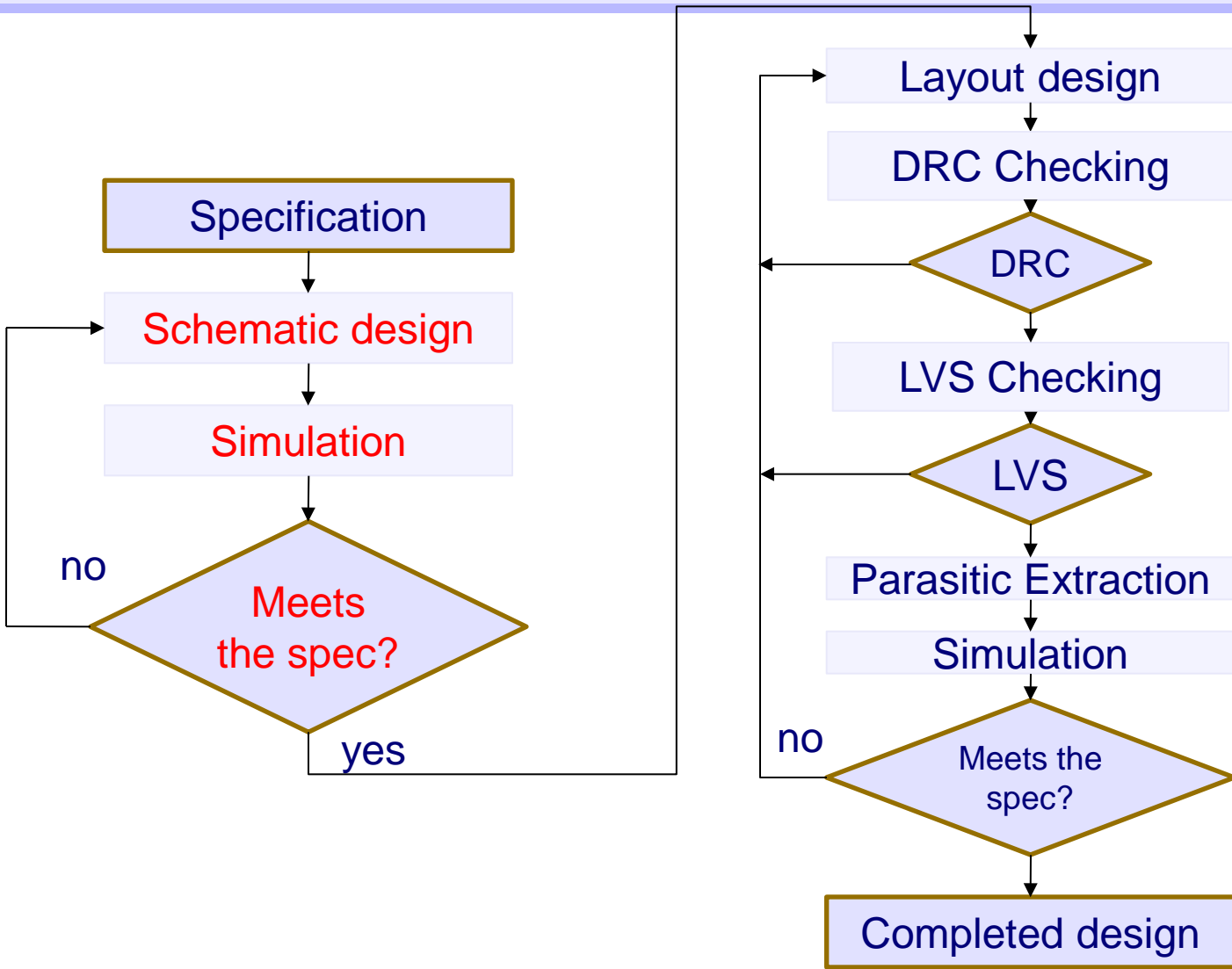


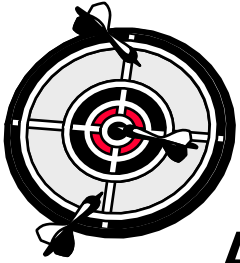
Symbols & Simulation

Dr Basel Halak

Custom Design Flow



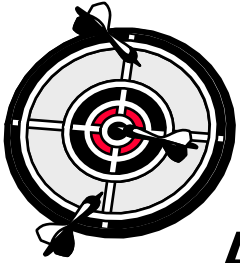
Learning Outcomes



After completing this unit, you should be able to:

- **Create a symbol from a schematic**
- **Set up the simulation environment**
- **Simulate your design**
- **Plot the results**
- **Use parameter passing and design variables**
- **Use parametric analysis**

Learning Outcomes



After completing this unit, you should be able to:

- **Create a symbol from a schematic**
- **Set up the simulation environment**
- **Simulate your design**
- **Plot the results**
- **Use parameter passing and design variables**
- **Use parametric analysis**

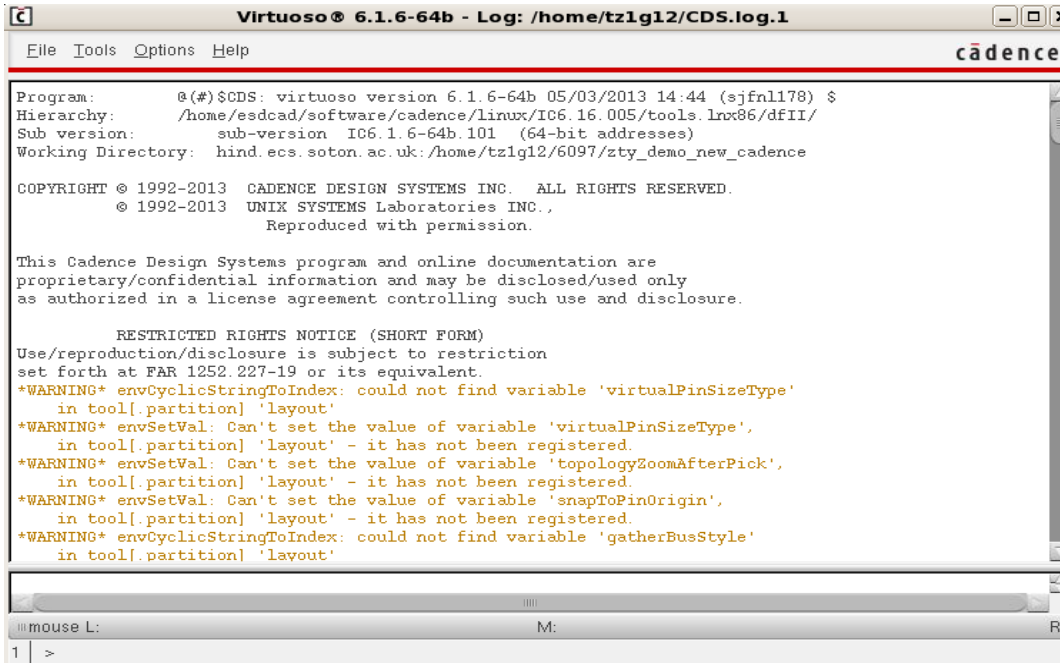
Set up the current working directory

- **The first step is to set up the local working directory for the AMS 0.35 μ m process.**
- **open a new terminal and type the following:**
tcsh
source /opt/esdcad/scripts/ams_v400_tcshrc



Run Cadence Design Manager

- This will bring up the Cadence Design Manager:



```
Virtuoso® 6.1.6-64b - Log: /home/tz1g12/CDS.log.1
File Tools Options Help
cadence

Program:      @(#)SCDS: virtuoso version 6.1.6-64b 05/03/2013 14:44 (sjfnll178) $
Hierarchy:    /home/esdcad/software/cadence/linux/IC6.16.005/tools.lnx86/dfII/
Sub version:  sub-version IC6.1.6-64b.101 (64-bit addresses)
Working Directory:  hind.ecs.soton.ac.uk:/home/tz1g12/6097/zty_demo_new_cadence

COPYRIGHT © 1992-2013 CADENCE DESIGN SYSTEMS INC. ALL RIGHTS RESERVED.
© 1992-2013 UNIX SYSTEMS Laboratories INC.,
Reproduced with permission.

This Cadence Design Systems program and online documentation are
proprietary/confidential information and may be disclosed/used only
as authorized in a license agreement controlling such use and disclosure.

RESTRICTED RIGHTS NOTICE (SHORT FORM)
Use/reproduction/disclosure is subject to restriction
set forth at FAR 1252.227-19 or its equivalent.
*WARNING* envCyclicStringToIndex: could not find variable 'virtualPinSizeType'
in tool[.partition] 'layout'
*WARNING* envSetVal: Can't set the value of variable 'virtualPinSizeType',
in tool[.partition] 'layout' - it has not been registered.
*WARNING* envSetVal: Can't set the value of variable 'topologyZoomAfterPick',
in tool[.partition] 'layout' - it has not been registered.
*WARNING* envSetVal: Can't set the value of variable 'snapToPinOrigin',
in tool[.partition] 'layout' - it has not been registered.
*WARNING* envCyclicStringToIndex: could not find variable 'gatherBusStyle'
in tool[.partition] 'layout'

mouse L: M: R:
1 >
```

Create an inverter schematic

- Add a PMOS and NMOS transistor to a schematic from the PRIMLIB library with the following dimensions:

- NMOS 4u/0.35u
- PMOS 12u/0.35u

Add Instance

Library: PRIMLIB

Cell: pmos4

View: symbol

Names:

Add Wire Stubs at:
 all terminals registered terminals only

Array: Rows: 1 Columns: 1

Model name: modp

Width: 12u M

Width stripe: 12u M

Length: 350.00n M

Number of Gates: 1

MOS transistor shape: normal ▼

Number of bends: 1

Top Contact:

Bottom Contact:

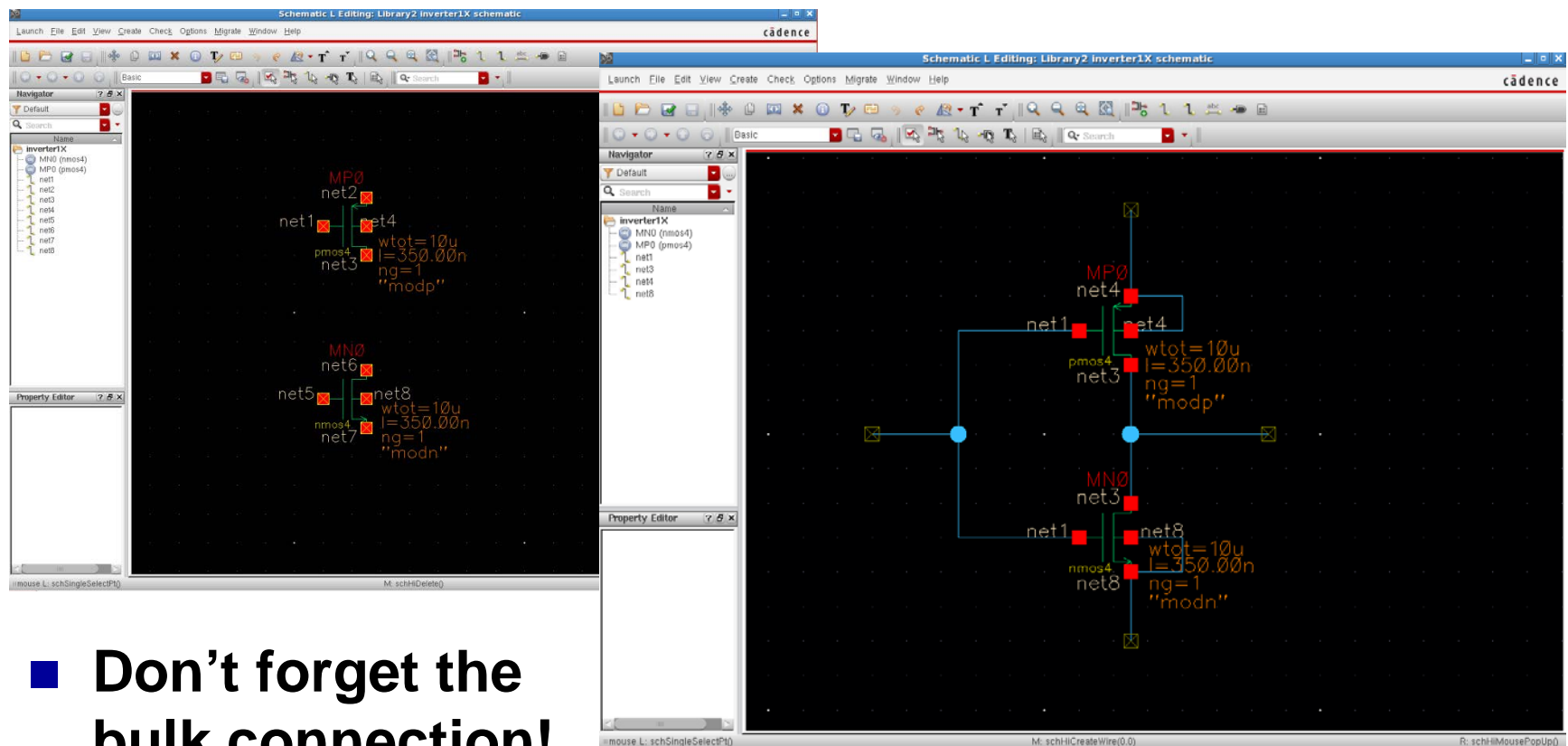
Join Gates: right ▼

Join all drains:

Join all sources:

Create an inverter schematic

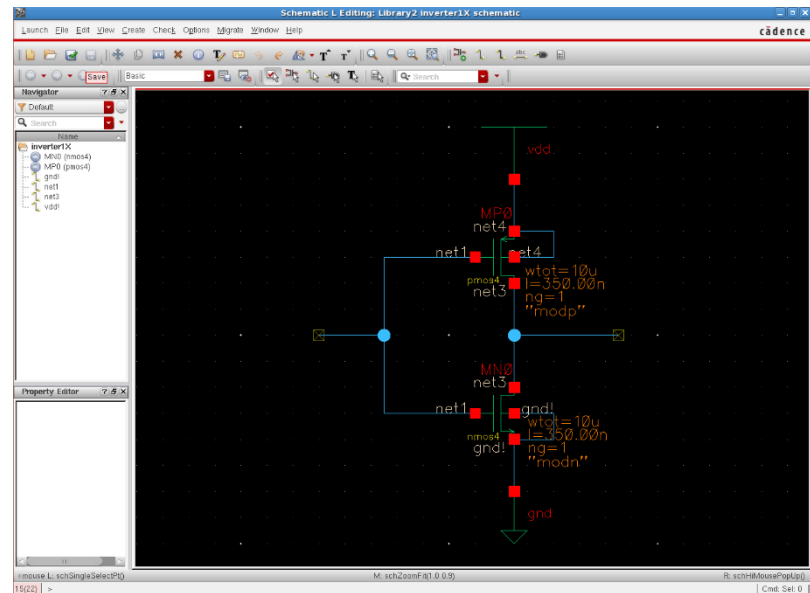
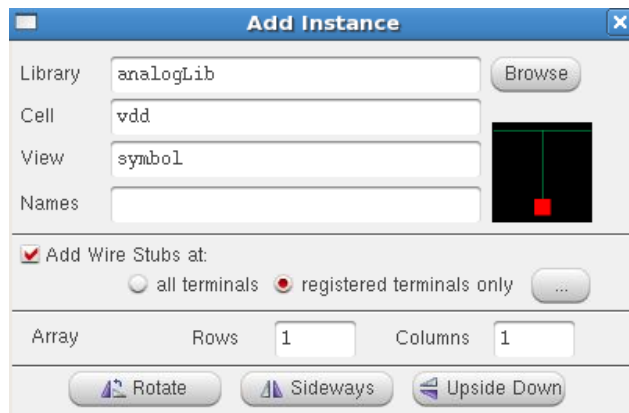
- Then wire them up using the add wire command



- Don't forget the bulk connection!

Connecting things up...

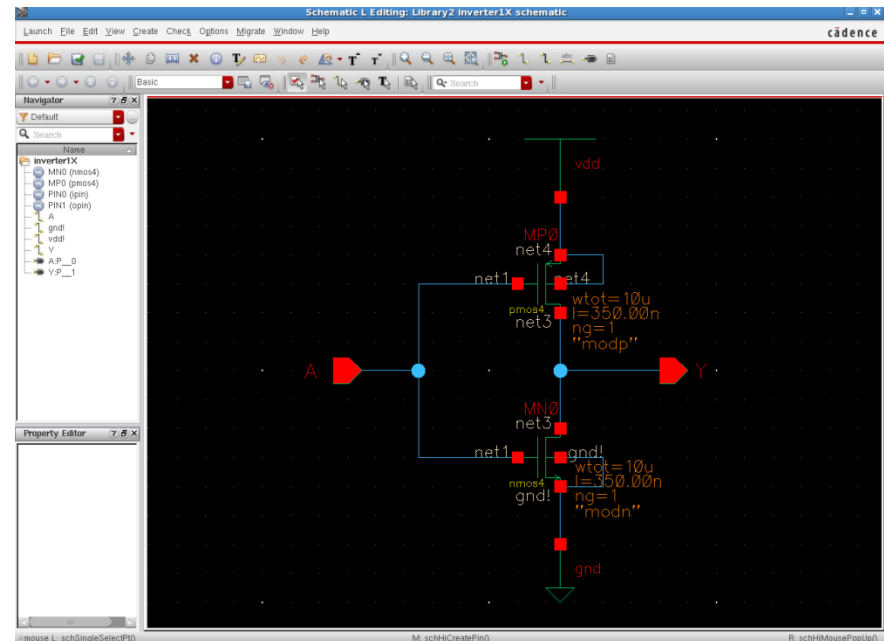
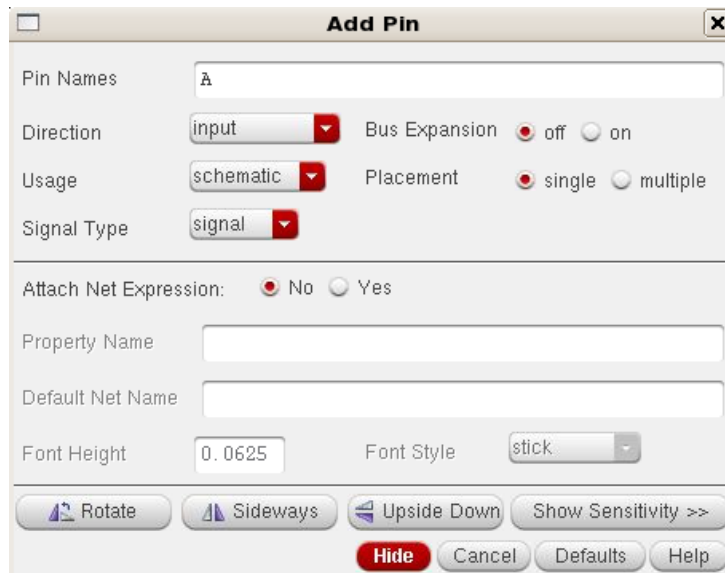
- In Cadence we can add pins for inputs and outputs.
- For Vdd and Ground we use globals.
- Cadence includes symbols which are inherently global – e.g. vdd and gnd.



- Add these to your schematic

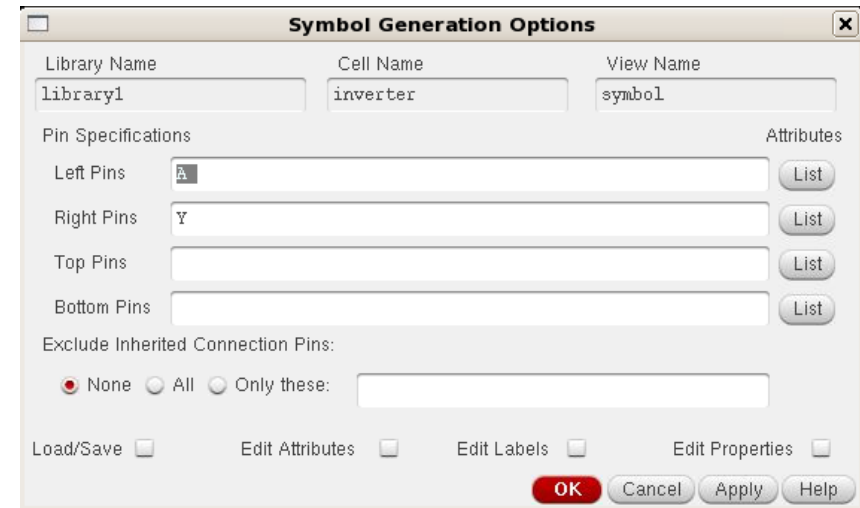
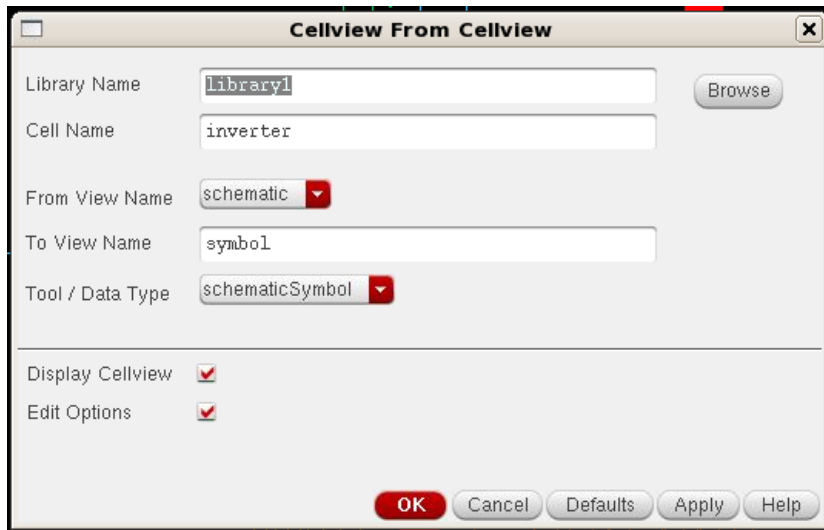
Adding Pins

- Specific pins (inputs and outputs) are added using the Create -> Pins command
- Remember to specify the direction



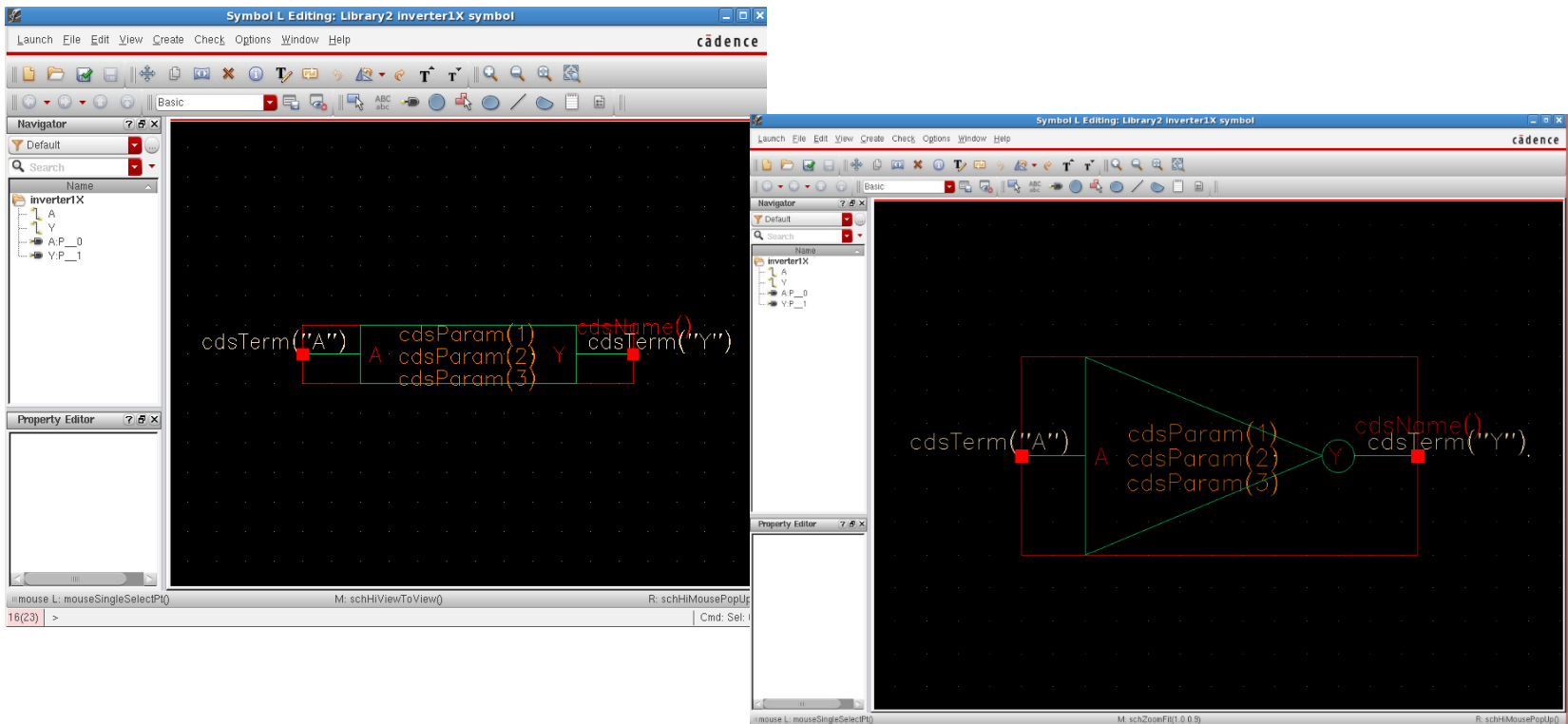
Creating a symbol from a schematic

- To create a symbol from a schematic then we can use the create cell view -> from cell view command in the Create menu



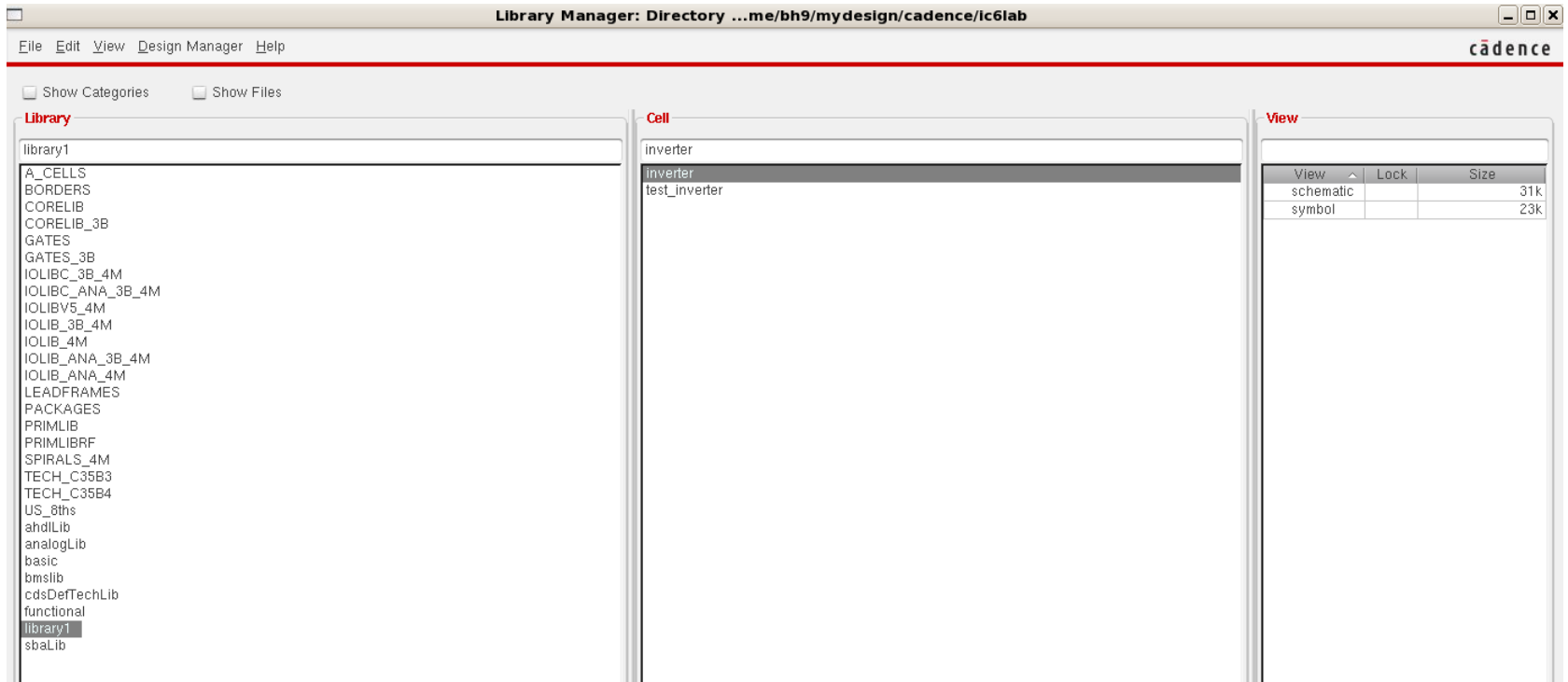
Symbol Creation

- The symbol will be automatically created as shown
- Use the graphics commands from Edit menu :
Move, Delete, Stretch and add to draw the symbol



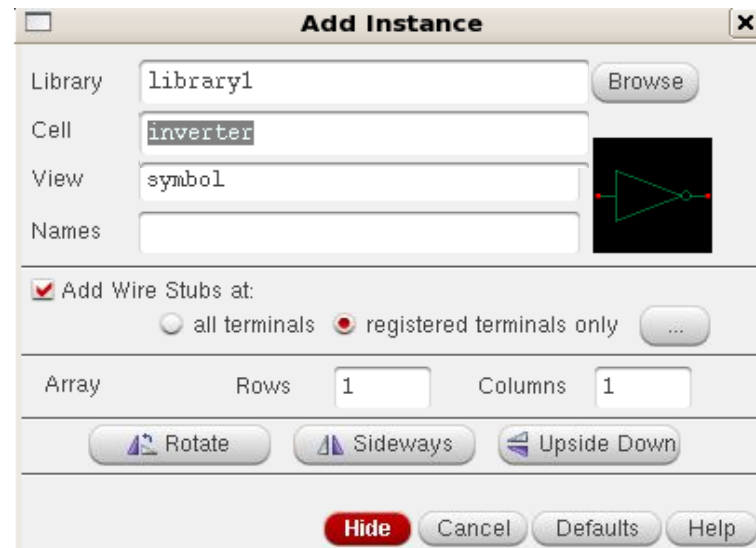
Symbol Creation

- Don't forget to **SAVE** the symbol
- Close it and you will see the new cell view appear in library manager



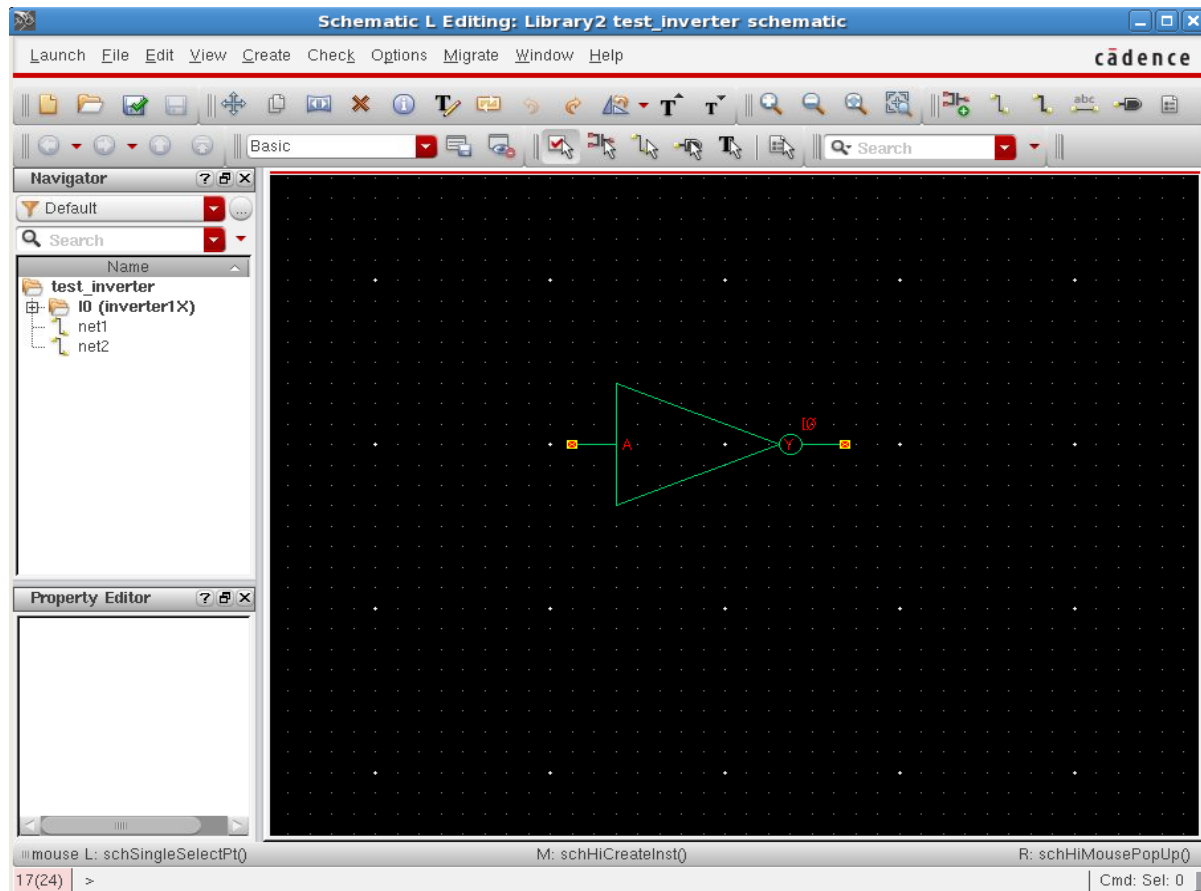
Using the symbol

- We can now use the new symbol in a hierarchical design or a test circuit
- From the library manager – create new cell view, schematic and call it ‘test_inverter’
- Use the Create-> instance command
- Place your new symbol



Placing the symbol on a schematic

- We get the schematic looking like this:



Analog test bench requirements

- **In an analogue test bench we need:**
- **Inputs**
 - sources (dc, transient, ac) depending on the analysis to be carried out
- **Outputs**
 - Measurement models
 - Loads
- **Supplies**
 - Vdd, Vpulse
 - Ground
 - gnd



test_inverter – test bench requirements

■ Inputs:

- pulse input source between 0 and 3.3V at about 100MHz

■ Outputs:

- capacitive load of about 1pF

■ Supplies:

- $V_{dd} = 3.3V$

■ Ground:

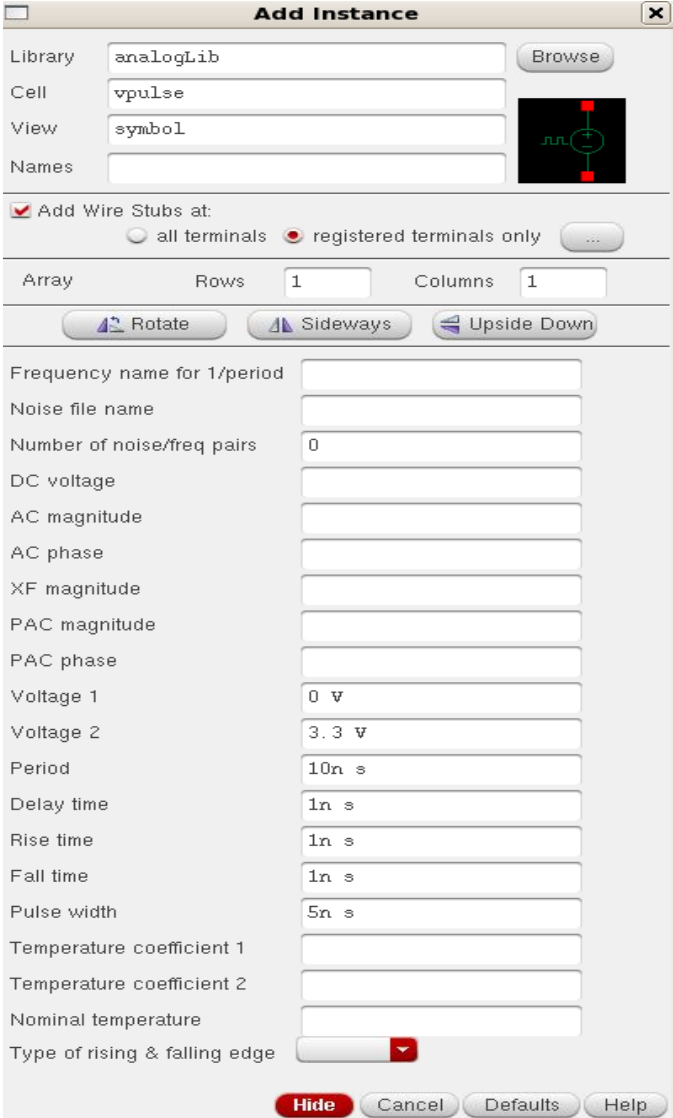
- $gnd = 0V$



Specifying the vpulse instance

■ analogLib -> vpulse

- $v1 = 0$
- $v2 = 3.3$
- $\text{delay} = 1\text{n}$
- $\text{rise} = 1\text{n}$
- $\text{fall} = 1\text{n}$
- $\text{pulse width} = 5\text{n}$
- $\text{period} = 10\text{n}$

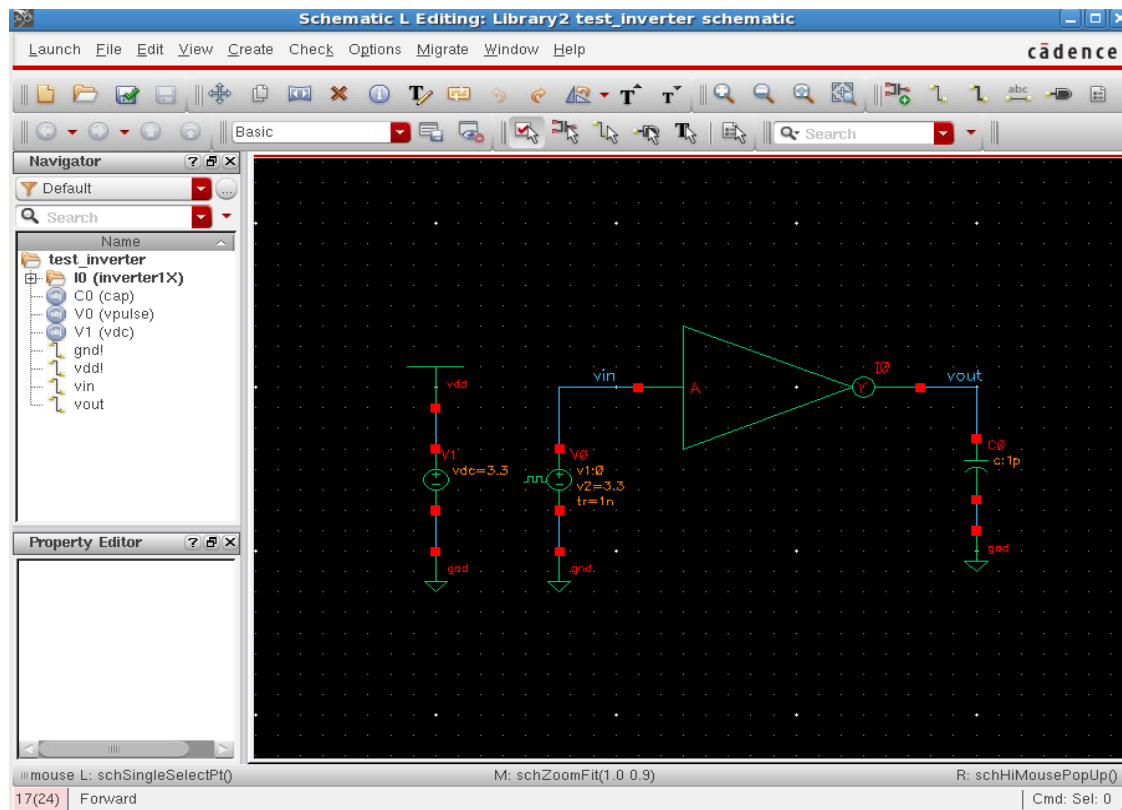


The screenshot shows the 'Add Instance' dialog box for the 'vpulse' component. The dialog is titled 'Add Instance' and has a close button (X) in the top right corner. It contains the following fields and controls:

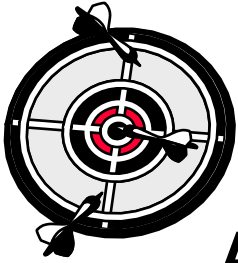
- Library: analogLib (with a 'Browse' button)
- Cell: vpulse
- View: symbol
- Names: (empty field)
- Preview: A small schematic diagram showing a pulse source symbol with two terminals.
- Options: Add Wire Stubs at: all terminals registered terminals only (with a '...' button)
- Array: Rows: 1, Columns: 1
- Orientation:
- Frequency name for 1/period: (empty field)
- Noise file name: (empty field)
- Number of noise/freq pairs: 0
- DC voltage: (empty field)
- AC magnitude: (empty field)
- AC phase: (empty field)
- XF magnitude: (empty field)
- PAC magnitude: (empty field)
- PAC phase: (empty field)
- Voltage 1: 0 V
- Voltage 2: 3.3 V
- Period: 10n s
- Delay time: 1n s
- Rise time: 1n s
- Fall time: 1n s
- Pulse width: 5n s
- Temperature coefficient 1: (empty field)
- Temperature coefficient 2: (empty field)
- Nominal temperature: (empty field)
- Type of rising & falling edge: (dropdown menu with a red arrow pointing down)
- Buttons: Hide (red), Cancel, Defaults, Help

Complete the test circuit

- Wire up the source and load
- Give the wires meaningful names: you can name a wire by selecting it then write click and select Add name from the menu



Learning Outcomes



After completing this unit, you should be able to:

- Create a symbol from a schematic
- **Set up the simulation environment**
- **Simulate your design**
- **Plot the results**
- Use parameter passing and design variables
- Use parametric analysis

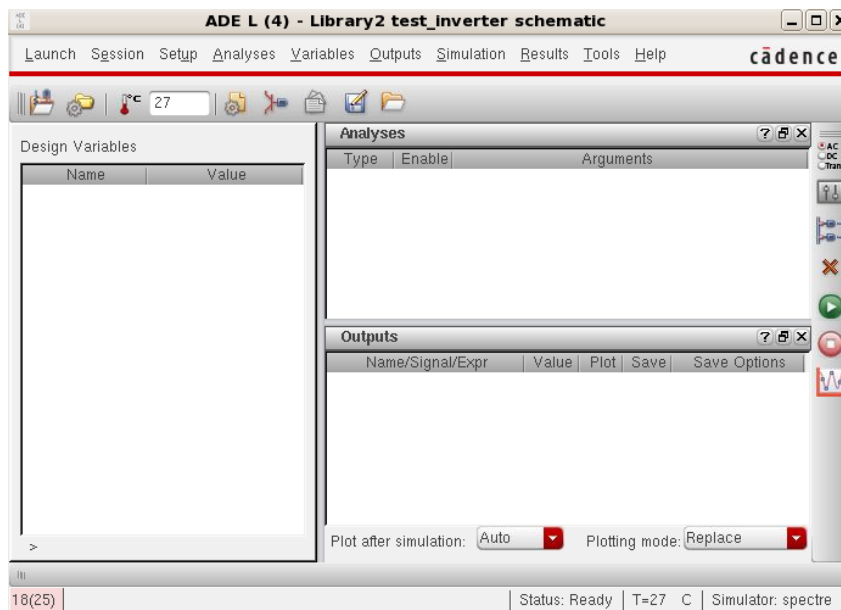
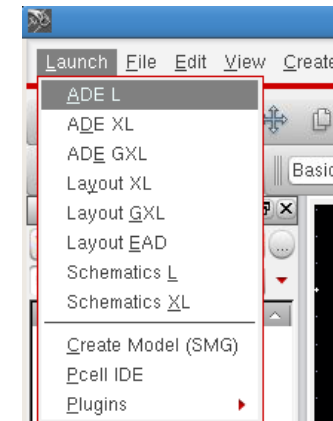
The Cadence simulator

- **Now we want to simulate this circuit using the Cadence simulator: spectre**
- **The simulator environment in Cadence is called Analog Design Environment (ADE) and allows you to run simulations and plot results**
- **Although it looks pretty boring(!) it is actually very powerful**



How to start Analog Design Environment

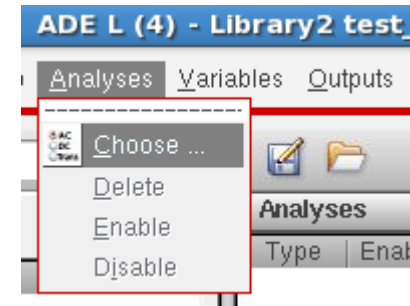
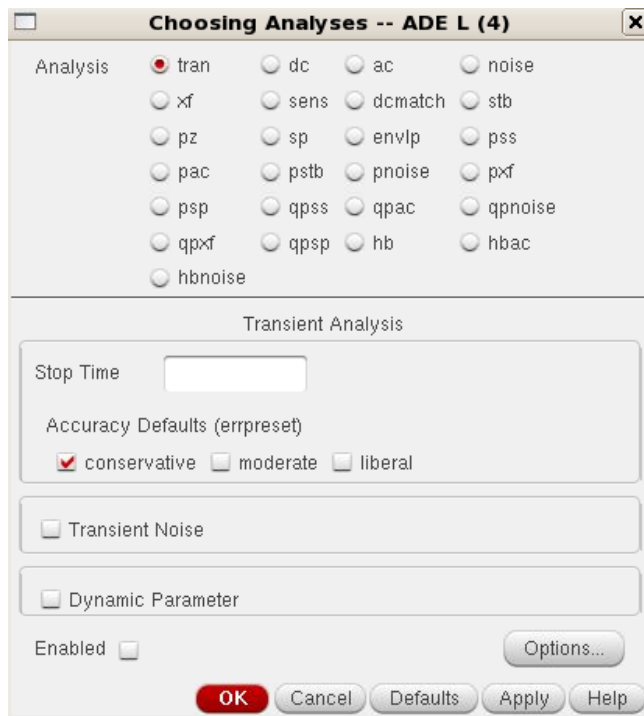
- Open your test circuit and go to Launch->ADE L.
- This starts the ADE interface



- You can see here we are using spectre simulator and that it has initialised the environment with our test circuit

Types of simulation

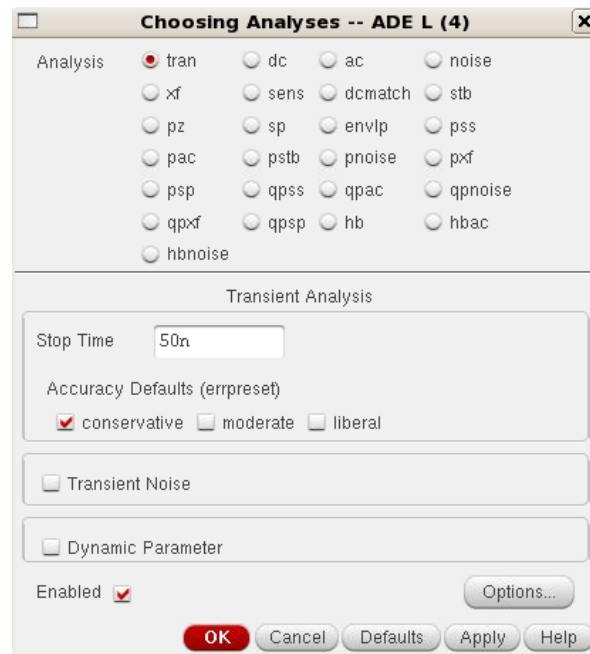
- We can choose from many different types of simulation
- Click the choose simulation button



- First we will set up a transient (i.e. time domain) simulation

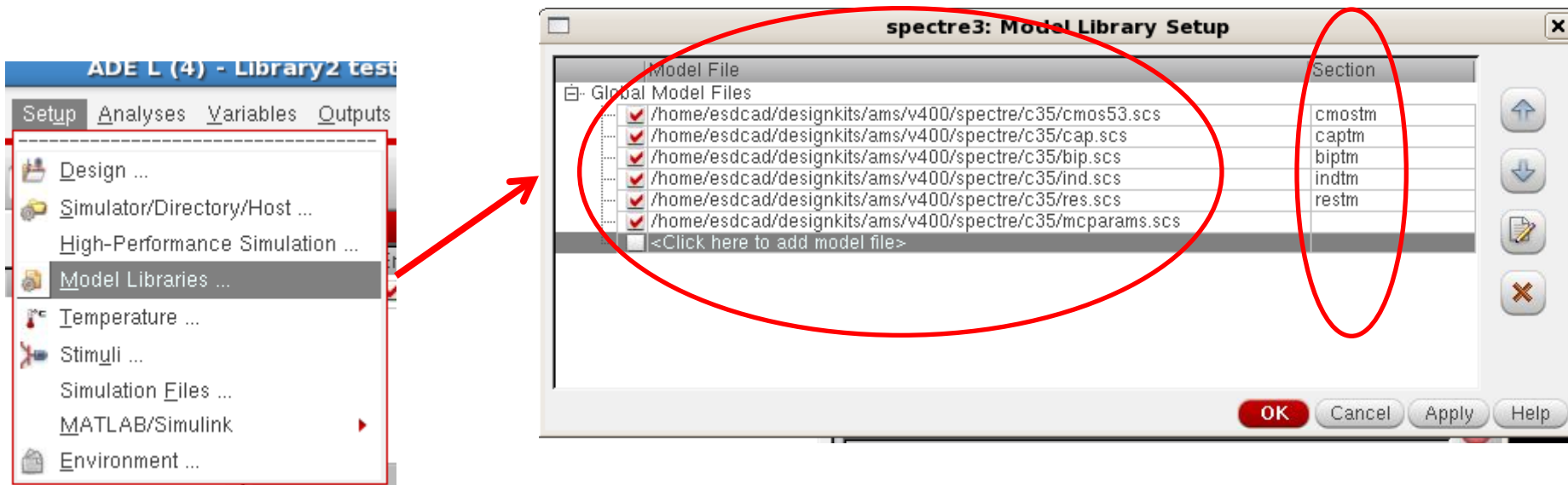
Transient simulation setup

- **Different simulation accuracies are available**
 - Conservative is the most accurate but the slowest
- **In this example we will run the simulation for 5 cycles (50ns)**



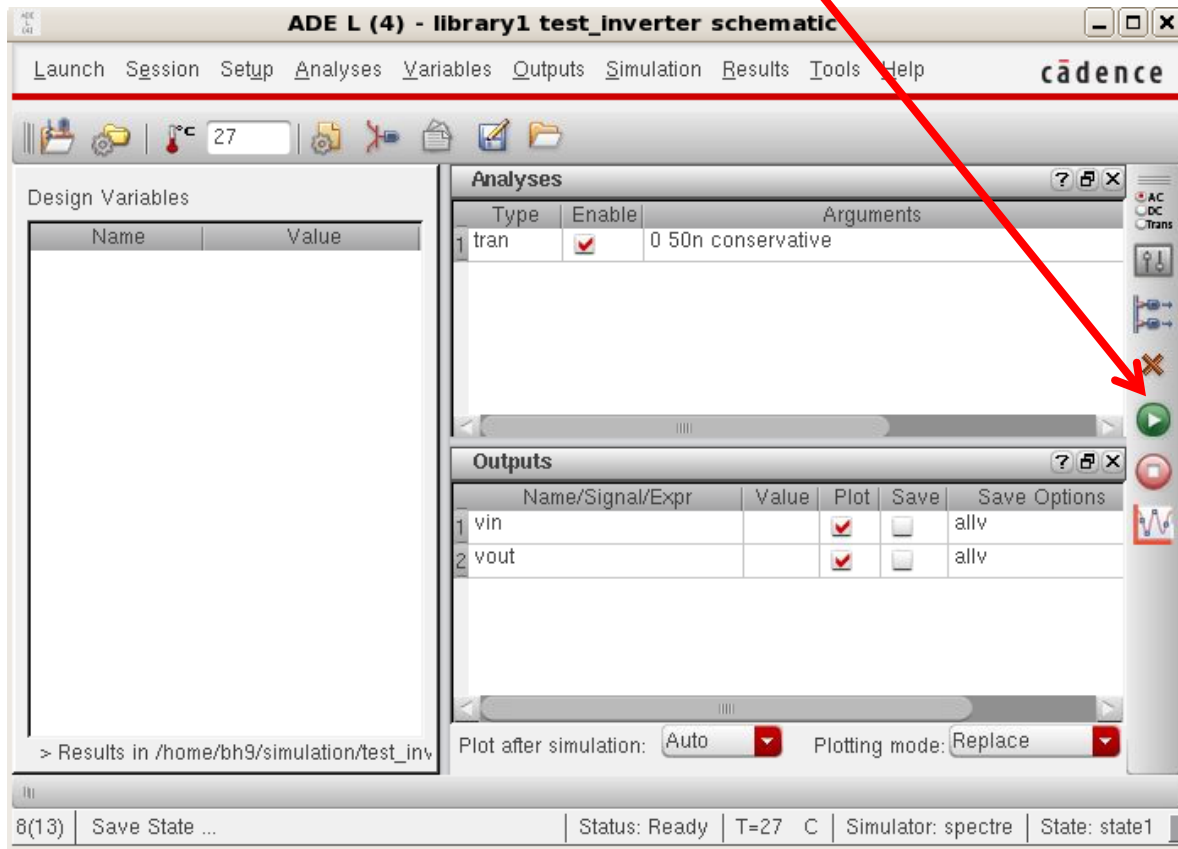
Running the simulation

- You need to add the Model Library before you run the simulation. Go to setup



Running the simulation

- Use the 'Netlist and Run' (green arrow) button to run the simulation



Running the simulation

- A window pops up showing the progress of the simulation

```
maxstep = 500 ps
ic = all
skipdc = no
reltol = 10e-06
abstol(V) = 1 uV
abstol(I) = 1 pA
temp = 27 C
trnom = 27 C
tempeffects = all
errpreset = conservative
method = gear2only
lteratio = 10
relref = alllocal
cmin = 0 F
gmin = 1 pS

tran: time = 1.257 ns (2.51 %), step = 20.76 ps (41.5 m%)
tran: time = 3.901 ns (7.8 %), step = 178 ps (356 m%)
tran: time = 6.546 ns (13.1 %), step = 500 ps (1 %)
tran: time = 8.776 ns (17.6 %), step = 28.86 ps (57.7 m%)
tran: time = 11.28 ns (22.6 %), step = 29.11 ps (58.2 m%)
tran: time = 13.8 ns (27.6 %), step = 153 ps (306 m%)
tran: time = 16.29 ns (32.6 %), step = 500 ps (1 %)
tran: time = 18.77 ns (37.5 %), step = 28.44 ps (56.9 m%)
tran: time = 21.25 ns (42.5 %), step = 25.46 ps (50.9 m%)
tran: time = 23.77 ns (47.5 %), step = 146 ps (292 m%)
tran: time = 26.6 ns (53.2 %), step = 402.7 ps (805 m%)
tran: time = 28.77 ns (57.5 %), step = 28.46 ps (56.9 m%)
tran: time = 31.25 ns (62.5 %), step = 24.88 ps (49.8 m%)
tran: time = 33.77 ns (67.5 %), step = 146.1 ps (292 m%)
tran: time = 36.6 ns (73.2 %), step = 401.9 ps (804 m%)
tran: time = 38.77 ns (77.5 %), step = 28.46 ps (56.9 m%)
tran: time = 41.25 ns (82.5 %), step = 24.88 ps (49.8 m%)
tran: time = 43.77 ns (87.5 %), step = 146.1 ps (292 m%)
tran: time = 46.6 ns (93.2 %), step = 401.9 ps (804 m%)
tran: time = 48.77 ns (97.5 %), step = 28.46 ps (56.9 m%)

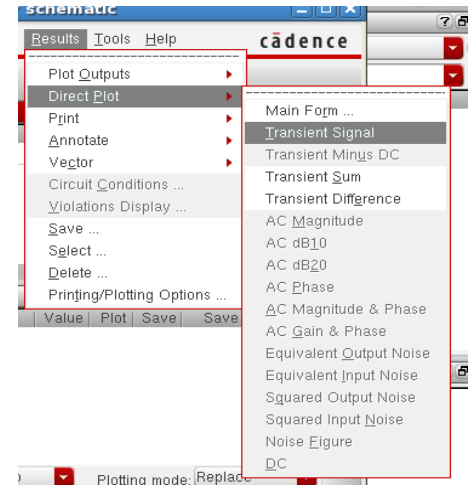
Number of accepted tran steps = 1167
Initial condition solution time: CPU = 1.999 ms, elapsed = 1.194 ms.
Intrinsic tran analysis time: CPU = 88.986 ms, elapsed = 93.2229 ms
Total time required for tran analysis 'tran': CPU = 103.984 ms, elapsed = 105.4228 ms
Time accumulated: CPU = 275.957 ms, elapsed = 1.89327 s.
Peak virtual memory used = 627 Mbytes.

finalTimeOP: writing operating point information to rawfile.
designParamVals: writing netlist parameters to rawfile.
primitives: writing primitives to rawfile.
subckts: writing subcircuits to rawfile.
```

Plotting signals

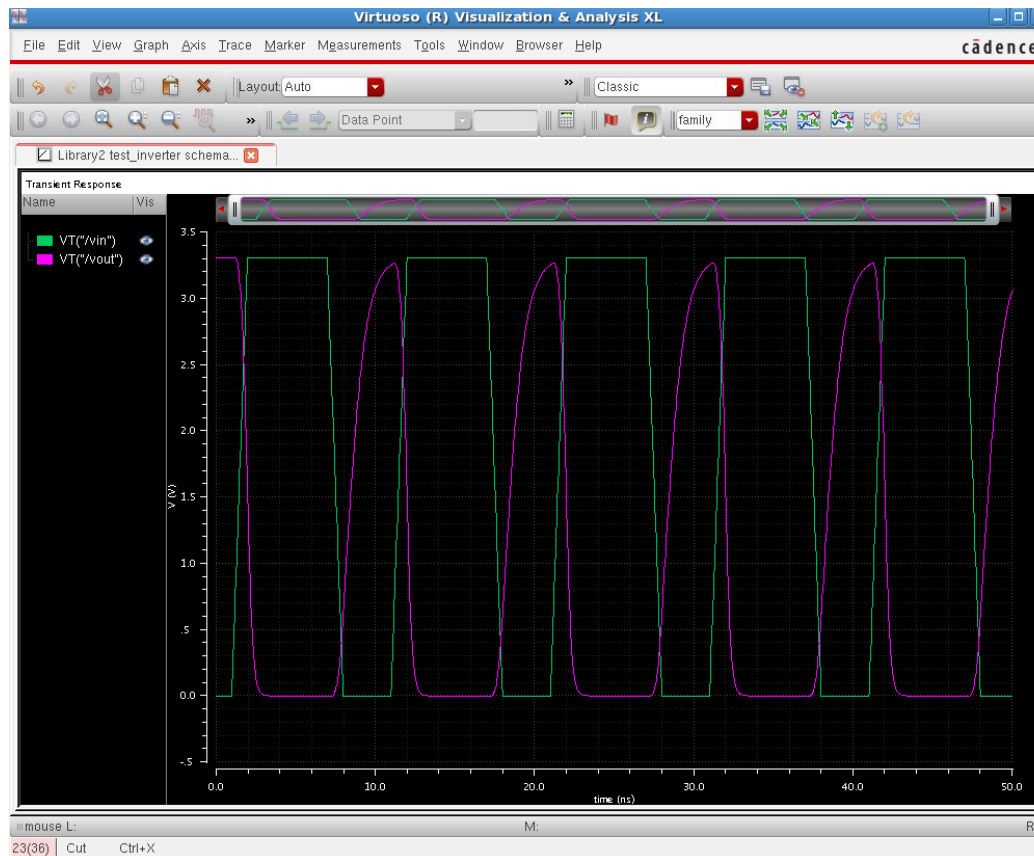
- There are many ways to plot a signal in ADE
- The easiest is with the Direct Plot option
- Choose Direct Plot -> Transient Signal
- Go to the schematic and choose the signals
 - Click a net for a voltage
 - Click a node for a current
- Note the prompt at the bottom of the schematic

> Select nodes or terminals, press <esc> to finish selection



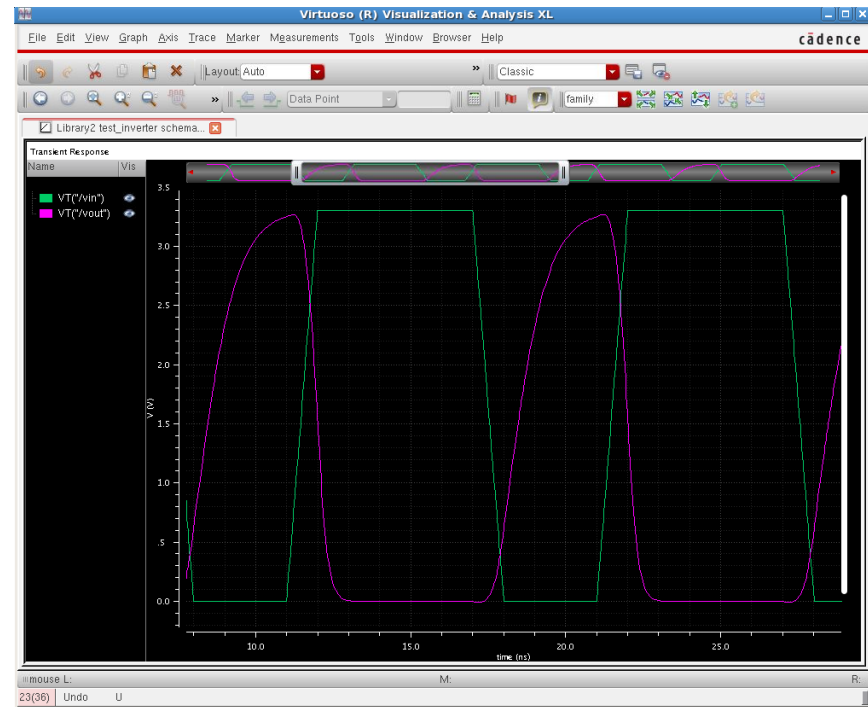
Plotting the signals

- Select the input and output nets to plot the input and output voltages

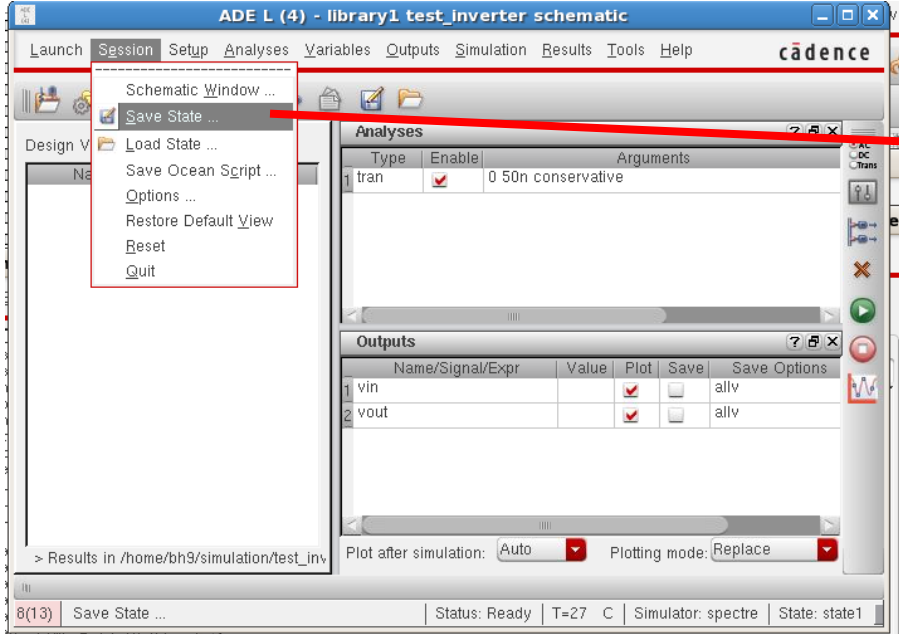


Customise the waveform display

- Right click on a waveform to change properties
 - Waveform name
 - Colour
- Right click and drag to zoom in
- 'f' to re-fit the graph
- 'H' for horizontal marker, 'V' for vertical
- Notice slow rise time



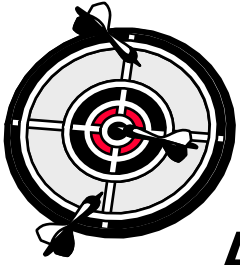
Save your simulation state



Click OK



Learning Outcomes



After completing this unit, you should be able to:

- Create a symbol from a schematic
- Set up the simulation environment
- Simulate your design
- Plot the results
- Use parameter passing and design variables
- Use parametric analysis

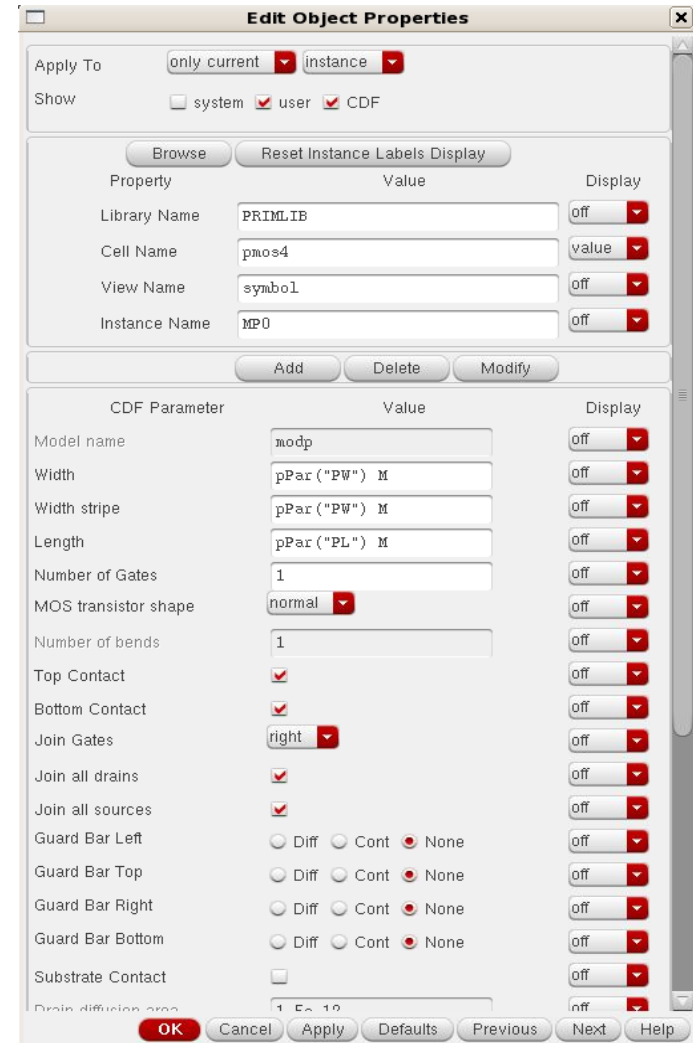
Lets optimise the rise time

- **We want to change the size of the PMOS and see the effect on the rise time**
- **What is the best way of doing this?**
- **Important points:**
 - Hierarchical parameter passing: from the schematic to the symbol
 - Design variables: schematic parameters which can be changed from ADE
 - Parametric simulations: run a set of simulations each with a different value for the design variable



Hierarchical parameter passing

- Open the inverter schematic
- Change the PMOS width and length properties to:
 - pPar("PW")
 - pPar("PL")
- Change the NMOS width and length properties to:
 - pPar("NW")
 - pPar("NL")



Hierarchical parameter passing

- **Recreate the symbol**
- **This will create a symbol where you can set the widths and lengths for that symbol instance**



Hierarchical parameter passing

- Go back to the inverter test schematic
- Re-instantiate the new inverter symbol
- Change the properties of the new symbol as follows
Put in 'PWIDTH' for the PMOS width and keep the others the same
- Check and save

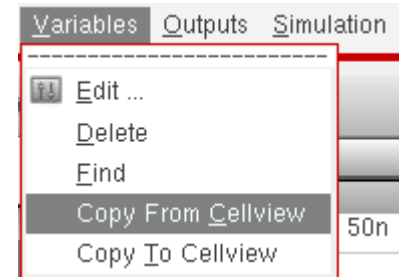
Property	Value	Display
Library Name	Library2	off
Cell Name	inverter1X	off
View Name	symbol	off
Instance Name	I9	off

User Property	Master Value	Local Value	Display
InterfaceLastCh...	17 11:32:31 2015		off

CDF Parameter	Value	Display
NL	0.35u	off
NW	10u	off
PL	0.35u	off
PW	PWIDTH	off

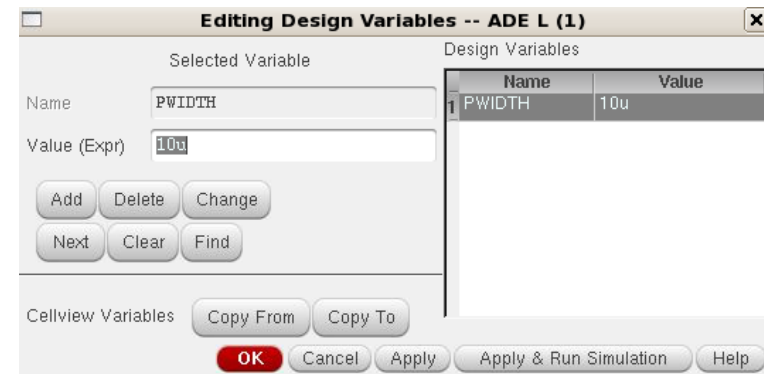
Importing design variables in to ADE

- Go back to the ADE form
- We need ADE to pick up our PWIDTH design variable from the schematic
- Go to Variables->Copy From Cellview
- Notice that PWIDTH appears in the design variables
- Now double click PWIDTH to bring up the design variable box
- Set PWIDTH to 10um initially



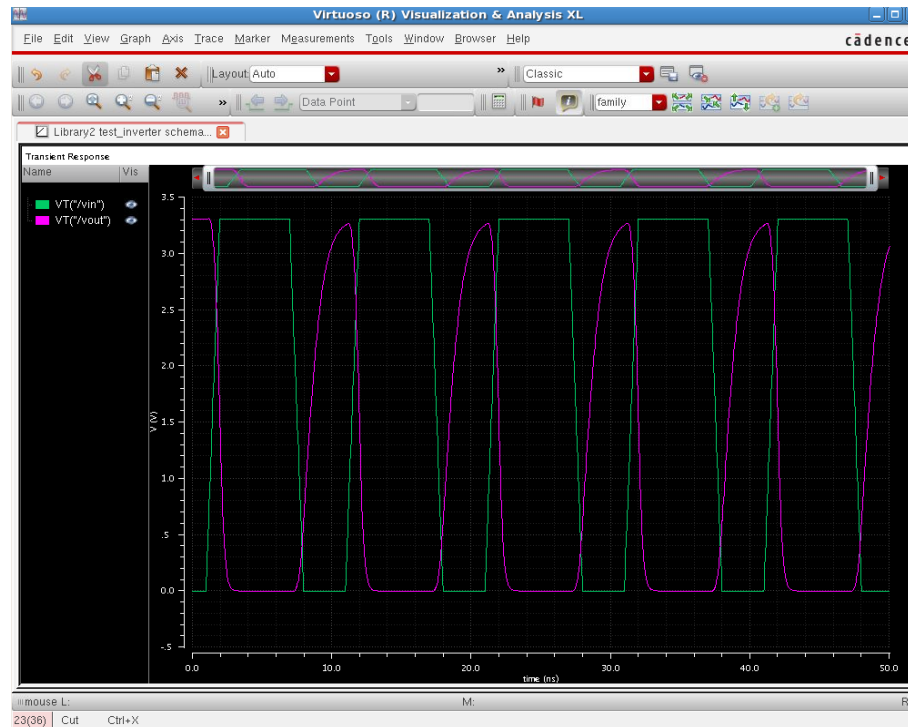
A screenshot of the 'Design Variables' window. It contains a table with two columns: 'Name' and 'Value'. The table has one row with 'PWIDTH' in the 'Name' column and an empty 'Value' column.

Name	Value
PWIDTH	



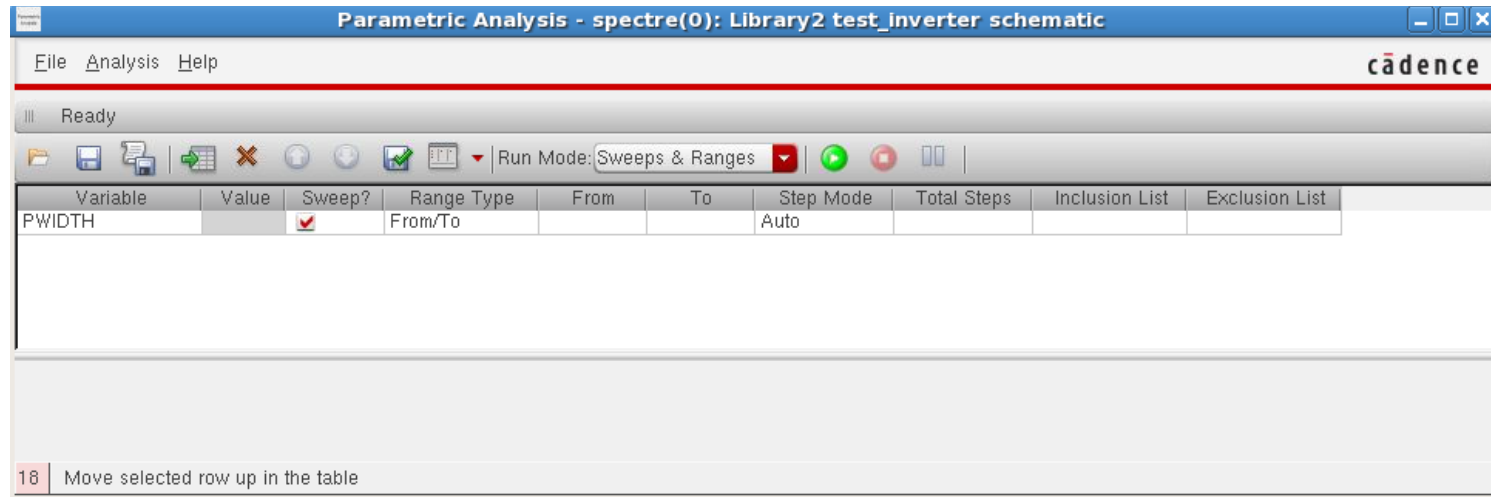
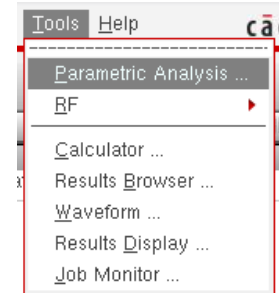
Importing design variables in to ADE

- Re-run the simulation, to check everything is ok
- Should get the same as before, but now we have the ADE set up and ready to do cool stuff



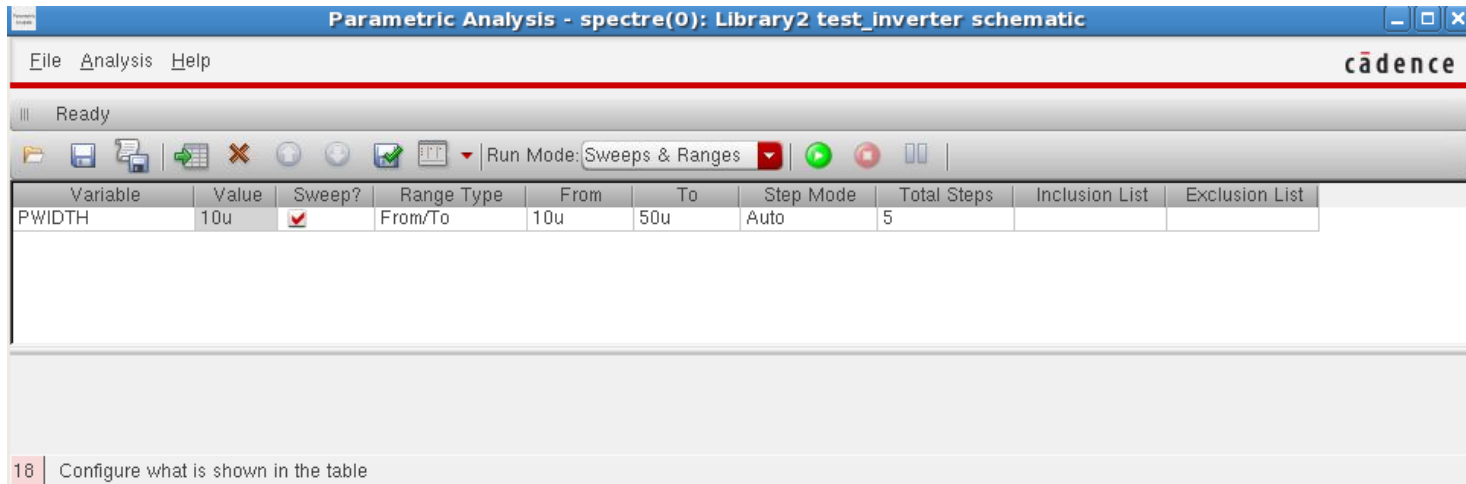
Running a parametric simulation

- Parametric analysis allows you to run multiple simulations whilst varying design parameters
- We will run a parametric simulation during which PWIDTH will be varied



Setting up the parametric simulation

- Put PWIDTH in the parameter box and choose to vary it from 10u to 50u in 5 steps



Run the parametric simulation

- Choose analysis -> start
- The log window will show progress of the runs
- Plot the output using the direct plot option as before
- Zoom into one cycle to examine the rise/fall times

```
2 parametric simulations remaining.

Info: Running PWIDTH=4e-05 1 remaining
Setting PWIDTH = 4e-05
compose simulator input file...
...successful.
start simulator if needed...
...successful.
simulate...
1 parametric simulation remaining.

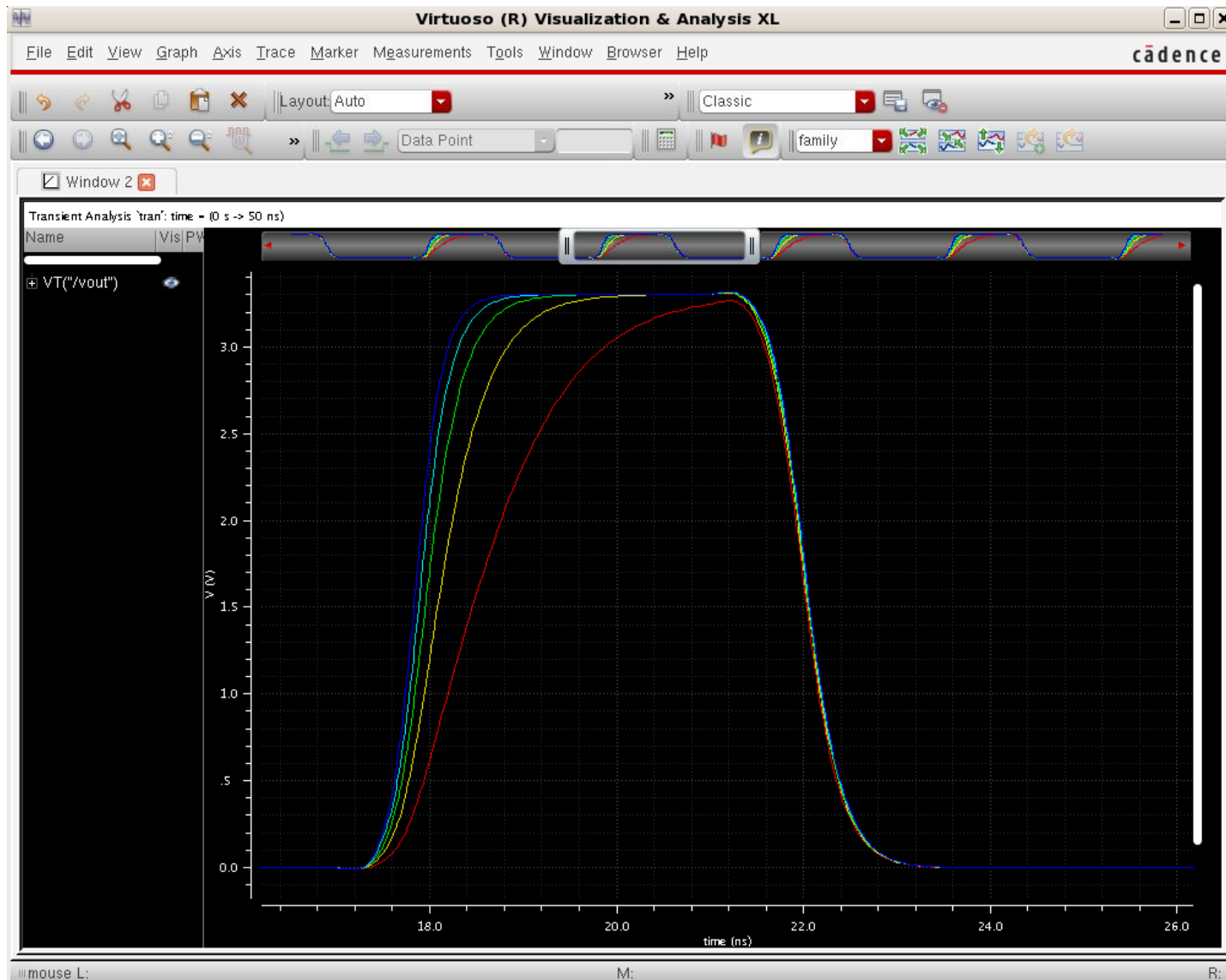
Info: Running PWIDTH=5e-05 last run
Setting PWIDTH = 5e-05
compose simulator input file...
...successful.
start simulator if needed...
...successful.
simulate...
reading simulation data...
...successful.

Info: Parametric Simulation Completed.
```

18 | Configure what is shown in the table



Parametric results

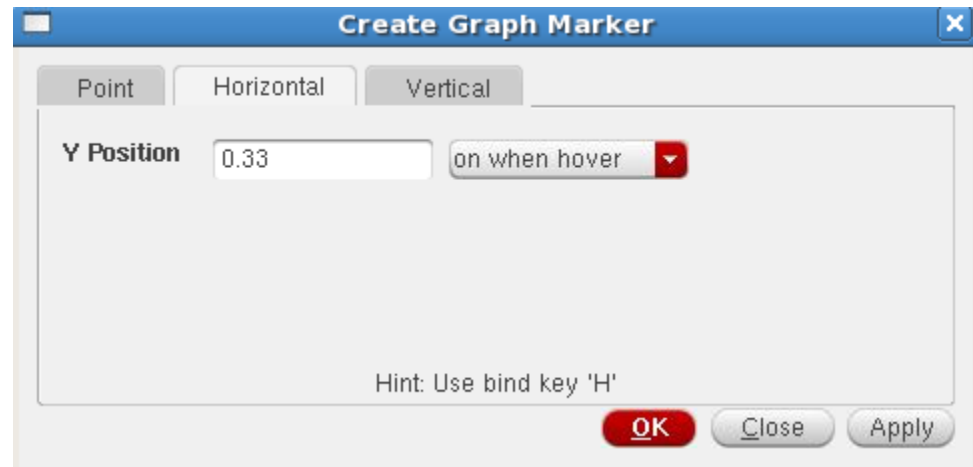
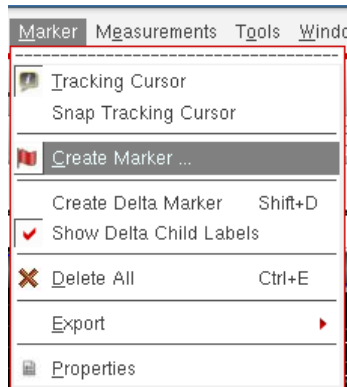


How to measure rise/fall time

■ A few ways

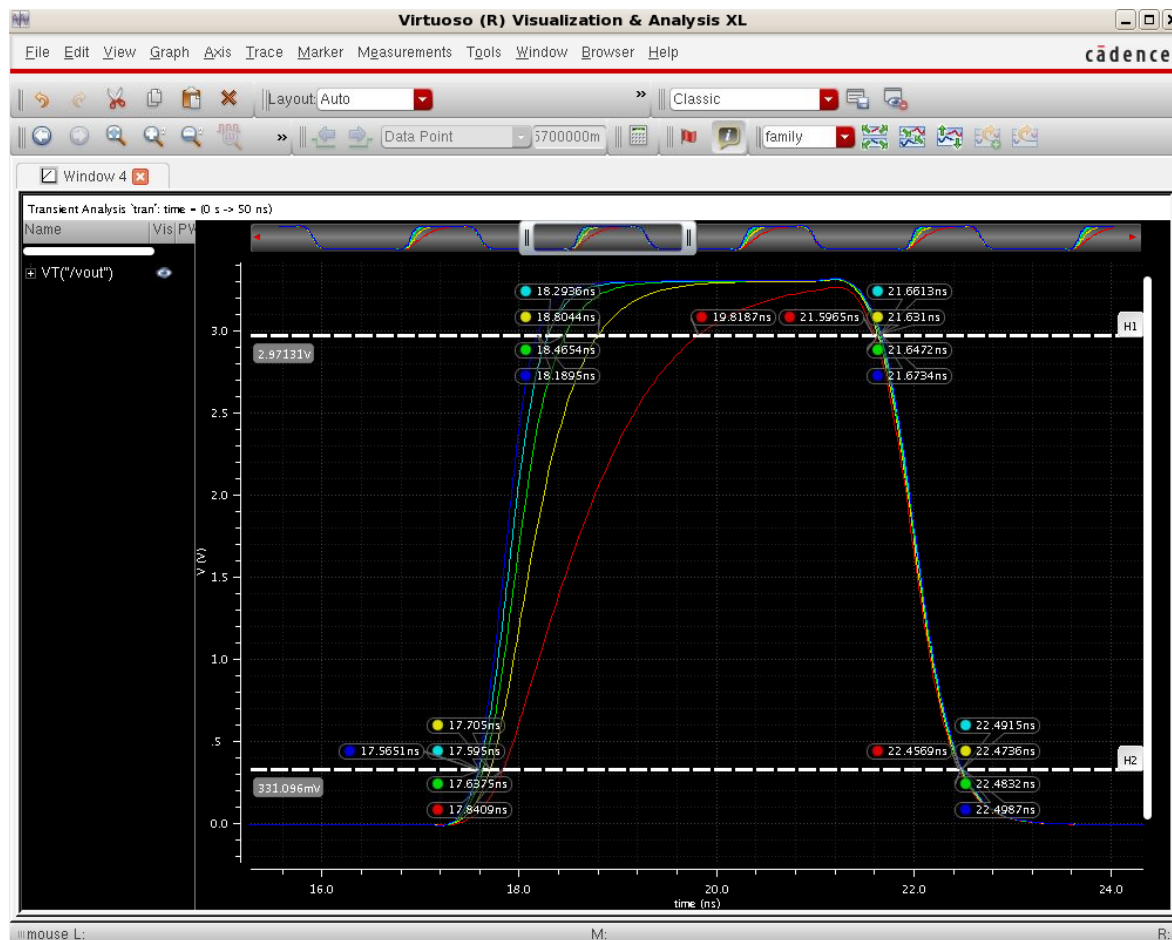
- On an individual graph, use the marker facility
- Use the calculator
- Use horizontal markers

■ Trace marker



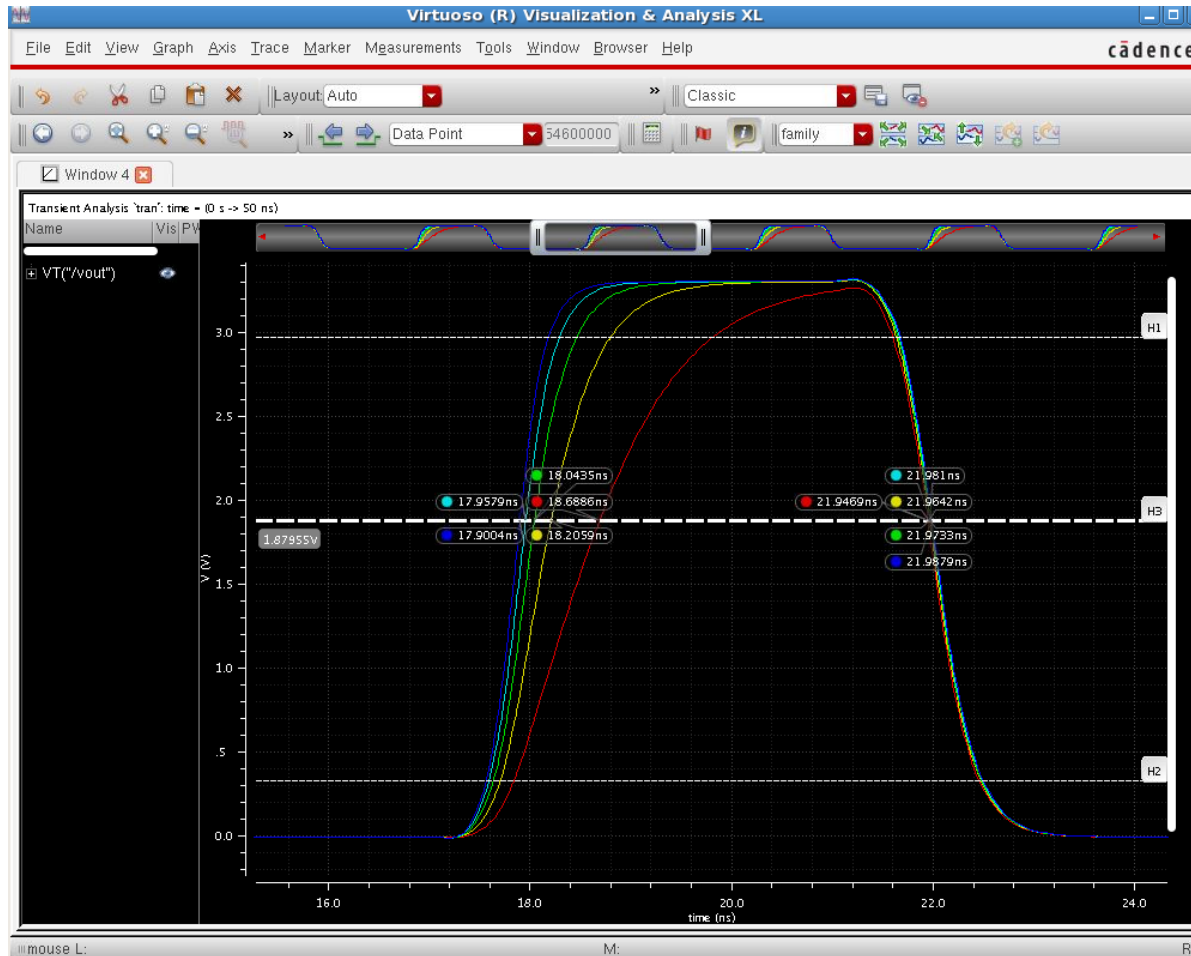
How to measure rise/fall time

- Place a trace marker at 0.33V and 2.97V and read off the times
- Rise time is $19.8-17.8\text{ns}$
 $=2\text{ns}$



How to measure rise/fall time

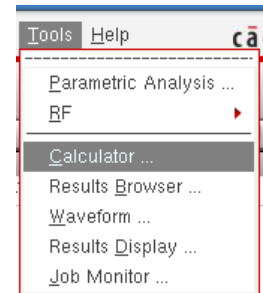
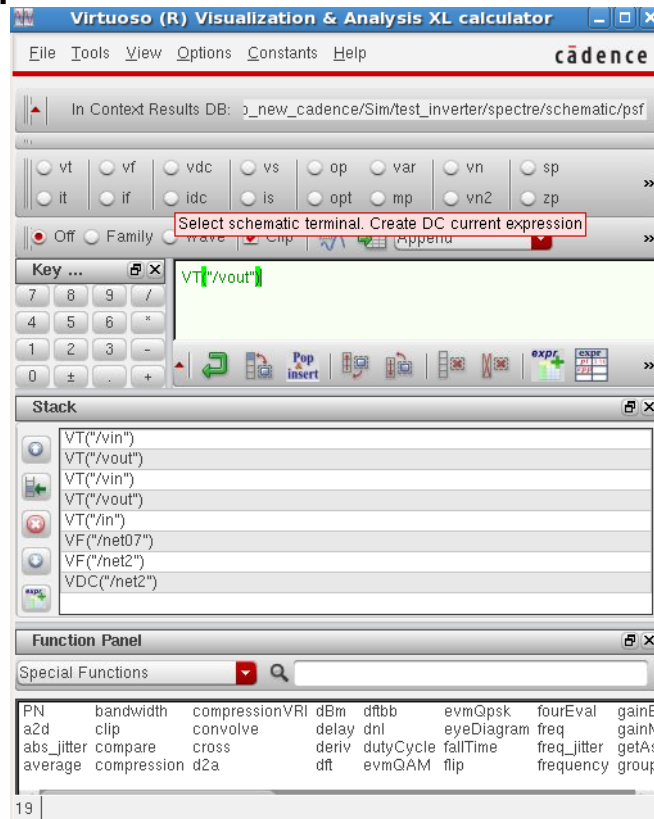
- Use horizontal markets 'H'



How to measure rise/fall time

■ Use the calculator

- Very powerful collection of tools



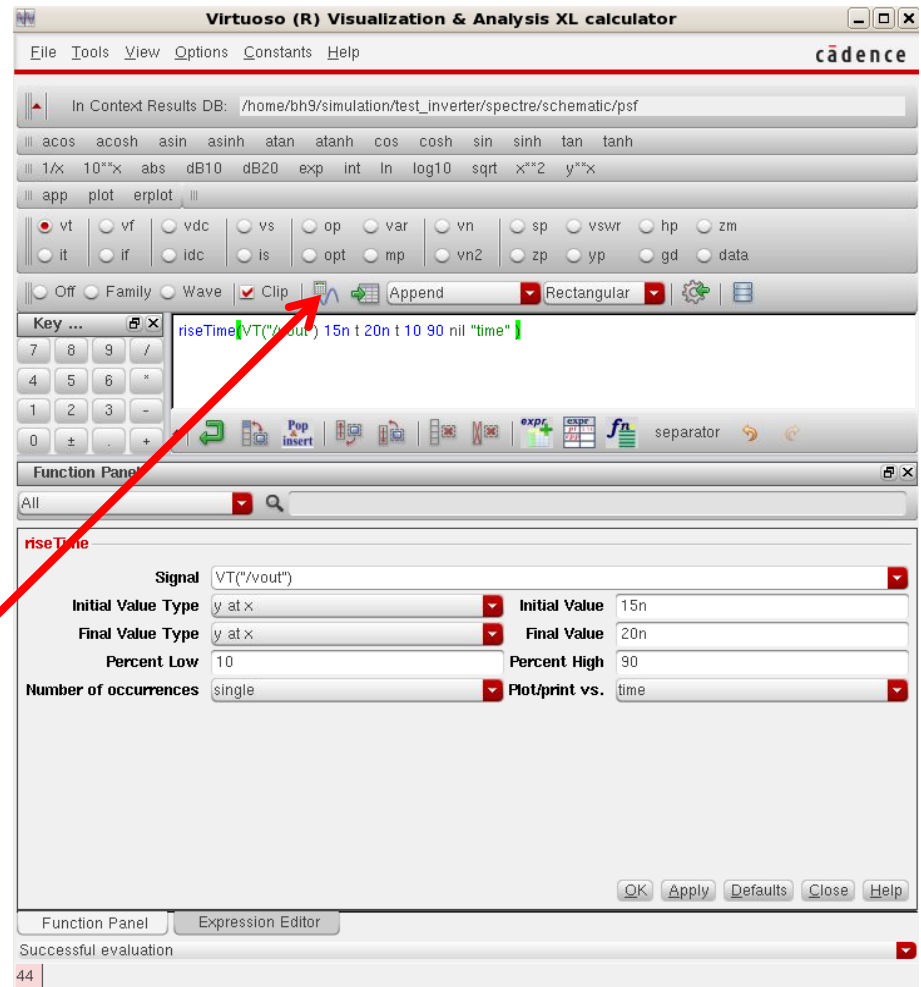
How to measure rise/fall time

- Choose vt
- Choose the net on the schematic window
- Select the riseTime from Function Panel
- Choose 'y at x'
- Set initial and final x values (15ns, 20 ns)
- Press apply
- Press Evaluate

The screenshot shows the Virtuoso (R) Visualization & Analysis XL calculator interface. The main window displays a list of functions, with 'vt' selected. The Function Panel is open, showing the configuration for the 'riseTime' function. The Signal is set to 'VT("/vout")'. The Initial Value Type is 'y at x' with an Initial Value of 15n. The Final Value Type is 'y at x' with a Final Value of 20n. The Percent Low is 10 and the Percent High is 90. The Number of occurrences is set to 'single' and the Plot/print vs. is set to 'time'. The Expression Editor shows the formula: `riseTime|VT("/vout") 15n t 20n t 10 90 nil "time"`. The status bar at the bottom indicates a 'Successful evaluation'.

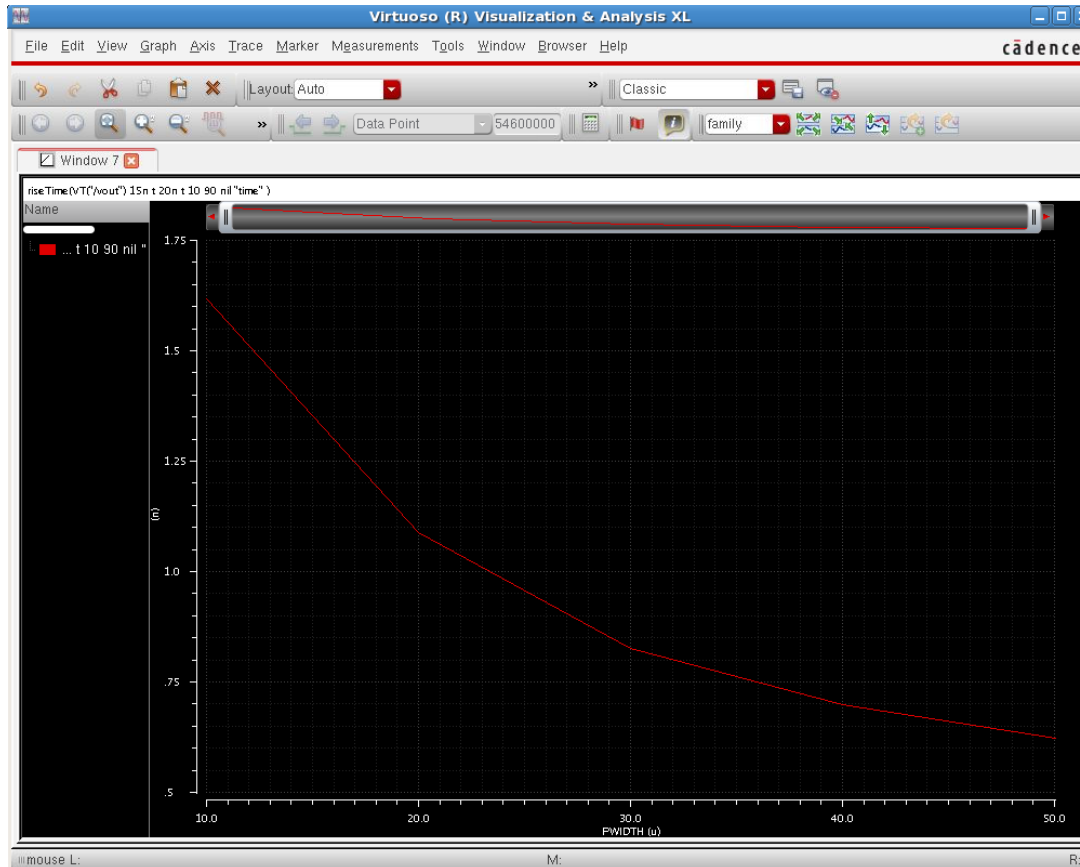
How to measure rise/fall time

- Choose vt
- Choose the net on the schematic window
- Select the riseTime from Function Panel
- Choose 'y at x'
- Set initial and final x values (15ns, 20 ns)
- Press apply
- Press Evaluate



How to measure rise/fall time

- Plots the rise time with different values of the PMOS width



Summary

- In this lecture we created a symbol from a schematic
- We made a test schematic containing the symbol
- We learnt how to simulate the design
- We have seen how to plot and measure the results
- We have shown how to set up hierarchical parameter passing in your design
- We have used design variables and parametric simulations to show rise time variation against PMOS width



Lab: Simulation



Objective: Create a symbol and test circuit. Then simulate your test circuit and use parametric simulations and design variables

