Cadence Inverter Transistor Sizing Tutorial
Cadence Inverter Ocean Introduction
Cadence Inverter Corners Tutorial
Cadence Inverter VerilogA Tutorial

Alfred Sargezi & Zain Ali

AMS Group - San Jose State University
ams.sjsu.edu

Rev . 01 --- September 7, 2012

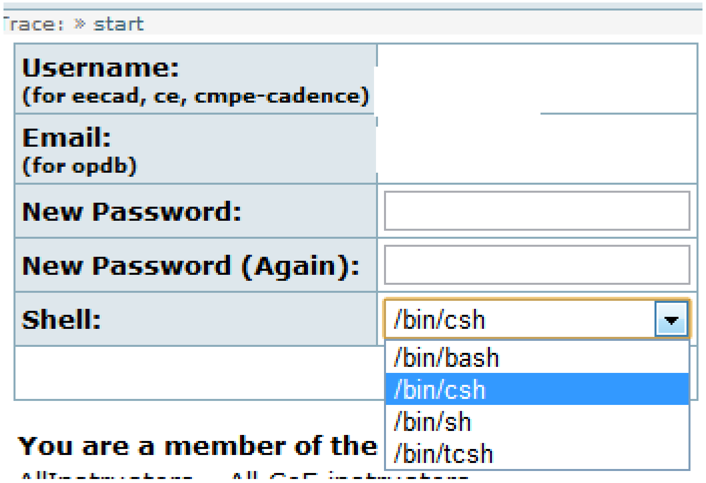
Unix account and Cadence Setup Instructions

1. Set up a Unix account by visiting the following website.

https://unix.engr.sjsu.edu/wiki/doku.php

2. Complete the Cadence Tutorial. This will setup cadence on your account and provide you with a general idea on how to use cadence. **Type "csh" in linux terminal to switch to your directory**.

Alternatively, you can also go to your unix account management online and set it up as csh there.



Trace: » start

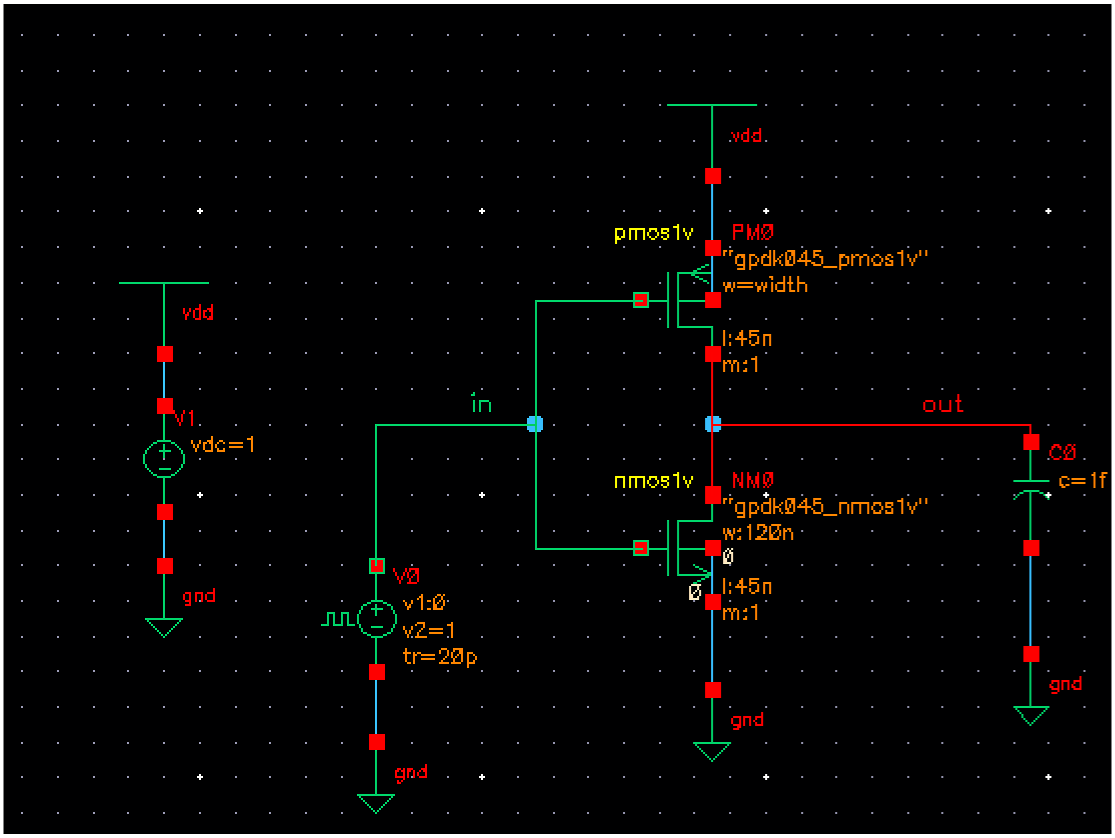
Username: **(for eecad, ce, cmpe-cadence)**	
Email: **(for opdb)**	
New Password:	
New Password (Again):	
Shell:	/bin/csh

You are a member of the

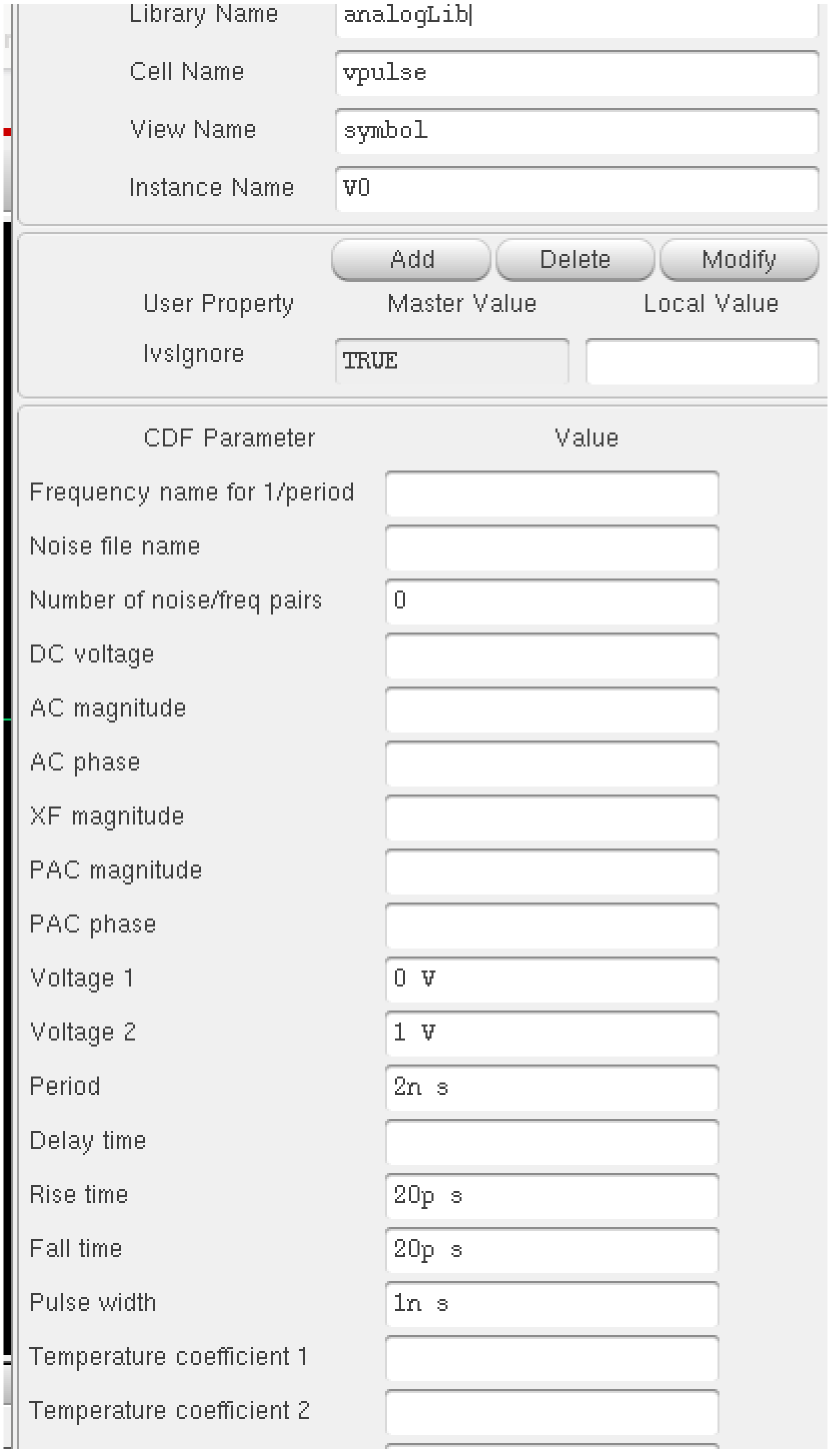
http://www.engr.sjsu.edu/mjones/cadence6.pdf

Testbench

1. Create a Inverter Testbench as follows:



2. Vpulse is created as follows:

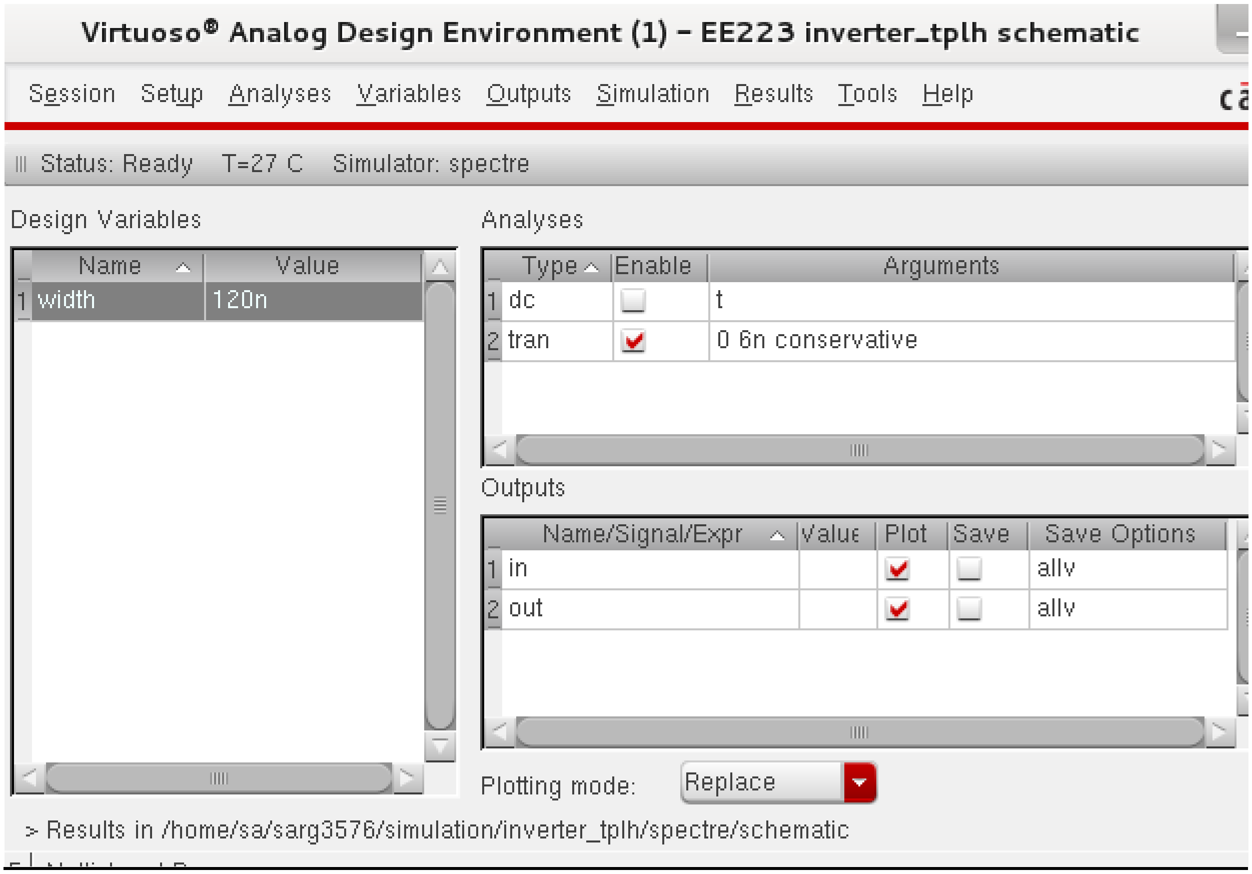


Library Name analogLib
Cell Name vpulse
View Name symbol
Instance Name V0

User Property	Master Value	Local Value
lvsIgnore	TRUE	

CDF Parameter	Value
Frequency name for 1/period	
Noise file name	
Number of noise/freq pairs	0
DC voltage	
AC magnitude	
AC phase	
XF magnitude	
PAC magnitude	
PAC phase	
Voltage 1	0 V
Voltage 2	1 V
Period	2n s
Delay time	
Rise time	20p s
Fall time	20p s
Pulse width	1n s
Temperature coefficient 1	
Temperature coefficient 2	

3. Launch ADE L and set up the following:



Virtuoso® Analog Design Environment (1) – EE223 inverter_tplh schematic

Session Setup Analyses Variables Outputs Simulation Results Tools Help

Status: Ready T=27 C Simulator: spectre

Design Variables

	Name	Value
1	width	120n

Analyses

	Type	Enable	Arguments
1	dc	☐	t
2	tran	☑	0 6n conservative

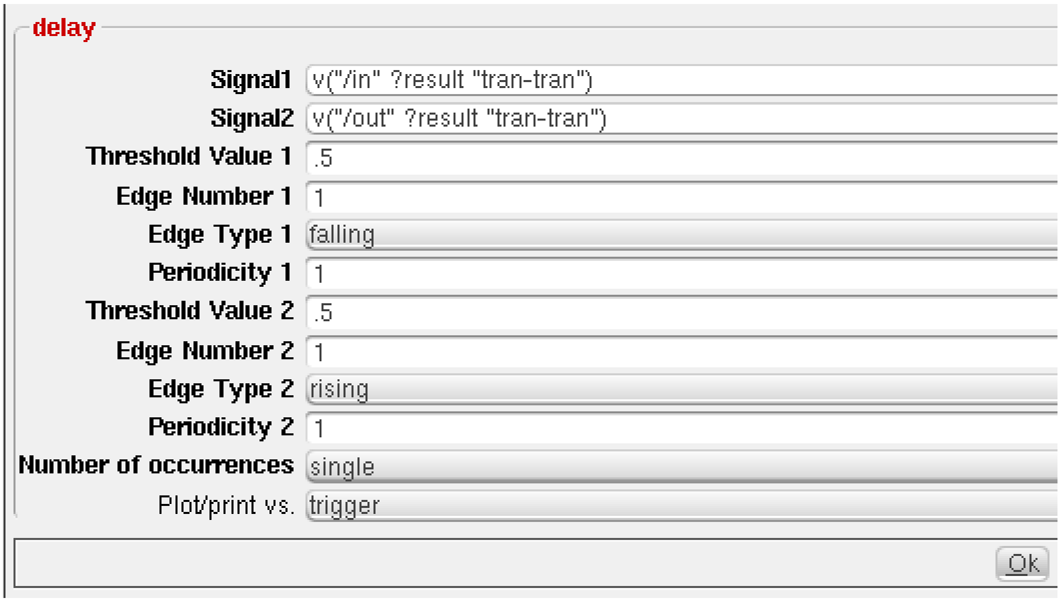
Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	in		☑	☐	allv
2	out		☑	☐	allv

Plotting mode: Replace

> Results in /home/sa/sarg3576/simulation/inverter_tplh/spectre/schematic

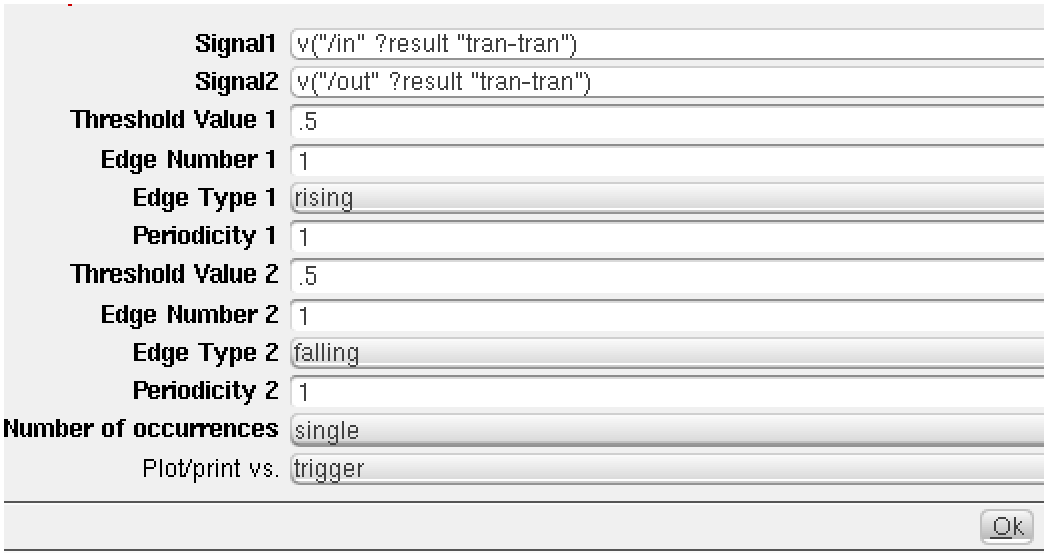
4. Run the ADE L once and launch calculator. Set up the following two test benches for tplh and tphl. We will be using the delay function in calculator. tplh



delay

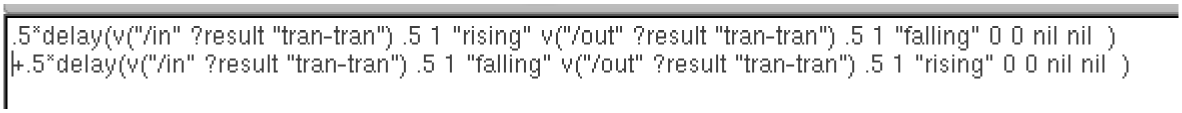
Signal1	v("/in" ?result "tran-tran")
Signal2	v("/out" ?result "tran-tran")
Threshold Value 1	.5
Edge Number 1	1
Edge Type 1	falling
Periodicity 1	1
Threshold Value 2	.5
Edge Number 2	1
Edge Type 2	rising
Periodicity 2	1
Number of occurrences	single
Plot/print vs.	trigger

Switch rising and falling.



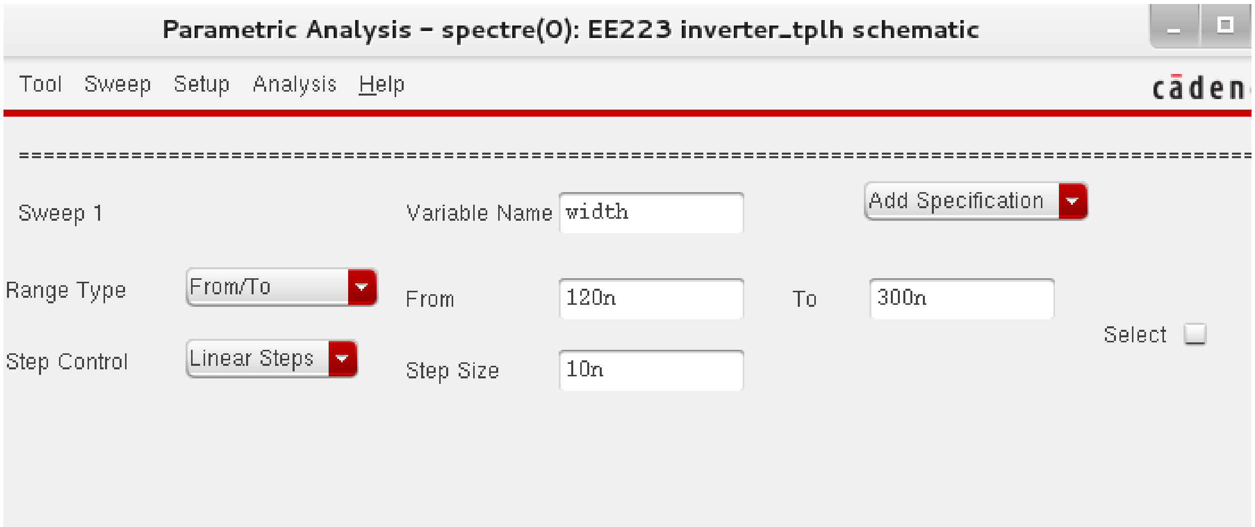
Signal1	v("/in" ?result "tran-tran")
Signal2	v("/out" ?result "tran-tran")
Threshold Value 1	.5
Edge Number 1	1
Edge Type 1	rising
Periodicity 1	1
Threshold Value 2	.5
Edge Number 2	1
Edge Type 2	falling
Periodicity 2	1
Number of occurrences	single
Plot/print vs.	trigger

To find tp, average them both.



```
.5*delay(v("/in" ?result "tran-tran") .5 1 "rising" v("/out" ?result "tran-tran") .5 1 "falling" 0 0 nil nil  )
+.5*delay(v("/in" ?result "tran-tran") .5 1 "falling" v("/out" ?result "tran-tran") .5 1 "rising" 0 0 nil nil  )
```

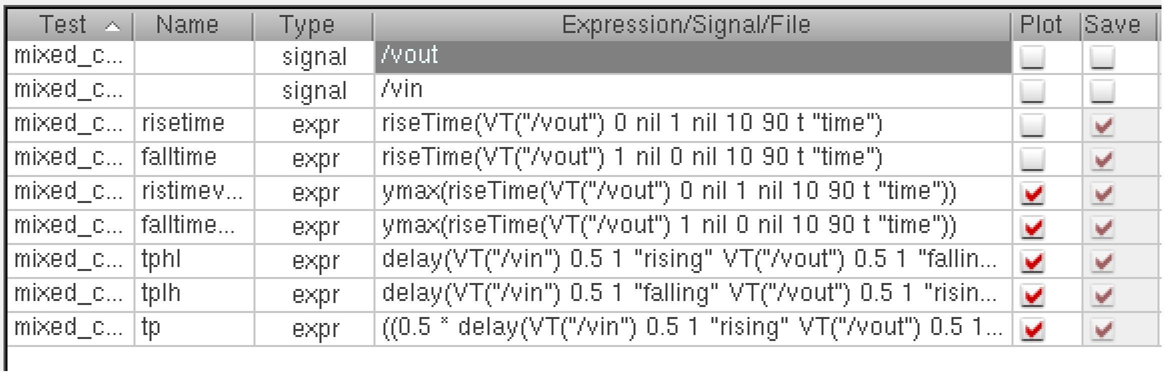
5. Run a parametric analysis sweeping the width. In ADE L , under Tool select Parametric Analysis.



Parametric Analysis – spectre(0): EE223 inverter_tplh schematic

Sweep 1		Variable Name	width		
Range Type	From/To	From	120n	To	300n
Step Control	Linear Steps	Step Size	10n		

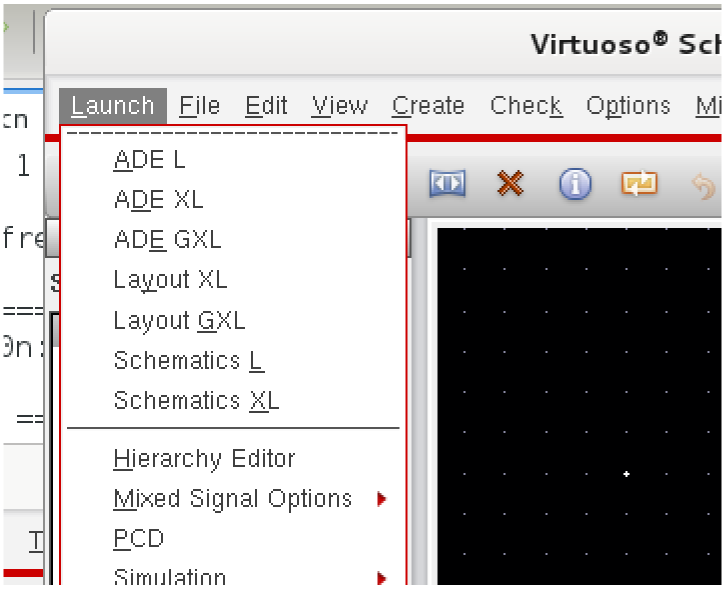
6. After the parametric analysis is complete, go back to calculator and plot the functions to find the optimal size for your transistors.
7. You can also use risetime and falltime to get the correct width value. We have to use the ymax function so we can extract a value since rise time and falltime are continuous functions in cadence. See below.



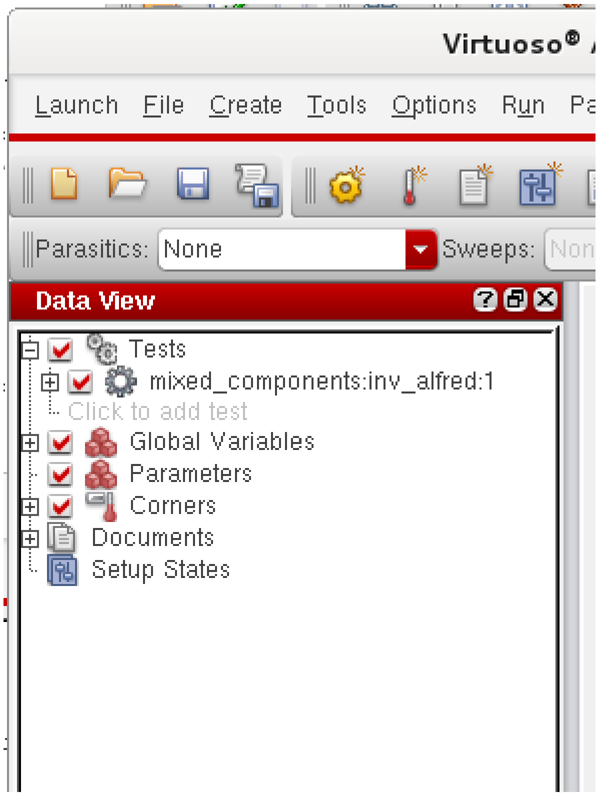
Test	Name	Type	Expression/Signal/File	Plot	Save
mixed_c...		signal	/vout	☐	☐
mixed_c...		signal	/vin	☐	☐
mixed_c...	risetime	expr	riseTime(VT("/vout") 0 nil 1 nil 10 90 t "time")	☐	☑
mixed_c...	falltime	expr	riseTime(VT("/vout") 1 nil 0 nil 10 90 t "time")	☐	☑
mixed_c...	ristimev...	expr	ymax(riseTime(VT("/vout") 0 nil 1 nil 10 90 t "time"))	☑	☑
mixed_c...	falltime...	expr	ymax(riseTime(VT("/vout") 1 nil 0 nil 10 90 t "time"))	☑	☑
mixed_c...	tphl	expr	delay(VT("/vin") 0.5 1 "rising" VT("/vout") 0.5 1 "fallin...	☑	☑
mixed_c...	tplh	expr	delay(VT("/vin") 0.5 1 "falling" VT("/vout") 0.5 1 "risin...	☑	☑
mixed_c...	tp	expr	((0.5 * delay(VT("/vin") 0.5 1 "rising" VT("/vout") 0.5 1...	☑	☑

OCEAN and ADE XL

8. Ocean. Lets say you want to use a different transistor for whatever reason and want to finds its sizing. We can simply save a couple of scripts(inside ADE L and XL) and modify some files to get desired results instead of recreating the whole test bench. The following uses XL.
9. Start by opening up ADE XL and setting up a test bench inside it. Select "Create New View" and press OK.



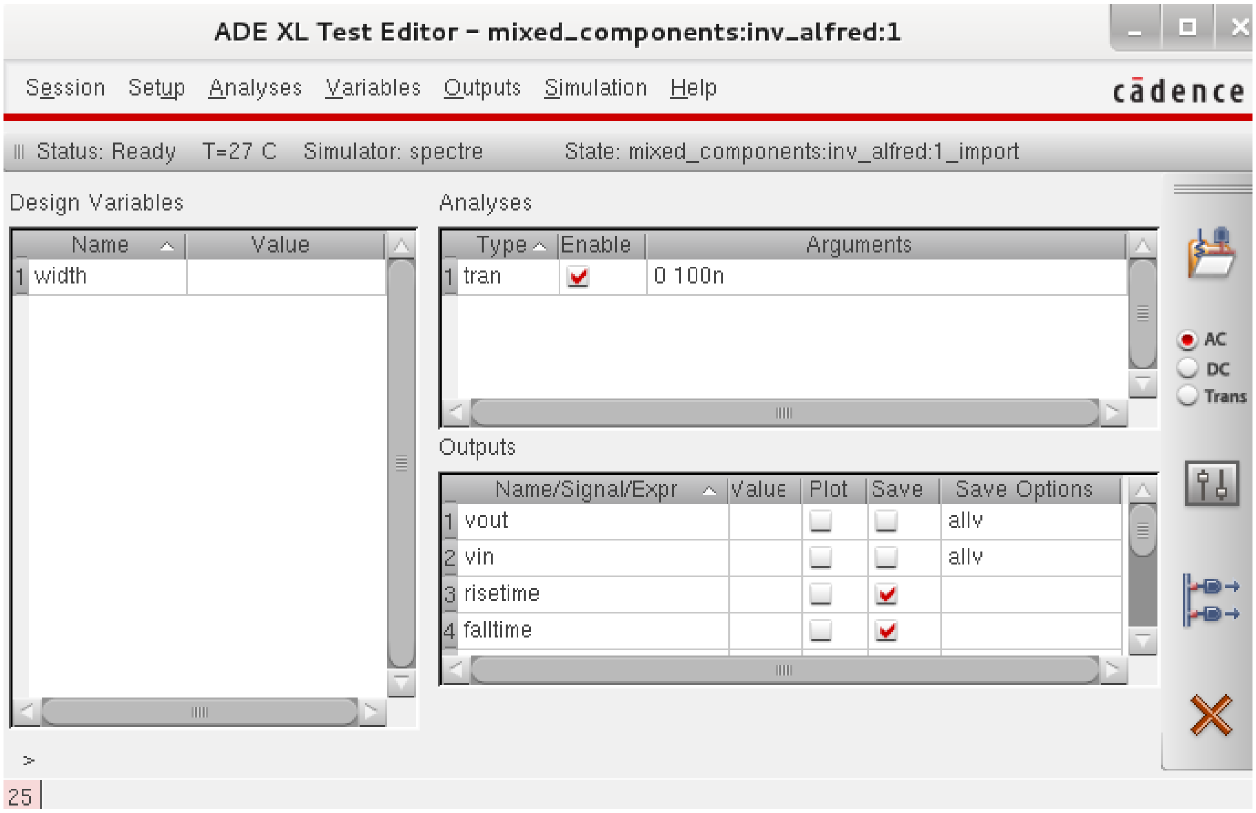
10. Underneath data view you will see a couple of options.



Parasitics: None Sweeps: Non

- ☑ Tests
  - ☑ mixed_components:inv_alfred:1
  - Click to add test
- ☑ Global Variables
- ☑ Parameters
- ☑ Corners
- Documents
- Setup States

11. Click and click to add test and it will open a window similar to ADE L.


ADE XL Test Editor – mixed_components:inv_alfred:1
Session Setup Analyses Variables Outputs Simulation Help
cādence
AC
DC
Trans

Status: Ready T=27 C Simulator: spectre State: mixed_components:inv_alfred:1_import

Design Variables

	Name	Value
1	width	

Analyses

	Type	Enable	Arguments
1	tran	☑	0 100n

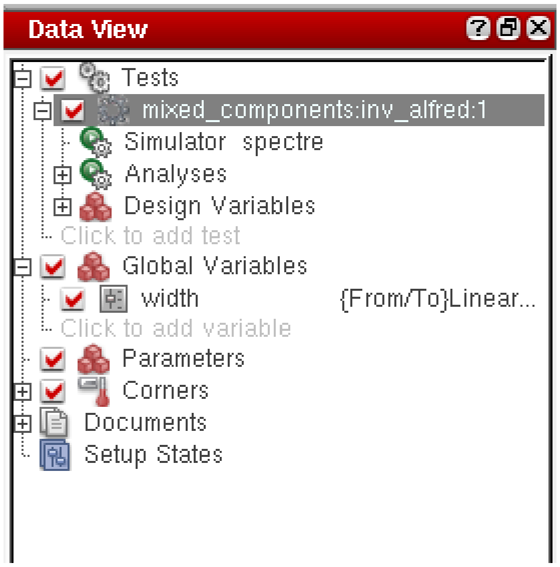
Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	vout		☐	☐	allv
2	vin		☐	☐	allv
3	risetime		☐	☑	
4	falltime		☐	☑	

>

25

12. This is very similar to L. Set up the test bench similar to what you did before. Run a transient analysis and select outputs and rise times, fall times, delays, etc. **Leave width blank.**
13. Under global variables in Data View you can add width. You can either set it up as a single value or run a sweep for certain variables.

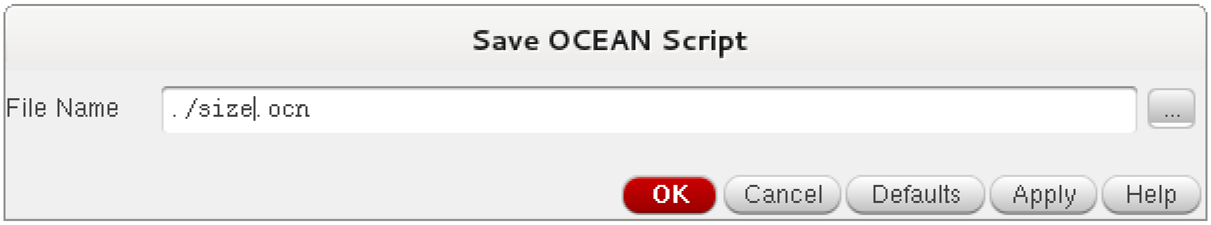


Data View

☑ Tests
☑ mixed_components:inv_alfred:1
Simulator spectre
Analyses
Design Variables
Click to add test
☑ Global Variables
☑ width {From/To}Linear...
Click to add variable
☑ Parameters
☑ Corners
Documents
Setup States

14. ADE XL is much easier for running parametric analysis. Next we will save an ocean script.

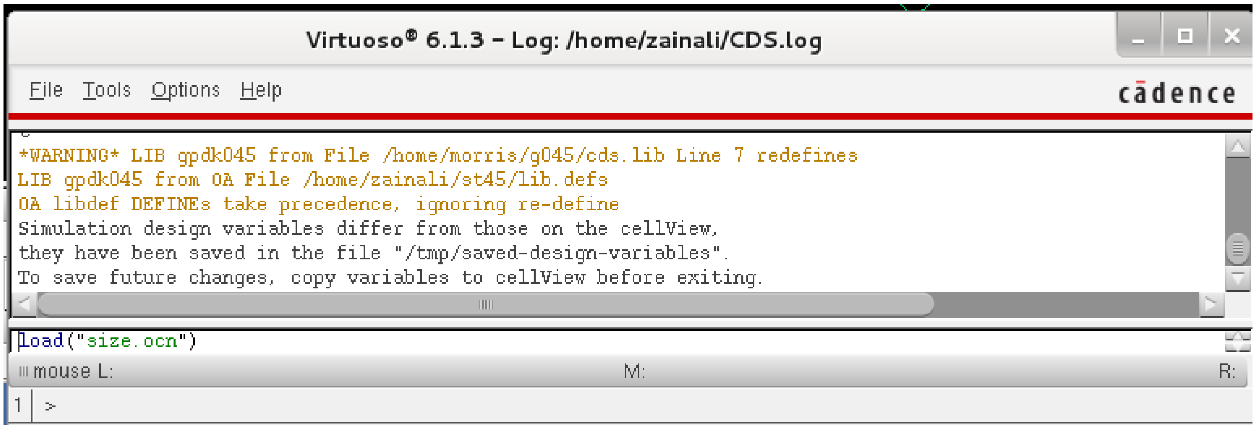
15. You can either click on File→ Save script or Click on the Save script icon ( ) next to the normal button on the toolbar. Next step is to run the ocean script. The default directory will be the st45 folder where you ./s45 script to start virtuoso is located.



Save OCEAN Script

File Name ./size.ocn

16. To run the ocean script simply type load("size.ocn") in the main virtuoso window and hit enter. Be patient, it will take a while to run depending on your sweep size.

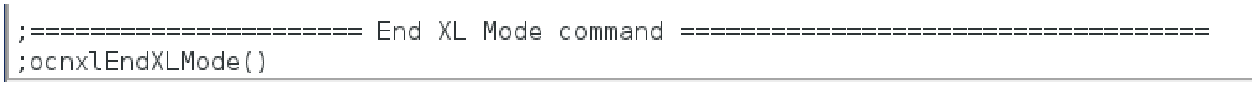


```
*WARNING* LIB gpdk045 from File /home/morris/g045/cds.lib Line 7 redefines
LIB gpdk045 from OA File /home/zainali/st45/lib.defs
OA libdef DEFINEs take precedence, ignoring re-define
Simulation design variables differ from those on the cellView,
they have been saved in the file "/tmp/saved-design-variables".
To save future changes, copy variables to cellView before exiting.
```

```
load("size.ocn")
```

```
1 >
```

17. If the plot window keeps closing go back to your ocean script. At the end there is a “End XL Mode command” section. Just comment ocnxlEndXLMode() by pulling a semicolon in front of it. Rerun the ocean file and it won’t close the plot anymore.

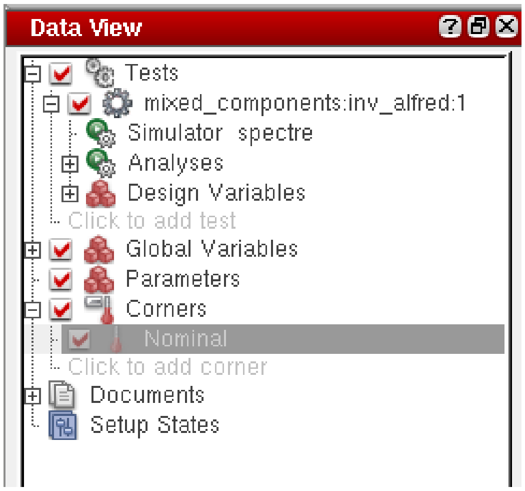


```
;====================== End XL Mode command ===================================
;ocnxlEndXLMode()
```

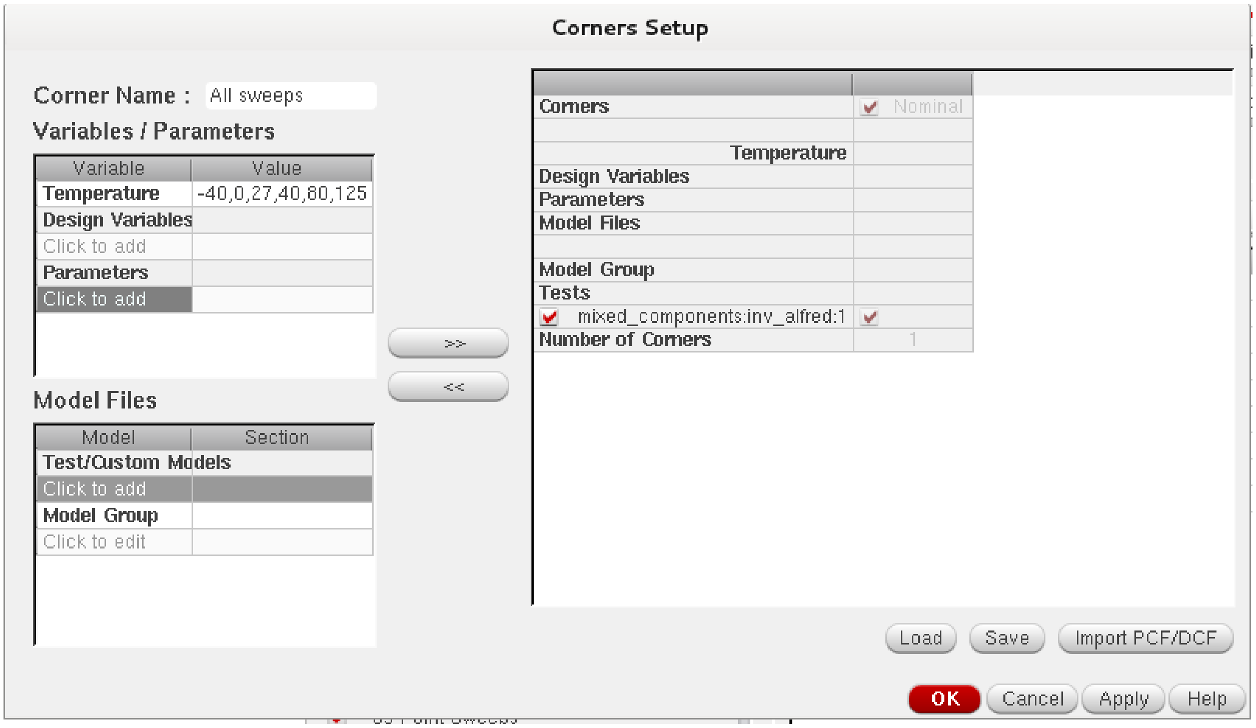
18. Once you have the plot you can figure out what the correct width ratio is.
19. Make a new schematic and name it inv_lowvt. Copy and paste the original components and simply change the transistors with pmos1v_lvt and nmos1v_lvt. Keep everything else the same.
20. Make a copy of size.ocn and name is size_lvt.ocn. Open it with a text editor (such as gedit) and replace all instances of inv with inv_lowvt. Run file size_lvt.ocn and it will plot the data with lowvt transistors. You can repeat the above steps with highvt and whatever else is in your library.
21. **Note: This is just a proof of concept.** If you are more comfortable with the ADE L/XL environment, just use that. This might help people process some information slightly faster. If you are interested in ocean just google a manual and look up functions.

Corners

22. **Corners:** Under data view select "Click to add corner". (If you are using ADE L then look under setup and then model libraries and you can do the same thing).



23. You will get the following window. Temperature is a predefined variable in Virtuoso. You can simply enter values. We are testing from -40 to 125. It's in degrees Celsius. You can also add parameters and design variables and sweep them here.


Corners Setup

Corner Name : All sweeps

Variables / Parameters

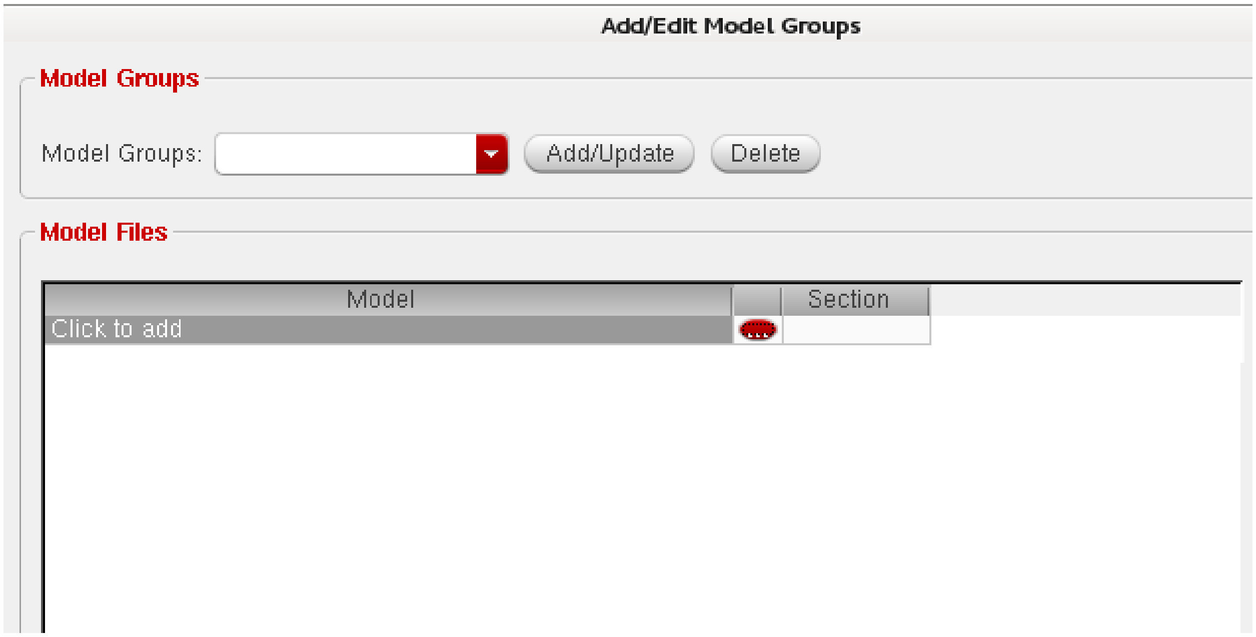
Variable	Value
Temperature	-40,0,27,40,80,125
Design Variables	
Click to add	
Parameters	
Click to add	

Model Files

Model	Section
Test/Custom Models	
Click to add	
Model Group	
Click to edit	

Corners	☑ Nominal
Temperature	
Design Variables	
Parameters	
Model Files	
Model Group	
Tests	
☑ mixed_components:inv_alfred:1	☑
Number of Corners	1

24. Under Model Group Select "Click to edit". You will get the following window. Select "Click to add". It might take a little while to load. Do not Click on the little explore button. Make sure you click on "Click to add".



Add/Edit Model Groups

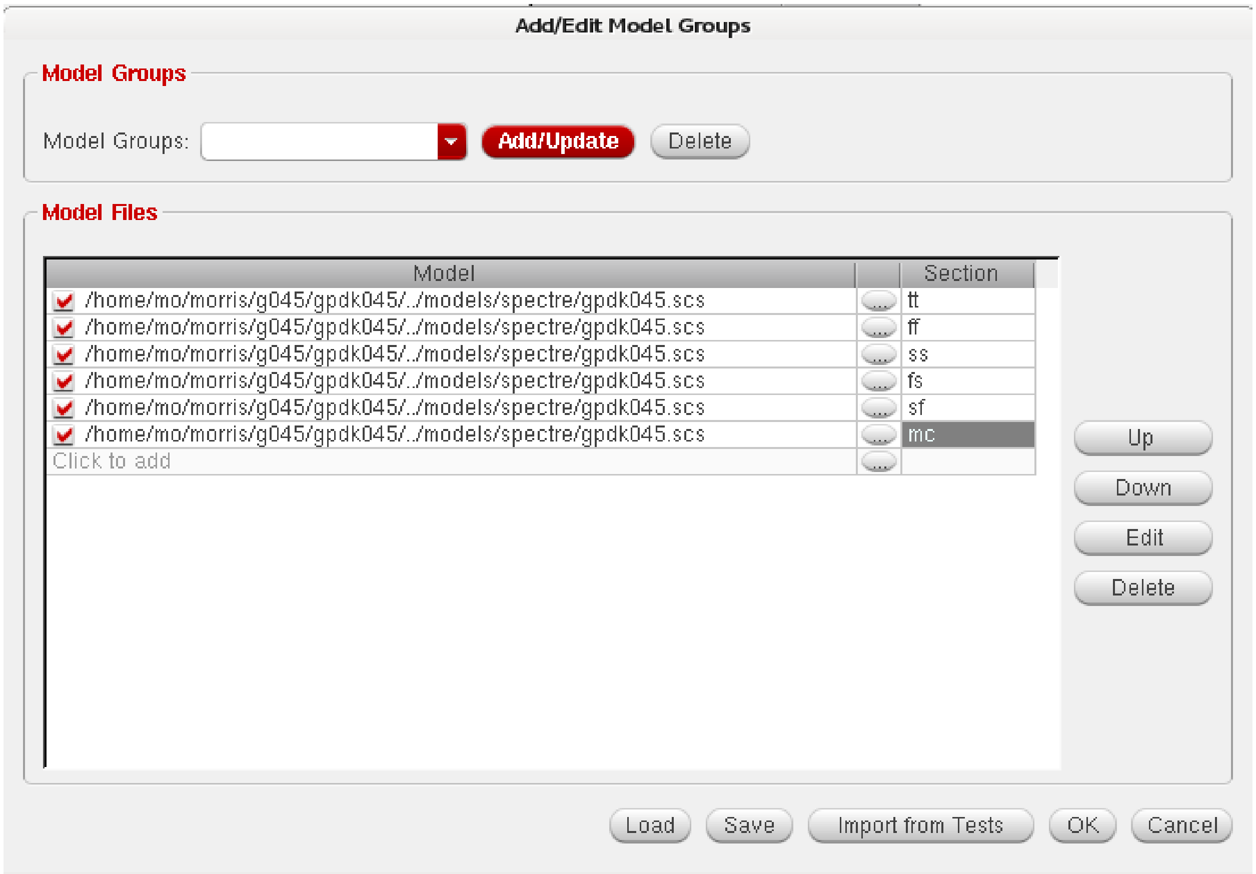
Model Groups

Model Groups:

Model Files

Model		Section
Click to add		

25. Mirror it to the image below. You can just type it in once and copy paste the model path. Make sure red arrow is selected and you should see a drop down menu. Add six of these so you can add all the different corners as shown below. Hit Ok.



Add/Edit Model Groups

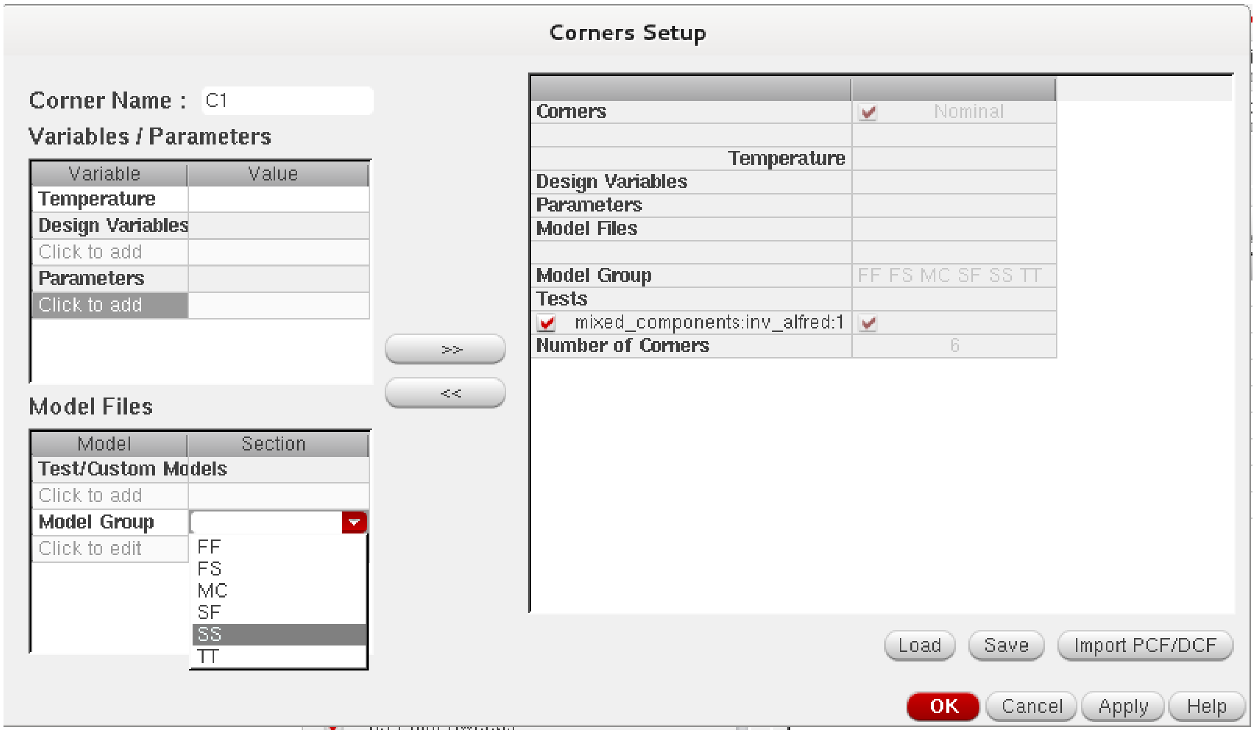
Model Groups

Model Groups:

Model Files

Model		Section
✓ /home/mo/morris/g045/gpdk045/../models/spectre/gpdk045.scs		tt
✓ /home/mo/morris/g045/gpdk045/../models/spectre/gpdk045.scs		ff
✓ /home/mo/morris/g045/gpdk045/../models/spectre/gpdk045.scs		ss
✓ /home/mo/morris/g045/gpdk045/../models/spectre/gpdk045.scs		fs
✓ /home/mo/morris/g045/gpdk045/../models/spectre/gpdk045.scs		sf
✓ /home/mo/morris/g045/gpdk045/../models/spectre/gpdk045.scs		mc
Click to add		

26. Now go back to your Corner setup window. In the Model Group under Section you will see the 6 models if you set it up properly. Add all 6 of them one by one.



Corners Setup

Corner Name : C1

Variables / Parameters

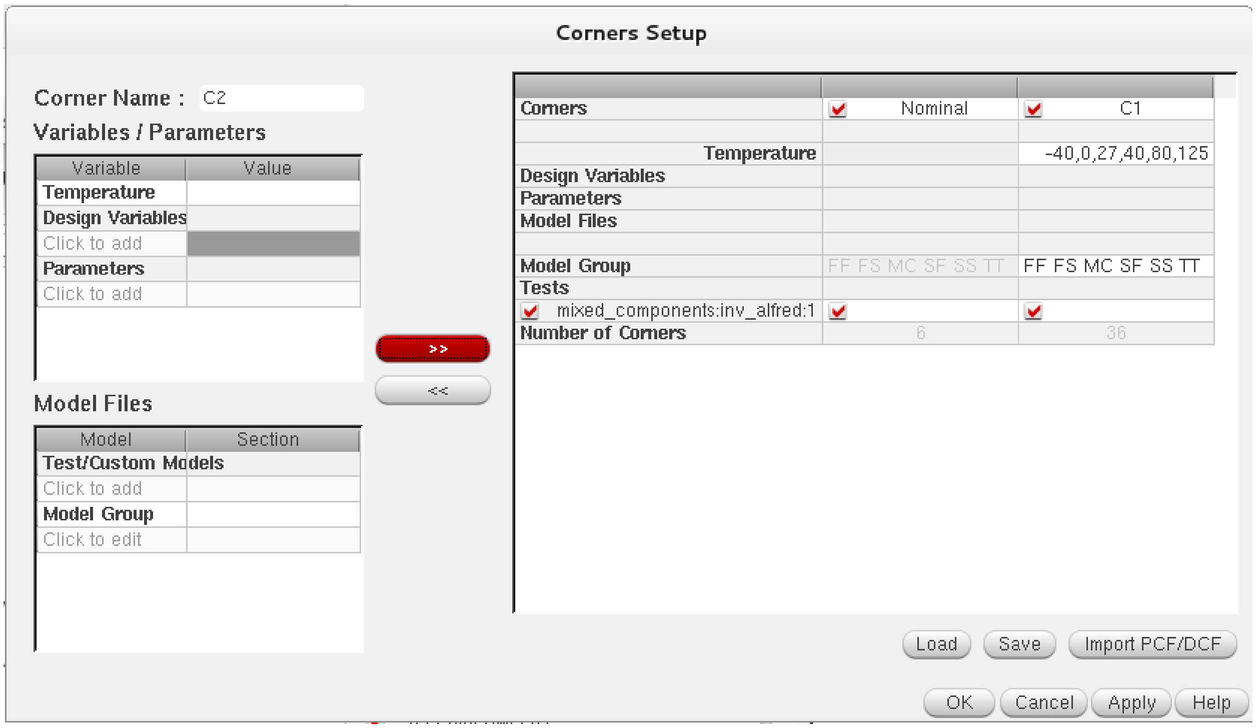
Variable	Value
Temperature	
Design Variables	
Click to add	
Parameters	
Click to add	

Model Files

Model	Section
Test/Custom Models	
Click to add	
Model Group	
Click to edit	

Corners	✓ Nominal
Temperature	
Design Variables	
Parameters	
Model Files	
Model Group	FF FS MC SF SS TT
Tests	
✓ mixed_components:inv_alfred:1	✓
Number of Corners	6

27. Now that temperature and Models are added simply hit the Add/Update Corner button(it's the red one below) and you should get something similar to this.



Corners Setup

Corner Name : C2

Variables / Parameters

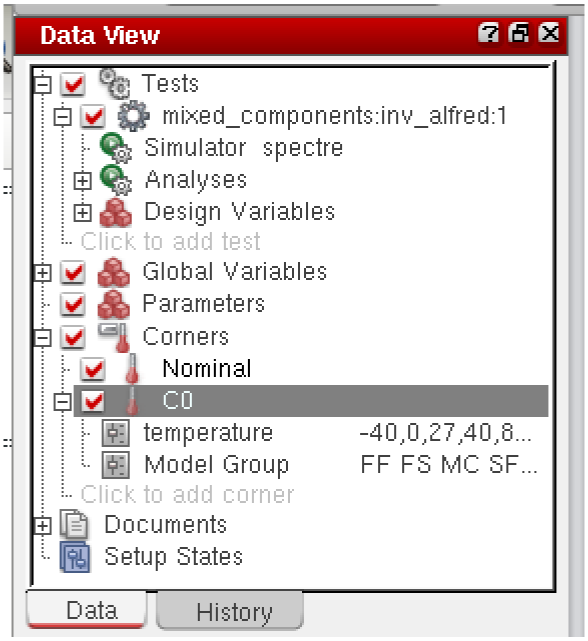
Variable	Value
Temperature	
Design Variables	
Click to add	
Parameters	
Click to add	

Model Files

Model	Section
Test/Custom Models	
Click to add	
Model Group	
Click to edit	

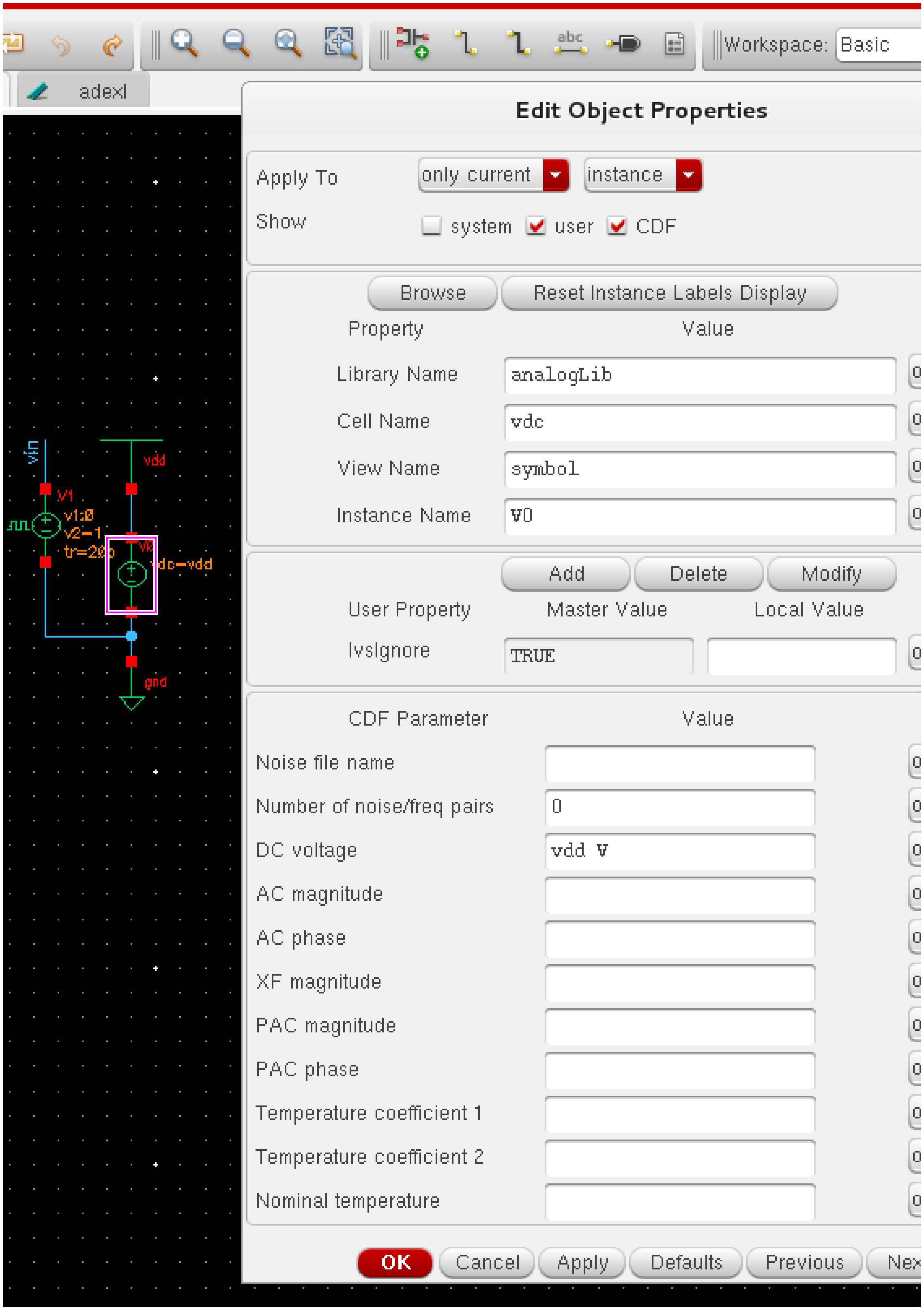
Corners	✓ Nominal	✓ C1
Temperature		-40,0,27,40,80,125
Design Variables		
Parameters		
Model Files		
Model Group	FF FS MC SF SS TT	FF FS MC SF SS TT
Tests		
✓ mixed_components:inv_alfred:1	✓	✓
Number of Corners	6	36

28. Save your sweep file using the Save button. Next time you need to run a corner analysis, simply reload the file and you can skip all of the above steps.
29. Go back to data view and make sure corner is selected.



```
☑ Tests
  ☑ mixed_components:inv_alfred:1
      Simulator  spectre
      Analyses
      Design Variables
    Click to add test
☑ Global Variables
☑ Parameters
☑ Corners
    ☑ Nominal
    ☑ C0
        temperature      -40,0,27,40,8...
        Model Group      FF FS MC SF...
    Click to add corner
  Documents
  Setup States
```

30. Make sure you only have one value for width. Rerun the simulation. If you sweep width again it will simply run 36*(number of widths) that you sweep. It will give you your rise and fall times for various corners.
31. Change the main power supply in the schematic and just call it vdd instead of 1V.



Workspace: Basic

Edit Object Properties

Apply To: only current, instance

Show: ☐ system ☑ user ☑ CDF

Browse | Reset Instance Labels Display

Property	Value
Library Name	analogLib
Cell Name	vdc
View Name	symbol
Instance Name	V0

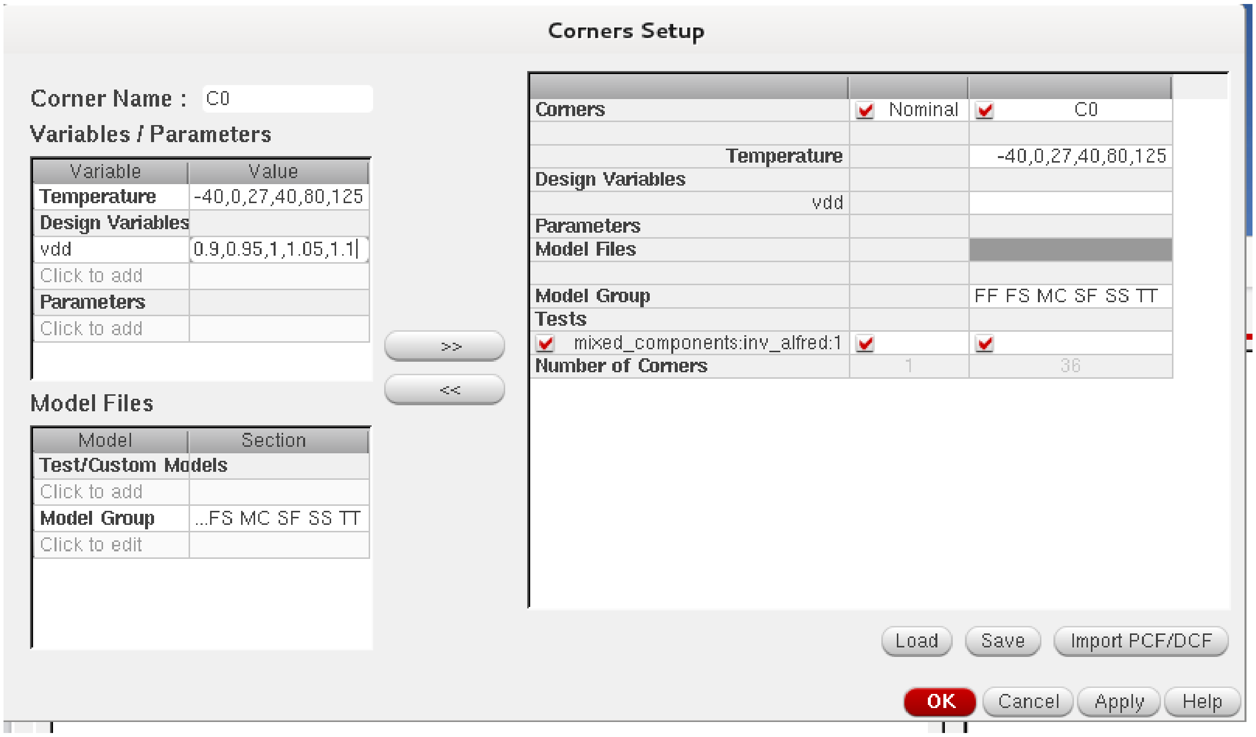
Add | Delete | Modify

User Property	Master Value	Local Value
lvsIgnore	TRUE	

CDF Parameter	Value
Noise file name	
Number of noise/freq pairs	0
DC voltage	vdd V
AC magnitude	
AC phase	
XF magnitude	
PAC magnitude	
PAC phase	
Temperature coefficient 1	
Temperature coefficient 2	
Nominal temperature	

OK | Cancel | Apply | Defaults | Previous | Nex

32. In Corners, under design variables , now select vdd and enter values 0.9, 1, 1.1.



Corners Setup

Corner Name : C0

Variables / Parameters

Variable	Value
Temperature	-40,0,27,40,80,125
Design Variables	
vdd	0.9,0.95,1,1.05,1.1
Click to add	
Parameters	
Click to add	

Model Files

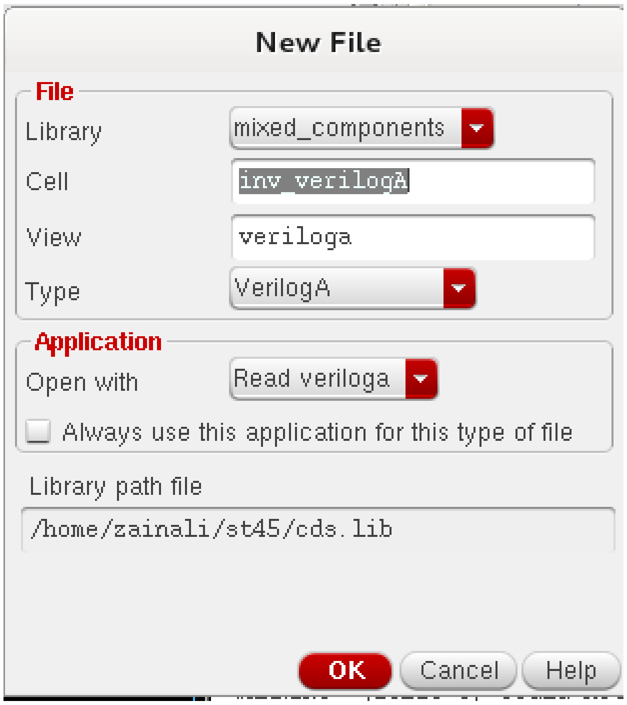
Model	Section
Test/Custom Models	
Click to add	
Model Group	...FS MC SF SS TT
Click to edit	

Corners	☑ Nominal	☑ C0
Temperature		-40,0,27,40,80,125
Design Variables		
vdd		
Parameters		
Model Files		
Model Group		FF FS MC SF SS TT
Tests		
☑ mixed_components:inv_alfred:1	☑	☑
Number of Corners	1	36

33. Re-run the simulation. Now you will have your voltage corners too. You can vary the numbers as required. Just keep it mind, each variation adds 36 tests already in there. Modify the test bench for faster results.

Verilog - A

34. In the main virtuoso window type in editor = “gedit” and press enter. This will change the editor to gedit. (If you really like to use vi, you can skip this step. If you don’t know what vi is, then just change your editor to gedit)
35. Create a new file. Use the following template. Save it in the same library as your inverter. We are basically making a verilogA file. It will open with gedit (if you set it up correctly). **Name the file inv_verilogA . If you name it the same as your inverter you will not be able to create a symbol.**



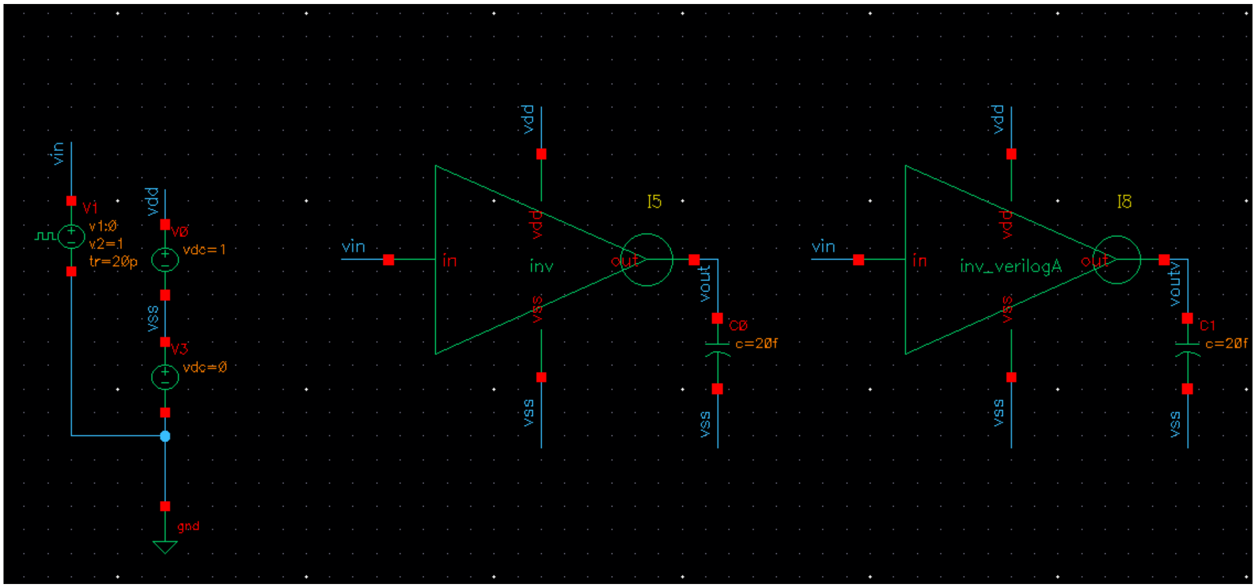
Library: mixed_components
Cell: inv_verilogA
View: veriloga
Type: VerilogA
Open with: Read veriloga
☐ Always use this application for this type of file
Library path file: /home/zainali/st45/cds.lib

36. Write your Verilog code. If you did it properly, it will ask you to create a symbol. Else if will give you syntax errors. Sample code is below.

```
// VerilogA for mixed_components, inv_verilogA, veriloga

`include "constants.vams"
`include "disciplines.vams"

module inv_verilogA(in,out, vss, vdd);

inout vdd, vss;
input in;
output out;
```

```
electrical in, out, vss, vdd;

 analog begin
         if(V(in)>((V(vdd)-V(vss))/2))
         V(out)<+ V(vss);
         else
         V(out)<+ V(vdd);
 end

endmodule
```

37. Put everything in the same inverter test bench.



38. In ADE , plot the vin, vout and voutv. You can see the difference between an ideal model (verilogA) and actual transistor level implementation.
39. VerilogA becomes more useful when you are trying to implement complex systems. You might need to model it in varying ways depending on your output requirement.