

## Lab 3 Additional functions in Cadence

Due at the beginning of class on. Please submit electronically in CANVAS.

Please contact Cindy Liu with questions: xindil@uw.edu

### Introduction

Cadence is a powerful simulation software. So far, we have only utilized very limited functionality of this software, namely the dc and transient simulation in ADE L, layout, DRC and LVS, and parasite extract.

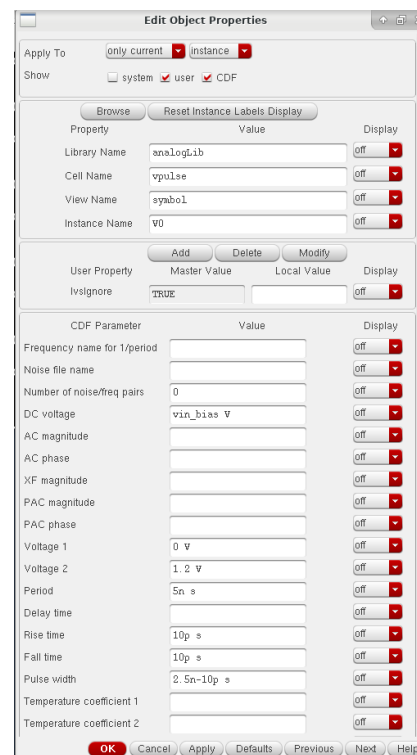
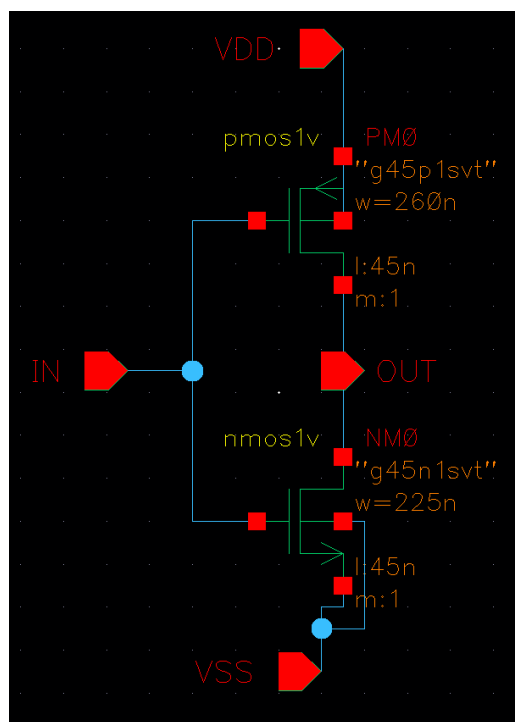
Based on the inverter schematic, we will explore more functions in Cadence in this lab.

1. Calculator
2. Multiparameter sweep

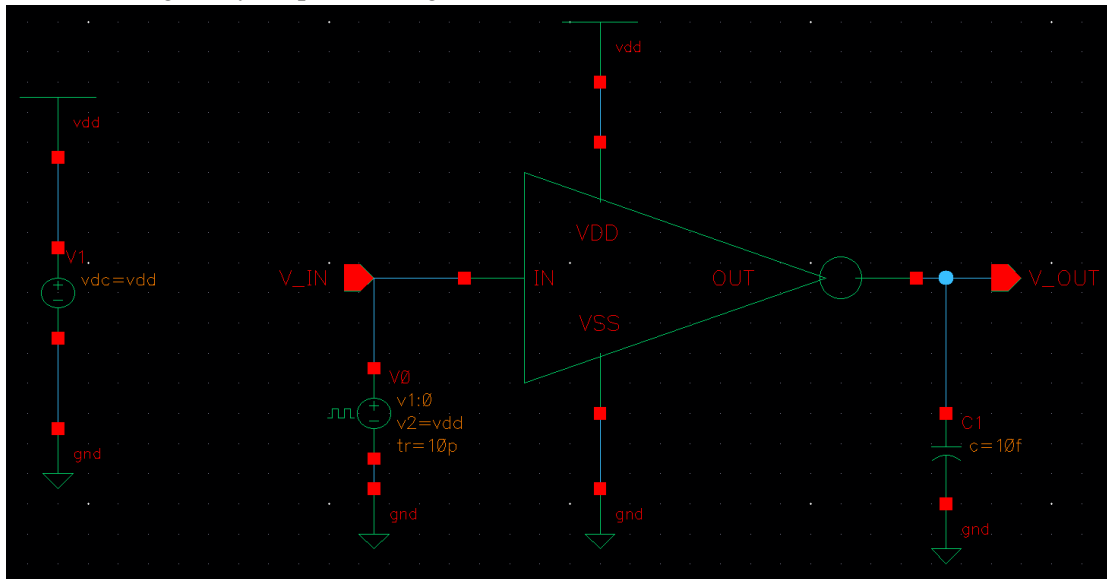
### Calculator <sup>[1]</sup>

The calculator is one of the most used features in Cadence since it's common that people need to extract some parameters from the simulation results, such as rise time, fall time, average, threshold values and so forth. In this section, we will use the calculator to extract rise and fall time of an inverter.

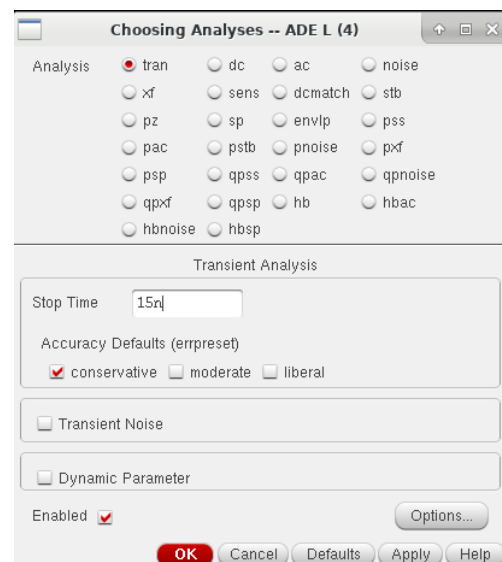
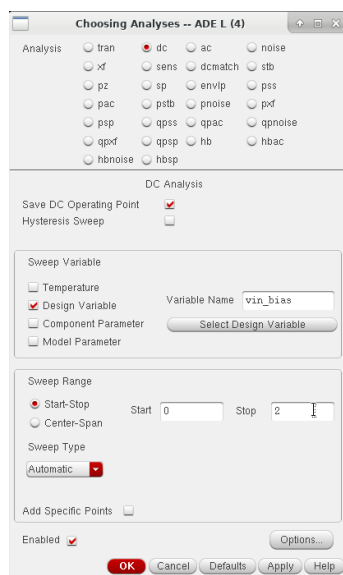
Make a new library for lab3 and attach, and when prompted click “**attach to an existing technology library**”, and attach it to “**gpdtk045**”. Similar to how you did in the previous lab, make an inverter, but this time, use the “**nmos1v**” and “**pmos1v**” devices. Use widths of 225 nm and 260 nm, respectively. Create input and output pins as follows:



Again, make a “triangle and dot” symbol for the inverter (same as you did in last week’s lab). To test the inverter, create a new cellview and design a circuit with a **pulse voltage** (“vpulse”) source input and a **10f capacitor** at the output (using the analog lib library). (You can name this cellview whatever you want; something like “INVD1\_timing” might be logical). Use the above settings for your pulse voltage source and use a **Vdd of 1.0V**.

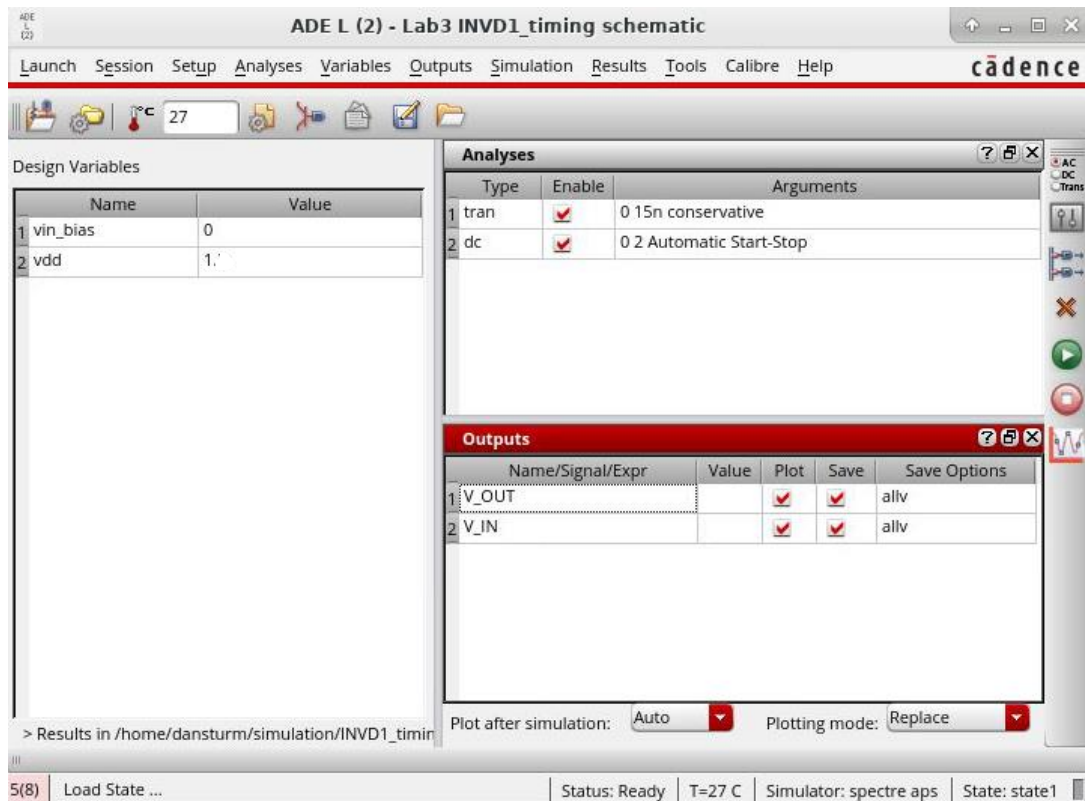


Open ADE. Right click on the blank in **Design Variables** -> **copy from cellview**, set the value of **vin\_bias** as 0 V. Choose “**Analyses -> Choose**”, choose **trans** and set it as below (left). Again, right click on the blank in Analyses, choose **dc** and set it as below (right).

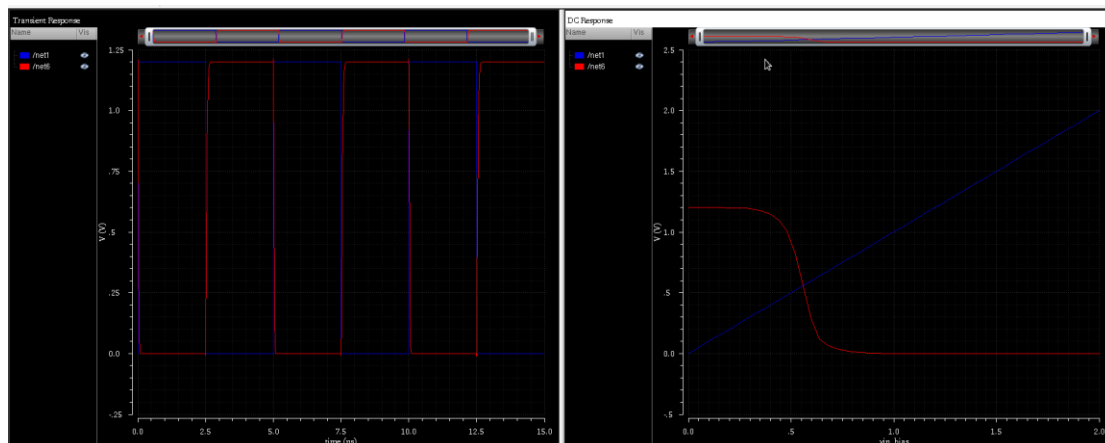


Right click on the blank in **Outputs** -> **Choose from Schematic**, choose the input and the output of inverter.

After you finish all the setup, the window of ADE L should look similar to the following (it might be different if you set Vdd directly in the voltage source).

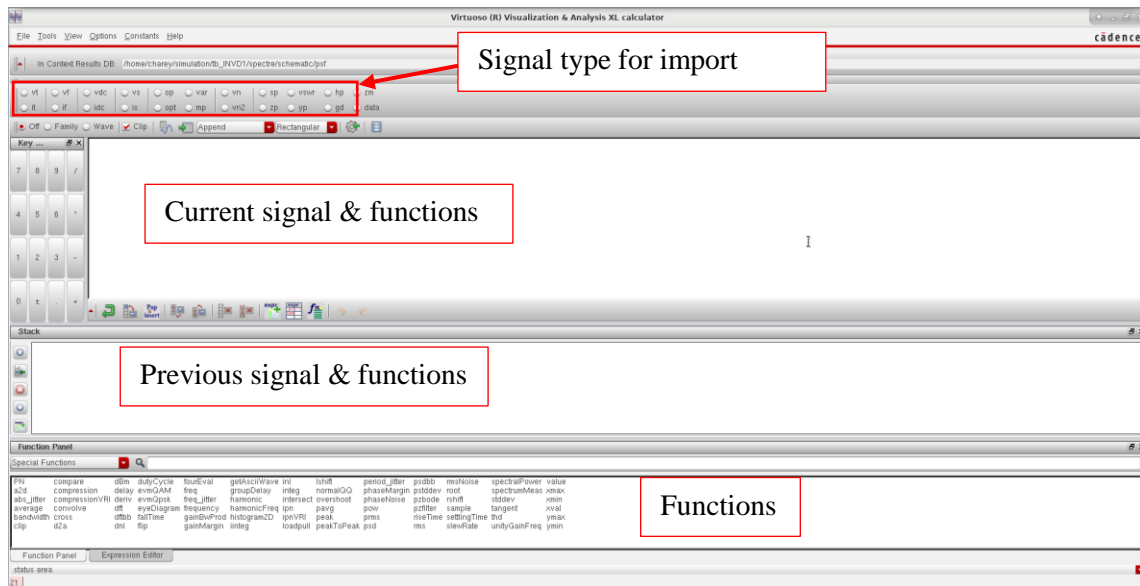


Remember to check the Save option in outputs. You can save your simulation setup by **Session -> Save state -> cellview -> some new name (do not overwrite the previous one!)**. Click on the run button, you should get the following plots:

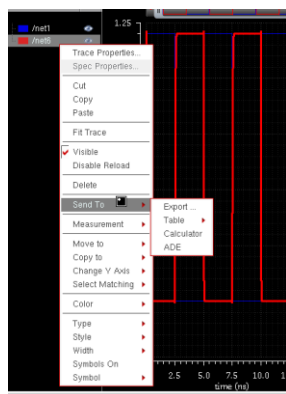
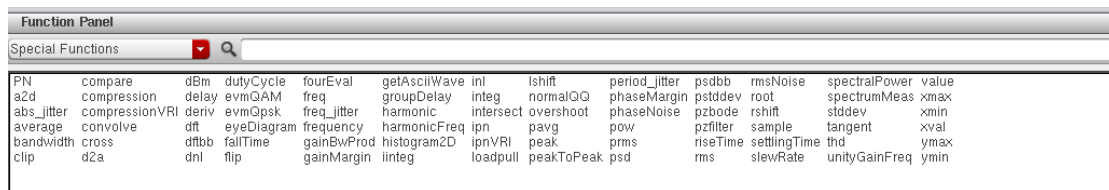


Now, from the transient characteristic, we want to extract the rise time and drop time for this inverter, how should we do that? Of course, you can zoom in the plot and get the value, but virtuoso provides much cleverer solutions. That is **calculator**!

On the plot window, click **Tools -> Calculator**, the calculator window would pop up:



You can also go back the **ADE L -> Tools -> Calculator** to open the window. The meaning of each sections in calculator is shown in the picture above. Note that the lower part of the calculator is the functions provided by **Cadence**, we will use the *riseTime* and *fallTime* functions. You can type the function name in the blank and search for functions.



How to use it though? There are multiple ways to import data into the calculator so that we can use the functions to process the data.

Go back to the plot window **and right click on the VOUT line -> Send to -> Calculator**, the data is then imported into the calculator OUT, right.

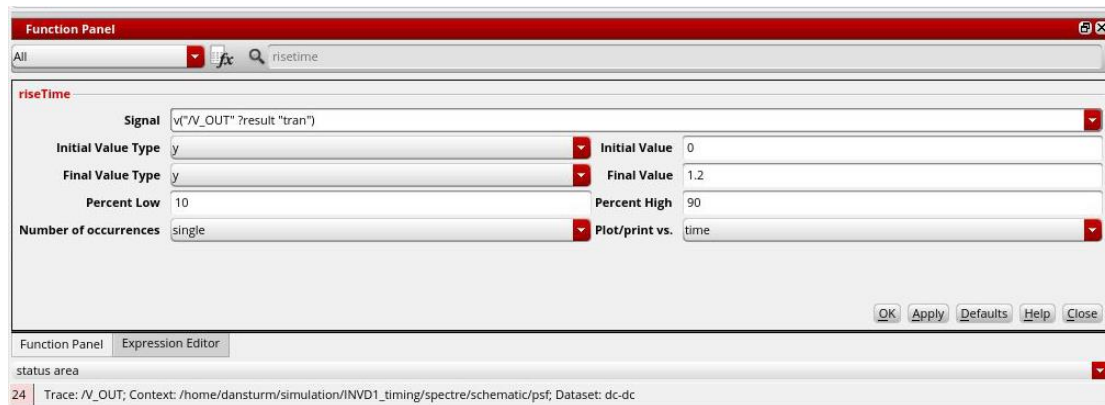
Another way to import the data is select **wave** in the calculator and go back to the plot and choose the line you want to import.




Go the function panel at the bottom and find the **risetime** function. In the “signal” section you may have to select “**buffer**” from the drop down menu to bring up text that represents the “**V\_OUT**” signal (or whatever your inverter output is called). In the risetime function, “initial value” refers to the lowest value the signal will hit, and “final value” refers to the highest value the signal will hit. “Percent low” and “percent high” refer to the percentages of the way

between initial and final value that we want to measure the signal timing for. That is, the calculator will give us the time that the signal takes to rise from *initial value \* percent low* to *final value \* percent high*. (One reason it can be important to have percent low and percent high instead of measuring rise time all the way from initial value to final value is that the value may get very close to the initial/final value without ever reaching it, in which case a rise time measurement all the way from initial to final value would not carry much meaning).

Enter the following values for initial/final value and percent low/high into the calculator.



After setting up the calculator, you can click the button evaluate , and the result will pop

up: 

Expression	Value
1 riseTime(v(\"/n...	54.05E-12

, which means the rise time is 54.05 ps (your value will likely be different).

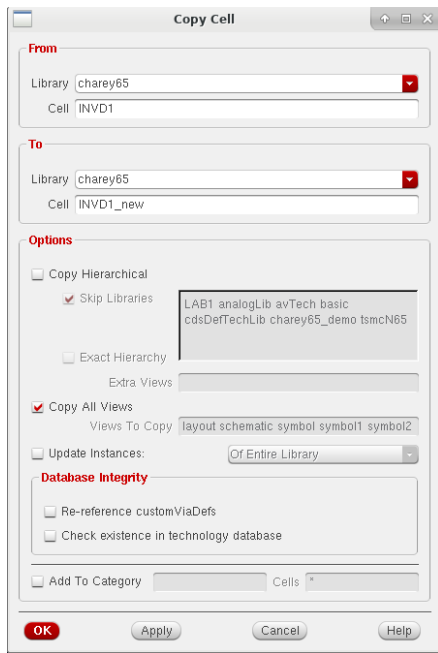
Cadence enables you to add the calculation result to your **ADE L**, so that you can get the rise time after running your simulation. To do that, go back to your ADE L window and right click on the blank of **Outputs -> Edit -> Get expression**, you can see an expression added to the expression blank. After, name it as “risetime”, click on apply and it has been added to your outputs list. Next time you run the simulation; the simulator will calculate the rise time automatically!

You can do the similar work if you want to try *falltime* function, make sure that you put in the right initial values (note, initial and final will switch).

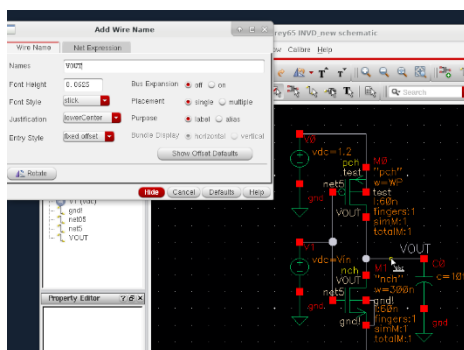
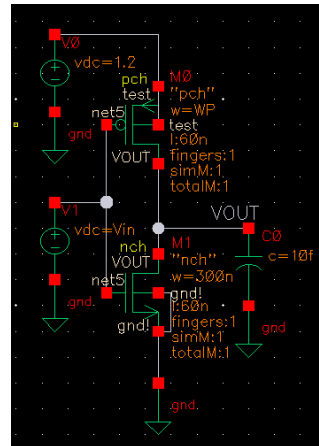
## Multiparameter sweep

So far, we have been dealing with the functions in ADE L. As I mentioned before, there are other options including **ADE XL** and ADE GL. We will introduce the ADE XL and use it do multiparameter sweep.

ADE XL is centered around the concept of **tests**. Think tests as the simulation taking into consideration about the actual fabrication errors or variations. <sup>[2]</sup> In this section, you will learn how to run a parameter sweep in **ADE XL** rather than using the **parametric analyses** as we did in the Lab 1. The intension of this is introduce you to **ADE XL**, so that you can further explore it when you need to do corner testing and Monte Carlo simulation. Let us get started!

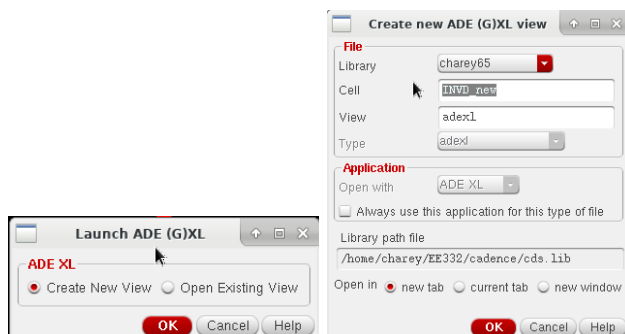


We first need to build up a simple testbench schematic. Copy the cell **INV1** to **INV1\_new**, right click on **INV1** -> **Copy** -> **To INV1\_new**. We are going to change the schematic into a testbench schematic. (Note, your circuit schematic might look a little different from the one below). Alternatively, after copying the inverter, you can leave it as is, and create a separate cellview with voltage sources and capacitor, and include the symbol of the inverter.

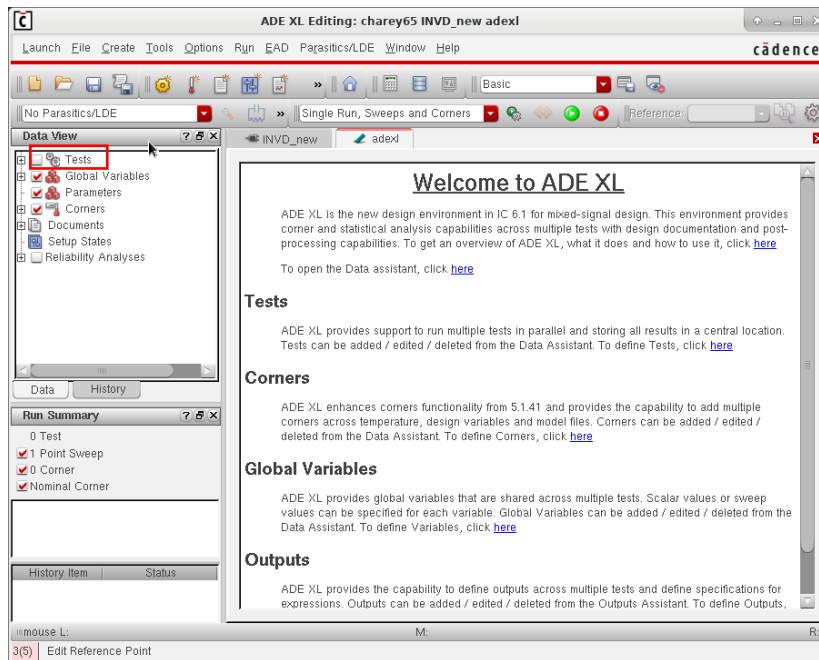


Set the DC Voltage of **V0** as 1.0V. Set the capacitance at the output end as 10fF. Set the DC Voltage of **V1** to variable “**Vin**”, set the width of PMOS as variable “**WP**”. You can name a net by pressing “**T**” key, which means label, as shown below, so that when you select the net next time, it will have a meaningful name rather than “net03”.

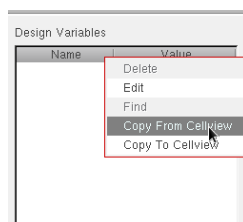
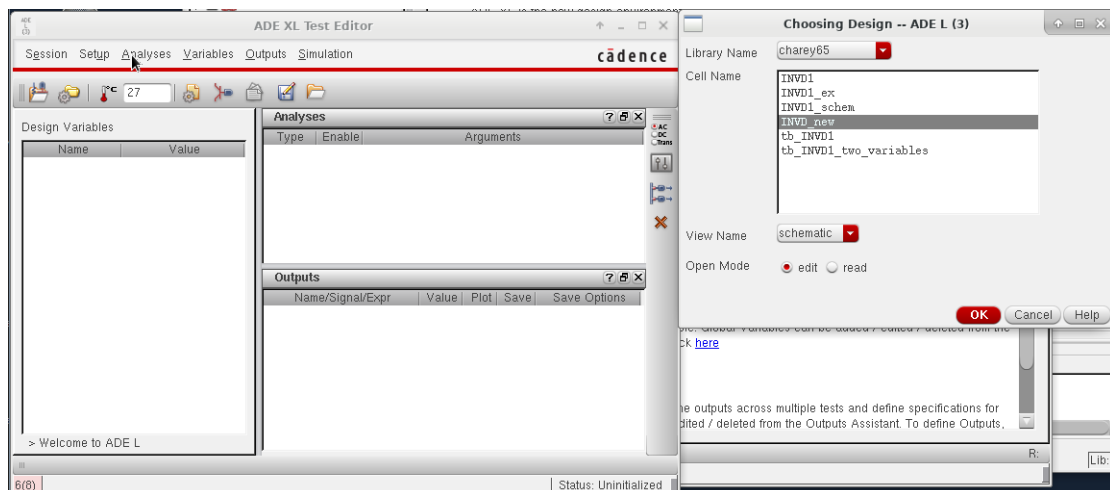
After drawing the testbench schematic, you can launch **ADE XL** by click on **Launch** -> **ADE XL**, you should see the following window



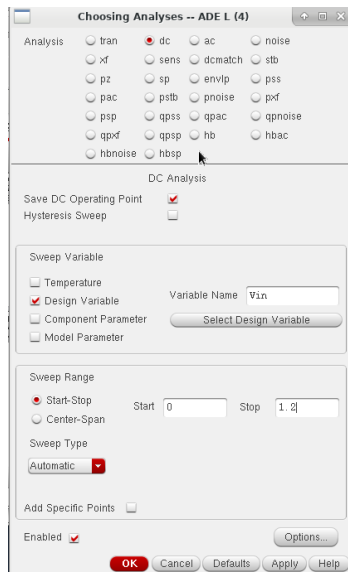
Choose **Create New View** then **OK**, since you have no **ADE XL** view right now. Then click **OK** again you will enter the **ADE XL** view as below:



We start by checking the box before Tests, then press the + and left click on “Click to add test”, you will see the following window pop up

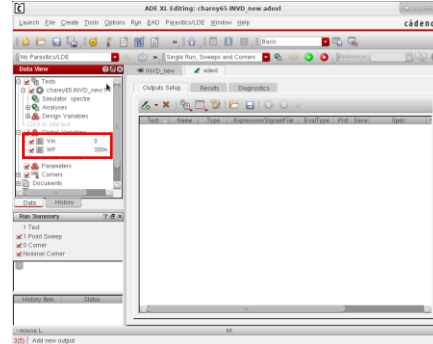


Choose the correct cell which is **INVD\_new** and click OK. Now we should focus on the **ADE XL Test Editor** window, note this window is quite like that of **ADE L**. Go ahead add design variables and add a DC analyses as previously we did in **ADE L**. Here set the default value of Vin as 0, WP as 260n.

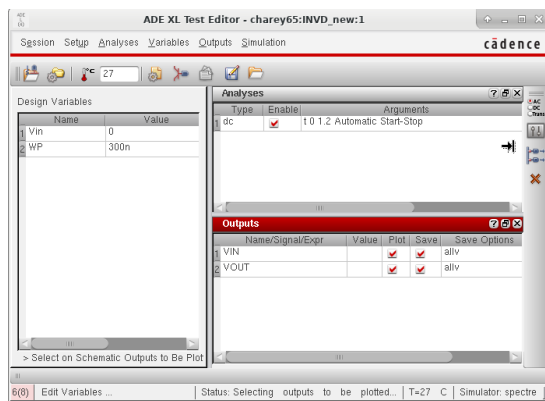


Create a new **dc** Analyses as on the left.

At the Outputs section, let us first add **Vin** and **VOUT**, as below. After finishing these settings, go back to **ADE XL** window, expand **Global Variables**, you can see “**Vin**” and “**WP**” as below.

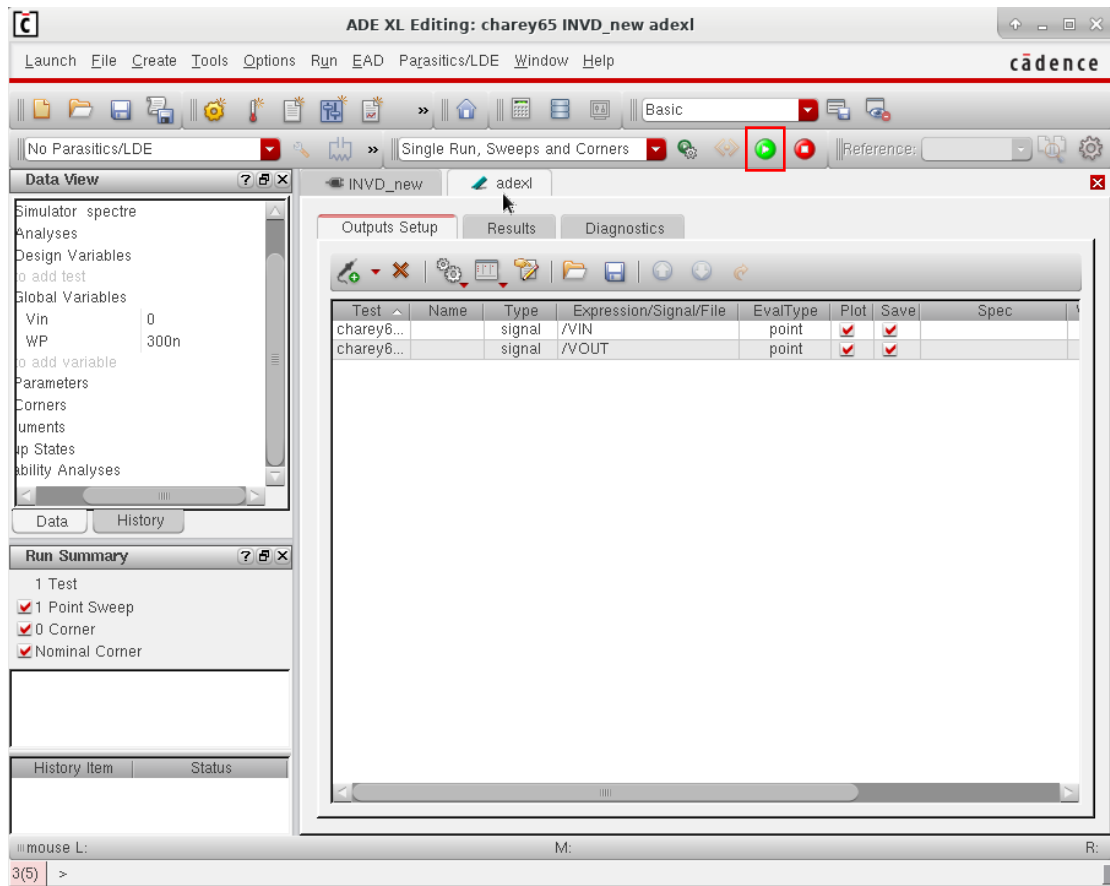


The final setup should look like this:




Now we can go ahead and run in the **ADE XL** for the first time! Go back to the **ADE XL** window and press the green button.

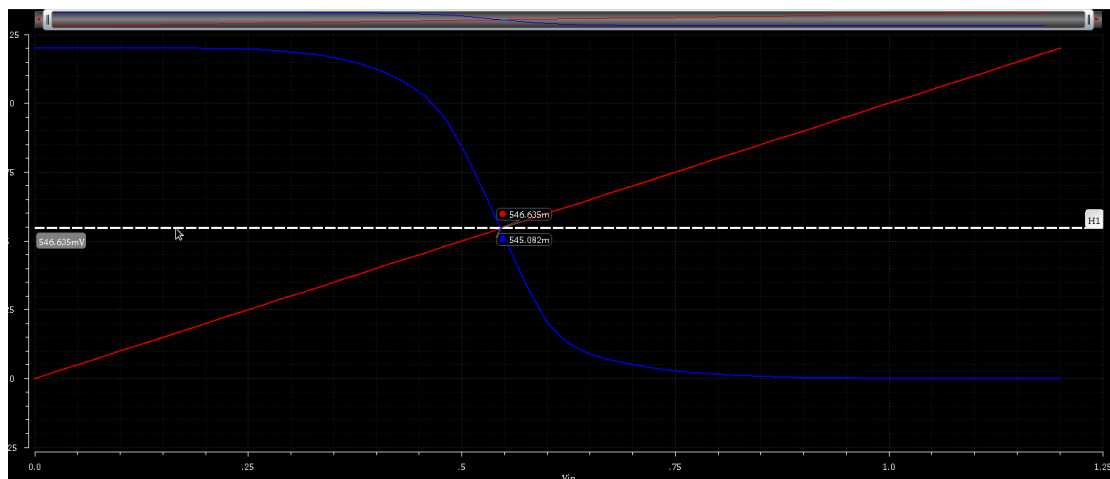




When the simulation is finished, you will be able to see the status at the bottom left corner



turns to finished. The result looks like this , choose them both by ctrl+click on them both -> Plot, you will see the following plot, which should be familiar to you: (The white horizontal line is inserted by press “H”, you can also insert a vertical line by “V”)



We call the intersection of **VIN** and **VOUT** the transition voltage **Vt**.

Now let us use ADE XL to sweep multiple WP values, and determine the WP when  $V_t = V_{DD}/2 = 0.6V$ .

As we have learnt how to use calculator, we can send both signals to calculator and process them. To do them, go to **Tools -> Calculator**. In the **Function Panel**, search for **intersect** and select it. Click **vs**, which means “sweep voltage”, then the schematic should pop up.



Select **VIN** first, you will see a signal added to the calculator, named **VS("/VIN")**. Copy and paste it after **Signal1** of **intersect** function (or select “**buffer**” from the drop-down menu). Similarly, then choose **VOUT** in the schematic as **Signal2** for intersect function. Then click on OK.

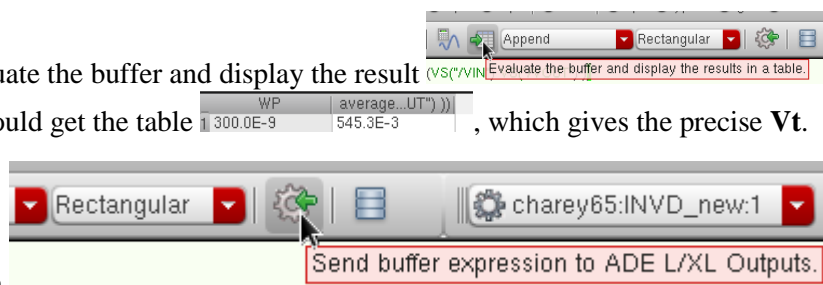


This function will return x and y coordinate of the two curves' intersection. In our case,  $y=x$ . We can use **average** function to obtain a single value as output. Again, search **average** in the **Function Panel**, and left click on it. You should see the expression in the calculator become **average(intersect(VS("/VIN") VS("/VOUT") ))**

click on evaluate the buffer and display the result **VS("/VIN")** in a table, you should get the table 

WP	average...UT") )
300.0E-9	545.3E-3

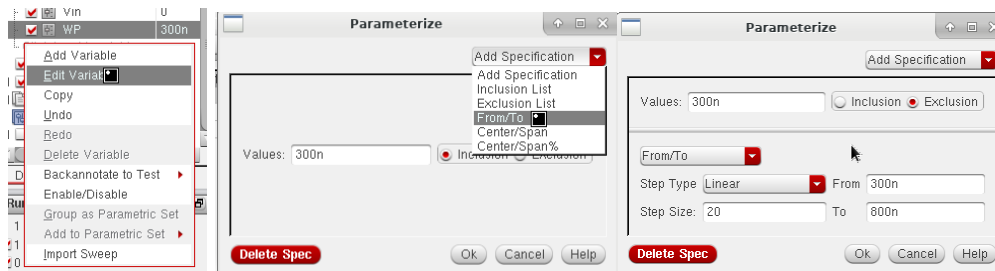
, which gives the precise **Vt**.



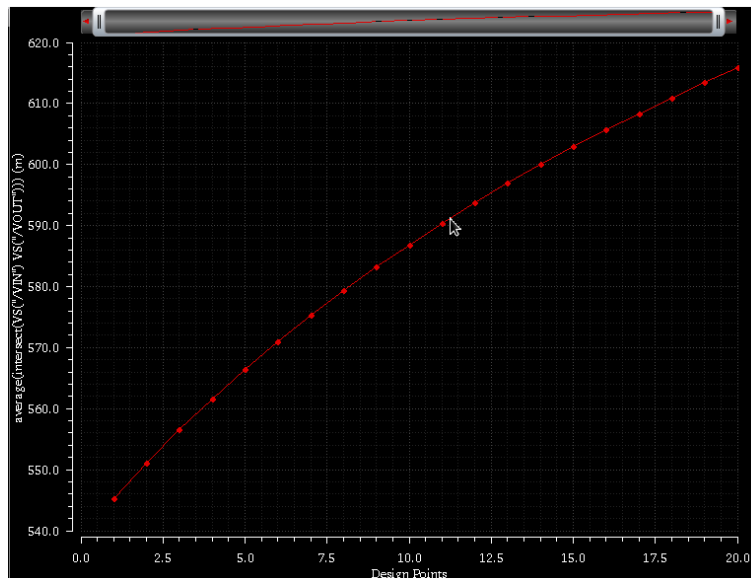
Then click on **Send buffer expression to ADE L/XL Outputs.**, the expression is then added to the simulator outputs.

Outputs						?	⌵	⌶
	Name/Signal/Expr	Value	Plot	Save	Save Options			
1	VIN		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	allv			
2	VOUT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	allv			
3	average(intersect(VS("/VI...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				

Then go back to **ADE XL**, define the sweep about **WP** by **right click WP->Edit Variable->From/To**, set the sweep. Sweep **WP** from **200 nm** to **500 nm** with a step size of **20**.



Then click on the green button and wait. The simulation takes some time. After the simulation finishes, **right click on the average output->Plot across design points**, you will be able to see this figure below (likely with slightly different values).



## Deliverable

### Calculator

Show a snapshot where both the falltime and risetime are in the **Outputs** list.

### ADE XL

Show a snapshot of **Vt vs Design points (WP values)**, then answer the question we asked originally:

What is the WP that makes  $V_t = V_{DD}/2$ ?

## reference

[1] <https://www.youtube.com/watch?v=BkE1XsnidEk>

[2] [https://community.cadence.com/cadence\\_blogs\\_8/b/cic/posts/things-you-didn-t-know-about-virtuoso-ade-xl](https://community.cadence.com/cadence_blogs_8/b/cic/posts/things-you-didn-t-know-about-virtuoso-ade-xl)

[3] [https://www.youtube.com/watch?v=l\\_9NRyphPPg](https://www.youtube.com/watch?v=l_9NRyphPPg)