Cadence Tutorial (Part Two)

By Kerwin Johnson

Version: 10/24/05

(based on 6.776 setup by Mike Perrott)

Table of Contents

Introduction

In this second tutorial we will build on what we learned in the first tutorial. We will learn more sophisticated modeling techniques and more powerful simulation skills. We will use them to answer the question, “For a given load capacitor and a maximum required rise and fall time what is the minimum power required to meet these specs?”

For this we will find useful models that automatically include the increase in source and drain depletion capacitance with increases in the device size. So we will use a parameterized subcircuit around the mos1 model that we created in the first tutorial.

In order to find the rise time, the fall time and the power per cycle we will run a transient simulation.

In order to answer the optimization question, we will use a parametric simulation as an insightful way to pursue the answer.

Upgrade the Device Models

We remember from the first tutorial that we didn't include the variation of lambda with length in the device model. Also the variation of the source and drain impedance is controlled by setting it per instance in the schematic. This is tedious. We will change the models.

We will add two new techniques to our modeling repertoire which we have seen in netlists previously. The first is that we will create our entire model within a subckt statement the same as our inverter was when we looked at the netlist. Secondly, we will use parameter statements to re-calculate the source and drain values for each instantiated mos device. Again we used this technique to vary the input DC source in the first tutorial.

We will also use one piece of spectre magic, called the inline statement. The inline statement precedes the word subckt and it makes the instantiated element within the subckt with the same name look like the entire subckt for the purposes of probing. All this means is that when we are plotting data, the subckt containing a pmos looks like a pmos model directly for the purposes of plotting.

Edit your mos6012.scs file and add the following. You can leave the models from the first tutorial intact because we are changing the model name from nmos6012 to nmos6012p for parameterized.

```
//   Created by: Kerwin Johnson Sept 2005.


simulator lang=spectre insensitive=yes


inline subckt nmos6012p (d g s b)
  parameters w=3e-6 l=1.5e-6 as=1.35e-11
  + ad=1.35e-11 ps=1.2e-5 pd=1.2e-5
  + nrs=1.5 nrd=1.5 lds=4.5e-6 ldd=4.5e-6


  model nmos6012i mos1
  + type = n
  + l = l
  + w = w
  + vto  = 0.75
  + kp   = 100e-6
  + lambda = 7e-2 * (1.5e-6 / l)
  + phi = 0.7
  + gamma=0.6
  + tox  = 1.5e-8
  + cj = 1e-4
  + cjsw = 5e-10
  + pb = 0.9
```

```
  + lds = lds
  + ldd = ldd
  + as = max( lds * w, as)
  + ad = max( ldd * w, ad)
  + ps = max(w + 2 * lds, ps)
  + pd = max(w + 2 * ldd, pd)

  nmos6012p (d g s b) nmos6012i

ends

inline subckt pmos6012p (d g s b)
  parameters w=6e-6 l=1.5e-6 as=2.7e-11
  + ad=2.7e-11 ps=1.5e-5 pd=1.5e-5
  + nrs=1.5 nrd=1.5 lds=4.5e-6 ldd=4.5e-6

  model pmos6012i mos1
  + type = p
  + l = l
  + w = w
  + vto  = -0.75
  + kp   = 50e-6
  + lambda = 7e-2 * (1.5e-6 / l)
  + phi = 0.7
  + gamma=0.6
  + tox  = 1.5e-8
  + cj = 3e-4
  + cjsw = 3.5e-10
  + pb = 0.9
  + lds = lds
  + ldd = ldd
  + as = max( lds * w, as)
  + ad = max( ldd * w, ad)
  + ps = max(w + 2 * lds, ps)
  + pd = max(w + 2 * ldd, pd)

  pmos6012p (d g s b) pmos6012i
```

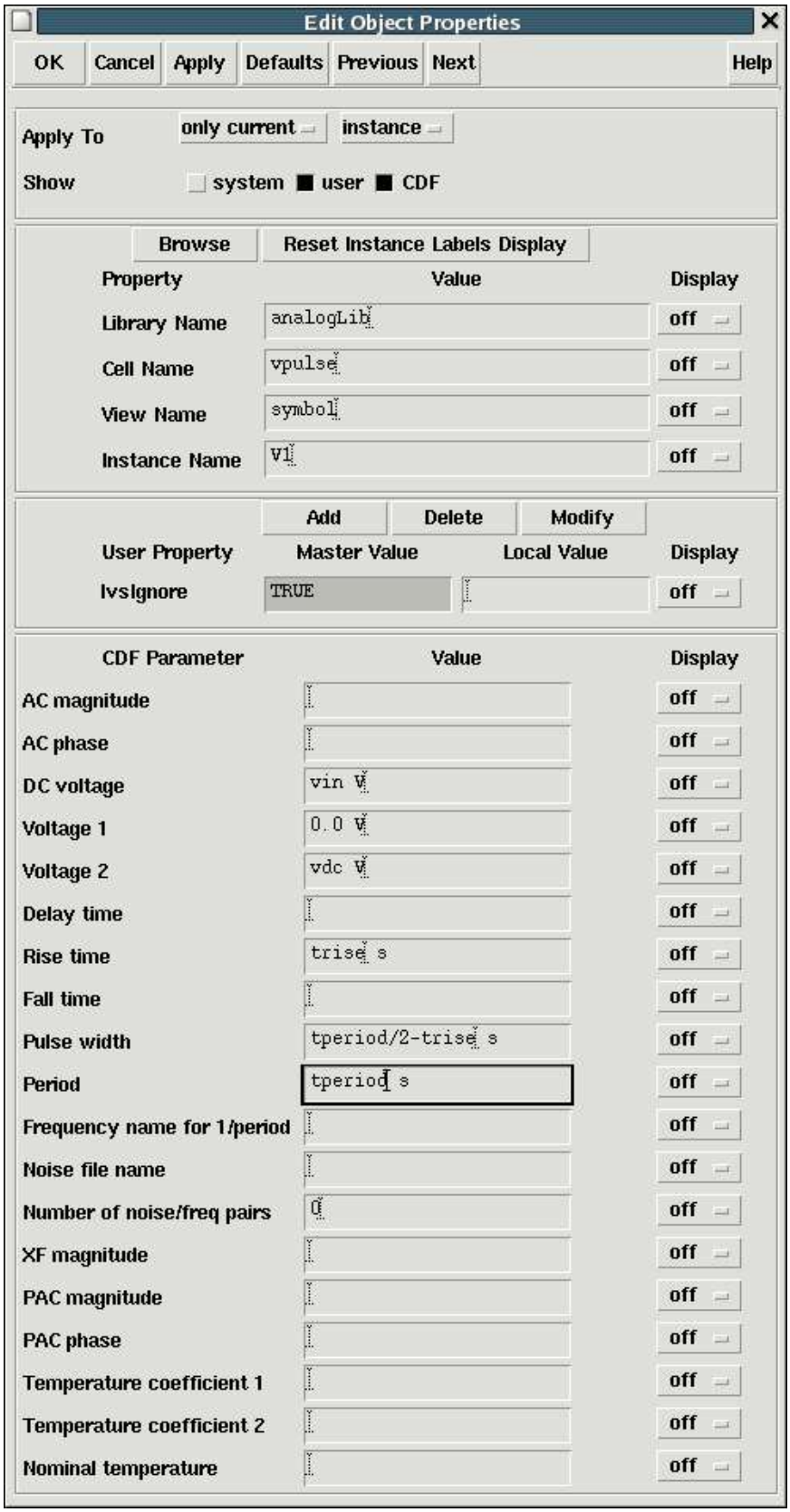
```
ends
```

If you have changed the name of the model file, then you need to change the location in ADE as well.

Update the Schematic for the Transient Simulation

We want to run a transient simulation to measure rise time, fall time and power consumed by the inverter driving a load capacitance of 500f. We will start with the previous test_inverter schematic and change what we need as we go along. To begin we need to change the input from a dc source into a square wave, and we need to change the models that the mos devices are referencing to our new models. Do the following.

1. Open the library browser CIW->Tools->Library Manager and open the test_inverter schematic.
2. Change the models on the FETs inside the inverter. Descent into the inverter (E), (q)uery the pmos and change its model to pmos6012p. Change the nmos model to nmos6012p. Check and Save (X) and then ascend (Ctrl-e) to the test_inverter schematic.
3. Change the input source to a square wave. (q)uery the vdc used for vin. Change the cell name to vpulse. Set voltage 1 = 0, voltage 2 = vdc, rise time = trise, period = tperiod and pulse width = tperiod/2-trise. When you are done compare it with the picture below and then click OK.



4. Change the load capacitance to 500f.
5. Check and Save (X).

Transient Simulation.

In this section we will run ADE and configure a transient simulation to run. In order to do this we will load the previous dc simulation that we ran and add a transient simulation to it. If you didn't save the previous dc simulation, then follow the instructions in the previous tutorial to setup the dc simulation.

1. Start ADE by Tools->Analog Environment.
2. Load your final state from the first tutorial, with Session ->Load State. Pick the name that you saved it under and click OK.
3. Add the design variables for the vpulse source with Variables->Edit. Set trise = 1n and tperiod = 1u. Click OK.
4. Add a transient simulation. Click Analyses->Choose. Choose a stop time of 6u, which will give plenty of time for any start up behaviour to die out so that we can measure our result on the fifth cycle. Check that your netlist matches the following.

5. Run the simulation by clicking on the green light.

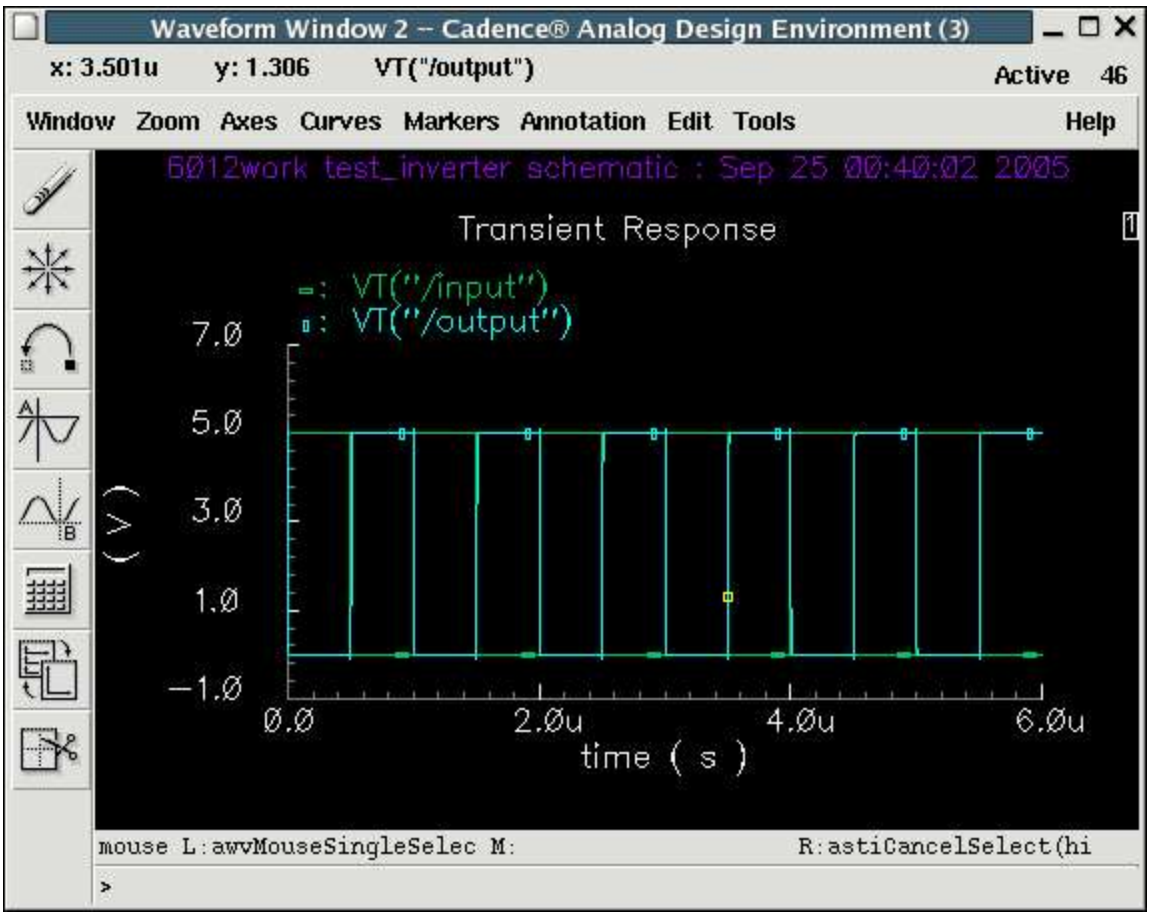
Analysis

Plot the Transient Output

In this section we will plot the transient outputs, show a quick way to measure time differences in the waveform window and calculate rise time, delay and power consumed.

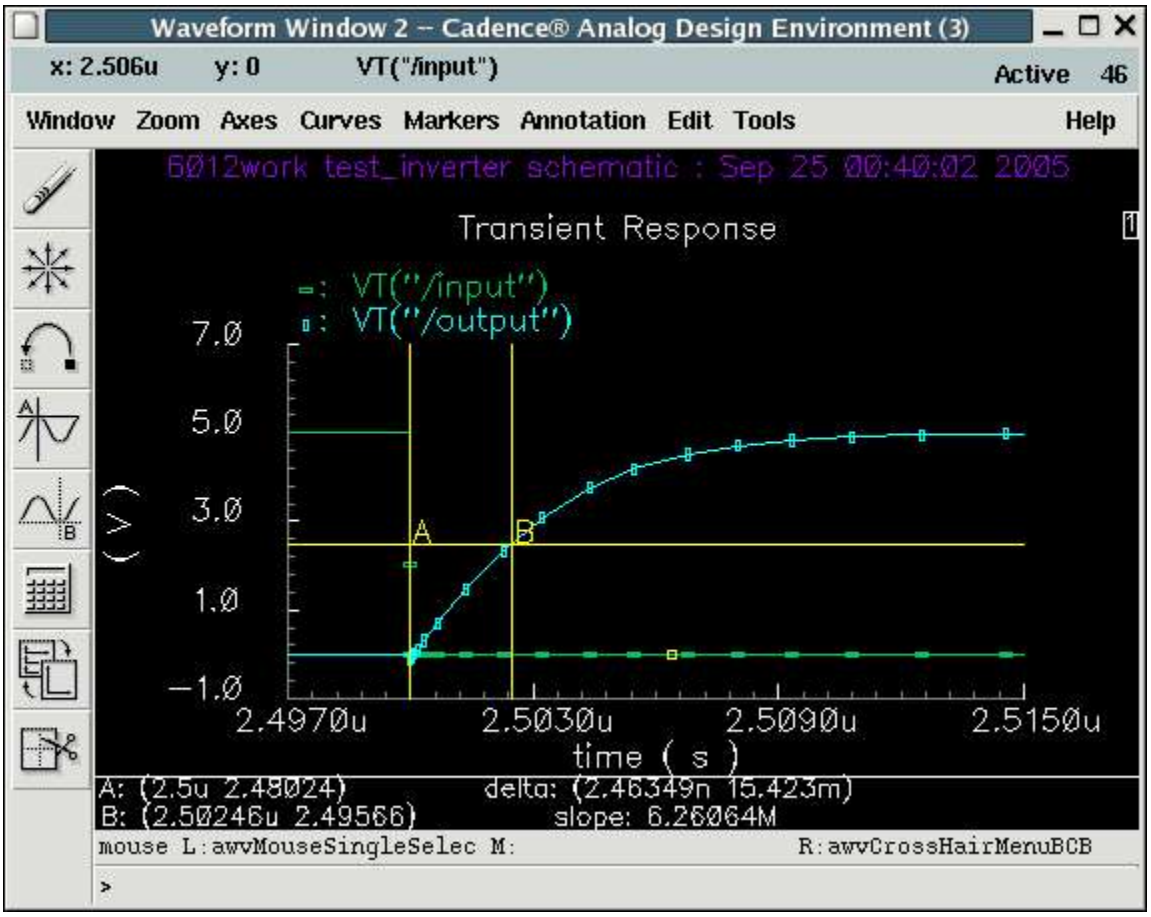
First we plot.

1. Click ADE->Results->Direct Plot->Transient Signal. Select the output and the input and press Esc.



Courtesy of Cadence Design Systems, Inc. Used with permission.

2. We can use cursors to get quick estimates of delays in transient plots. Zoom in on any edge using the right mouse button or the zoom menu. Press (a) and left click near 2.5 V on the input waveform. Press (b) and left click near 2.5V on the output waveform. Notice that in the bottom of the waveform window it reports the cursor position of both cursors and their difference, as shown below.



Courtesy of Cadence Design Systems, Inc. Used with permission.

Calculate Delays, Rise Times and Power Consumption.

We can use the Calculator to calculate the rising and falling delays through the inverter and the rise and fall time of the output signal. Once again they are precomputed calculations.

1. Open the Calculator with ADE->Tools->Calculator.
2. We will calculate rise time first.
   a) Select the transient output signal. Click vt for transient voltage and then select the output node in the schematic window. The calculator value should be VT ("/output"). You can also just type these values in once you are familiar with the equation syntax.
   b) Clip the output waveform so that we are only looking at the section of interest. Click SpecialFunctions->Clip. Enter 4.5u and 5u as the times and click OK. Plot this if you want to check that you've clipped the correct portion. Clipping makes sure that you have grabbed a section of the waveform away from any settling transients that you do not want to measure.
   c) Use the special functions list to calculate the nearest rise time. Click SpecialFunctions->riseTime. Fill out the form as follows then click OK. This will calculate the rise time of a signal changing from 10% of the difference above the

initial value to 90% of the difference above the initial value. This avoids counting long tails as part of the rise time.

Courtesy of Cadence Design Systems, Inc. Used with permission.

d) The final formula should look like.

riseTime(clip(VT("/output"),4.5u,5u),0,nil,5,nil,10,90)

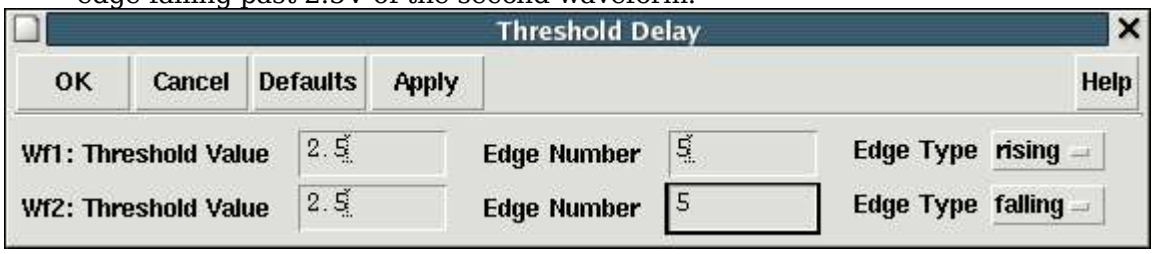
e) Add this to your ADE outputs window. Plot it or evaluate the buffer (which will consume the formula, so remember to push it onto the stack) to see the value.

f) Do the same for fall time.

3. Now we will calculate the delay through the inverter. We will calculate the delay from the mid point of the input waveform to the midpoint of the output waveform.

a) Click on vt. Then in the schematic window select the output node and then the input node, in that order.

b) Use SpecialFunctions->delay set as follows and then click OK. This computes the delay from the fifth edge rising past 2.5V of the first waveform to the fifth edge falling past 2.5V of the second waveform.



Courtesy of Cadence Design Systems, Inc. Used with permission.

c) The expression in the window should be:

delay(VT("/input"),2.5,5,"rising",VT("/output"),2.5,5,"falling")

d) Add this to your ADE->outputs as falldelay.

e) Compute the risedelay and add it to your ADE->outputs.

4. Finally, we will calculate the power consumed by the circuit. Since the entire circuit in the test_inverter schematic is part of the device under test(DUT) we can say that all of the current from our 5V vdc is power consumed by the DUT. We will measure that power over one cycle We are intentionally ignoring the power supplied by the input supply to the gate of the inverter.

a) Select the current out of the power supply. In the calculator click it and select

the red dot on the positive terminal of vdc symbol in the schematic. This will write IT("/V0/PLUS") in your calculator, if V0 is the name of your supply.

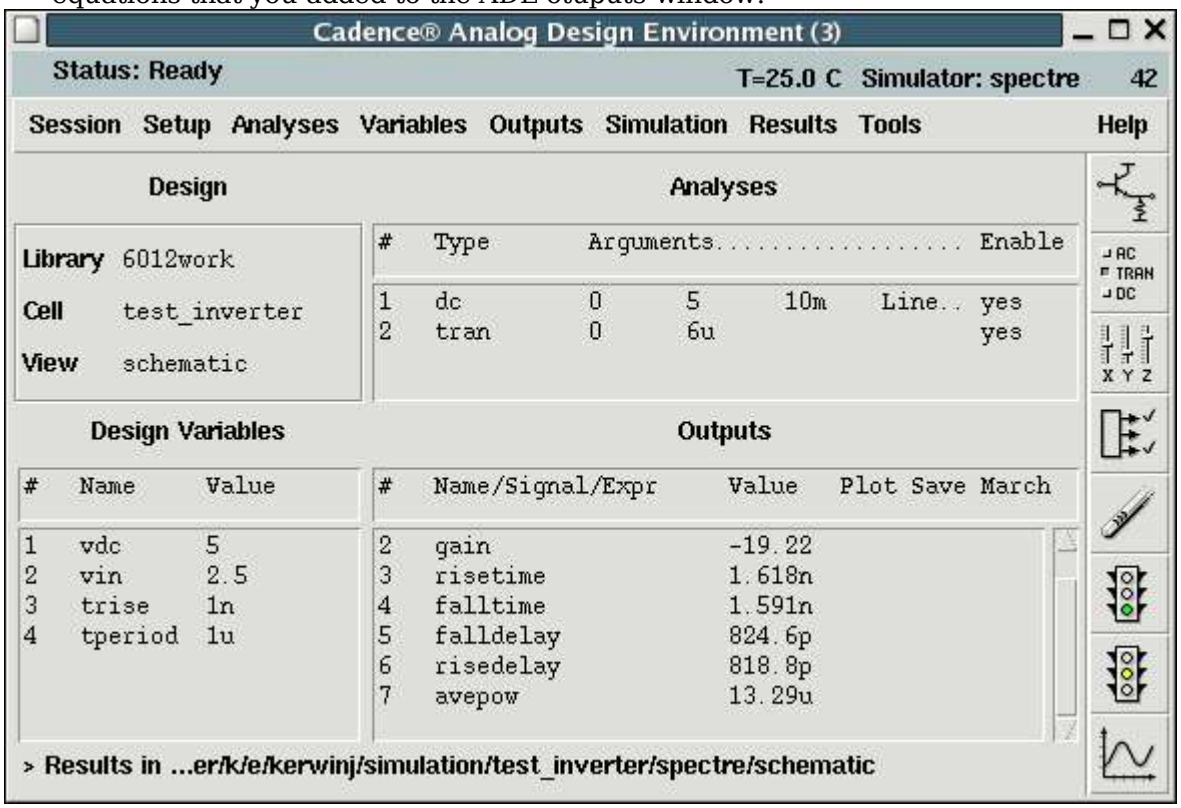
b) Calculate the instantaneous power. The instantaneous power will be the current leaving the supply times the voltage or -IT("/V0/PLUS")*VT("/vdd!"). You can plot this if you want.

c) Now calculate the average power consumed per period by integrating over one cycle and dividing by the duration. Use SpecialFunction->integ. Start at 4.5u and end at 5.5u. Divide by 1u. The formula in the calculator window will look like:

integ(-IT("/V0/PLUS")*VT("/vdd!"),4.5u,5.5u)/1u

d) Add this to your outputs as avepow.

5. Now go to ADE->Results->Plot Outputs->Expressions to evaluate all of the equations that you added to the ADE otuputs window.

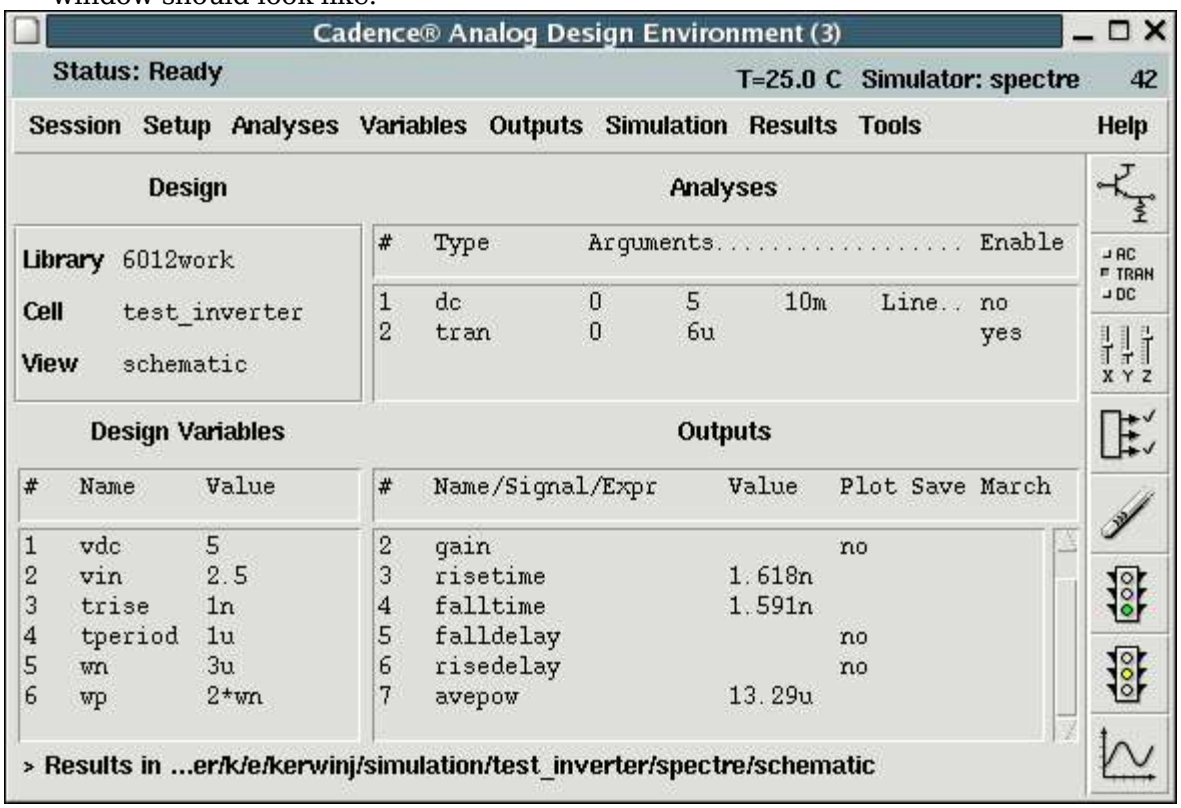


Parametric Simulation

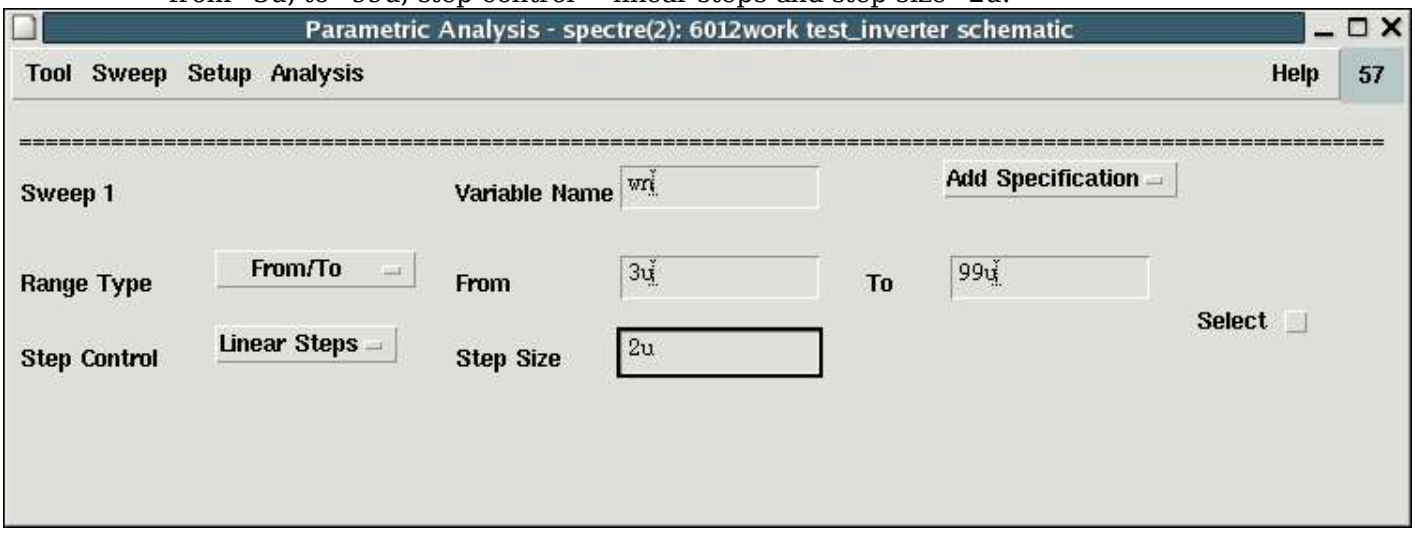
Here we return to the original question. What is the minimum power and minimum device width required for rise and fall times below 1/0.5/0.1 ns for our 500fF load? We are going to use parametric simulation to answer this question quickly. Parametric simulation means that we will run the transient simulation multiple times, while we sweep one or more design variables. It works like nested for loops. You can add multiple nested sweeps.

In our case we want to sweep the inverter's nmos and pmos width, while we look at the output power and rise and fall times. We will just plot this pick off the points of interest by eye.

1. First we will change our inverter so that we can sweep the NMOS and PMOS widths in the simulator. D(E)scend into the inverter and change(q) the pmos width to wp and the nmos width to wn. Remember to Check and Save(X) before you ascend(Ctrl-e).
2. Then add the variables wn and wp to the ADE->Variables->Edit window with defaults of wn =3u and wp =2*wn.
3. We want to look at rise time, fall time and power. Turn off the gain, vcross and delay measures in ADE->Outputs->Setup.
4. We only want to run the transient simulation this time so turn off the DC simulation. Click Analyses->Choose->dc and at the bottom of the form turn off the enabled button. Run the simulation once to make sure you haven't botched anything. The window should look like:

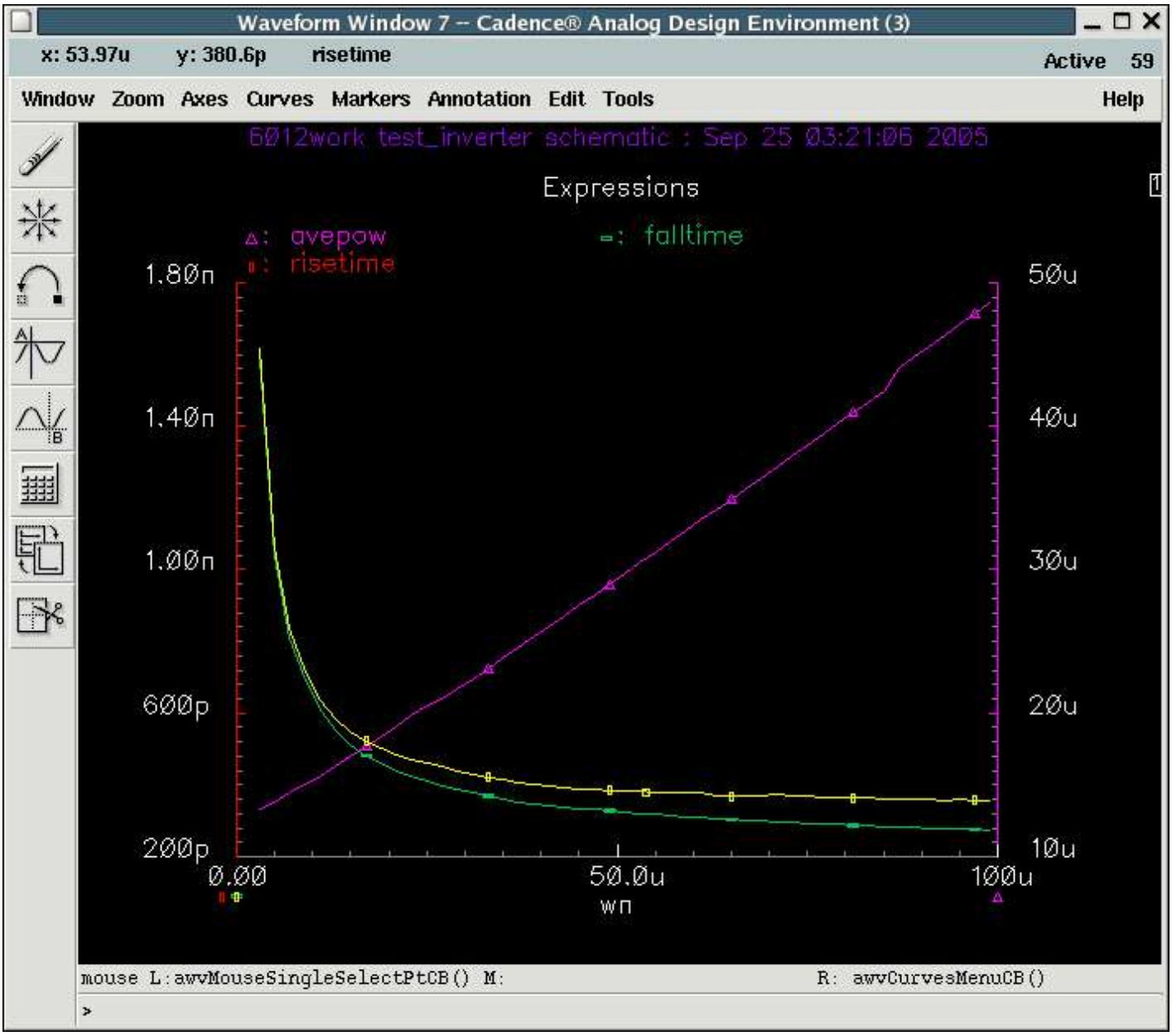


5. Setup the parametric simulation to sweep wn from 3u to 99u in 2u steps. Click on ADE->Tools->Parametric Analysis. Fill out the window with variable=wn, from=3u, to=99u, step control = linear steps and step size=2u.



Courtesy of Cadence Design Systems, Inc. Used with permission.

6. Open a new Waveform window, because parametric analysis will plot as soon as it completes and overwrite any plot window that you may have open. ADE->Tools->Waveform.
7. Start the parametric analysis. Parametric->Analysis->Start.
8. Once it plots change the axis so that both the rise and fall time are on the same axis. Curves->Edit. Then select falltime and change the axis to match risetime.
9. Now by inspection answer the questions. What is the minimum power and minimum device width required to yield rise and fall times below 1/0.5/0.1 ns given a 500 fF load?



Courtesy of Cadence Design Systems, Inc. Used with permission.

As you can see parametric analysis can allow you to explore 1, 2, 3 or more dimensional slices through your design space, if you can imagine a way to tie a variable to some portion of the design space.

More Play time.