

Cadence Tutorial D: Using Design Variables and Parametric Analysis

Created for the MSU VLSI program by Casey Wallace

Last Updated: 02/20/2006

Document Contents

Introduction

Using Design Variables

Running a Parametric Analysis

Passing (Localized) Parameters

Notes about Design Variables and Parametric Analysis

Introduction

Design variables provide a method for specifying circuit parameters just before a simulation. This allows you to make quick changes to parameter values across single or multiple circuit components without having to edit any schematics schematic. Design variables can also be used to enable parametric analysis, which can sweep a design variable across a range of values and displays multiple simulation results on a single plot.

Using Design Variables

This part of the tutorial will show you how to parameterize the width of the NMOS and PMOS transistors of an inverter by using design variables. These variable values are then set using the Analog Design Environment prior to simulation.

1. Make a copy of the inverter cell schematic you have previously designed, or create a new one.
2. Open the schematic of the inverter
3. Left click on the NMOS transistor to select it, and view its properties by center clicking on it and selecting Properties. (or you can press 'q' as a shortcut).
4. Delete the text in the width box, and type `nw`. This specifies that a variable called `nw` will be used to set the width of the NMOS. When you do this, observe that values for "Drain/Source Diffusion Area" change from a constant value to a formula that involves the variable `nw`. Always check to make sure that formula is updated when you change the width to a variable; otherwise, your simulation results will be incorrect. Click **OK** to apply the change.
5. Repeat this procedure to change the width of the PMOS to `pw`.
6. Select **Design** → **Check and save**. Verify that there are no errors.
7. Launch the **Affirma Analog Design Environment**.
8. Set up the simulation environment as you normally would for simulating a schematic. This includes specifying a stimulus file and transient run length, enabling the SPICE netlist reader in the environment options, and selecting the outputs to be plotted on the schematic.
9. In **ADE**, select **Variables** → **Copy from cell view**. The design variables `nw` and `pw` should appear in the Design Variables area of the **Analog Environment Window**.
10. In **ADE**, select **Variables** → **Edit**. Left-click on `nw`. In the value box, type `1.5u` and then apply the value by clicking **Apply**. Change the value of `pw` to `3.0u` and click **Apply**. Once the values have been assigned, click **OK**. See Figures 1 and 2 below.
11. In **ADE**, run the simulation by selecting **Simulation** → **Netlist and Run**. After waiting for the simulation to complete, the results window will pop up. The results should be

identical to those you would see if you sized the PMOS and NMOS widths to 3.0um and 1.5um, respectively.

- In ADE, select **Variables** → **Edit**. Left-click on **pw**. In the value box, type **1.5u** and then apply and run the simulation by clicking **Apply & Run Simulation**. A new waveform results window will pop up after a short time. Changing design variables in this way provides a quick way to run different simulations without needing to edit and save the schematic in between.

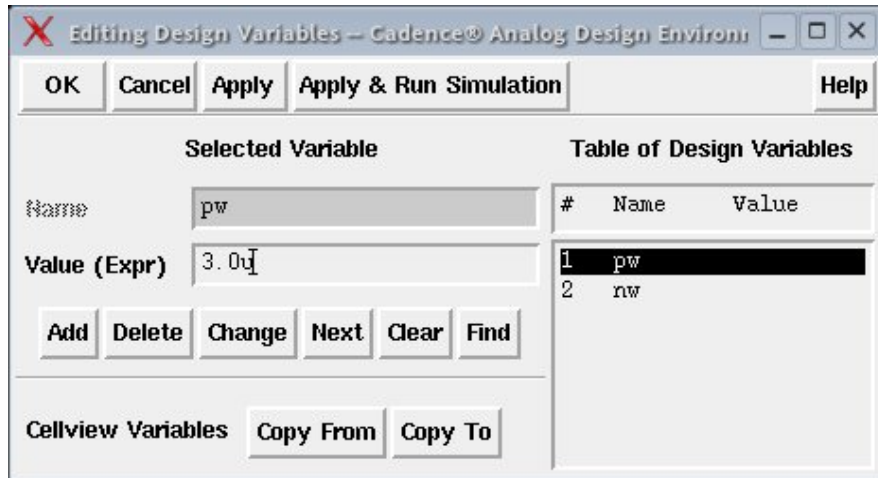


Figure 1: The Editing Design Variables dialog window.

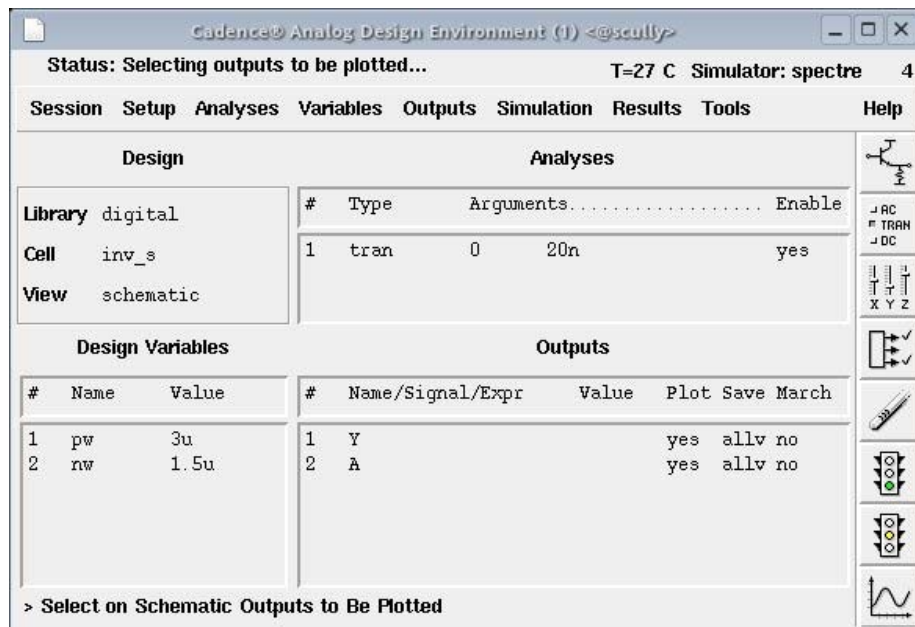


Figure 2: Analog Design Environment after initialization.

Running a Parametric Analysis

This part of the tutorial will show you how to set up a simple parametric sweep analysis in Cadence. In this example, the width of the PMOS transistor is swept from 1.5um to 3.0um in 11 linear steps, and each waveform is plotted in the same results window. This allows you to observe the effect of increasing the transistor size ratio on the delays of the inverter circuit

through a single simulation.

1. Perform any steps 1-8 from the previous section needed to parameterize your inverter size by values `nw` and `pw` and to set up the [Analog Design Environment](#) for simulation.
2. In [ADE](#), select [Variables](#) → [Copy from cell view](#). The design variables `nw` and `pw` should appear in the Design Variables area of the Analog Environment Window.
3. In [ADE](#), select [Variables](#) → [Edit](#). Left-click on `nw`. In the value box, type `1.5u` and then apply the value by clicking [Apply](#). Leave the value of `pw` blank. Once the values have been assigned, click [OK](#).
4. In [ADE](#), select [Tools](#) → [Parametric Analysis](#). This should bring up the parametric analysis window.
5. In the box named Variable Name, enter `pw`. In the From box, enter `1.5u`. In the To box, enter `3.0u`. In the Total Steps box, enter `11`. Verify that the Step Control selection box is set to “Auto”.
6. Verify that the [Parametric Analysis](#) window appears as it does in Figure 3. Once you are certain, begin simulation by selecting [Analysis](#) → [Start](#).
7. After waiting for the simulation to complete, the results window will pop up. If you have selected the input and output to be plotted, there will be 22 output waveform present - 11 identical inverter input waveforms, and 11 different output waveforms.
8. Depending on the number of simulations being displayed you may need to maximize the results window to see all of the legend text. You can also turn each waveform display on or off. Select [Curves](#) → [Options](#). A Curves window should pop up. Select the name of the curve by clicking on it and press [On](#) or [Off](#) and then [OK](#). There are many other useful options in this window for changing the display of results.

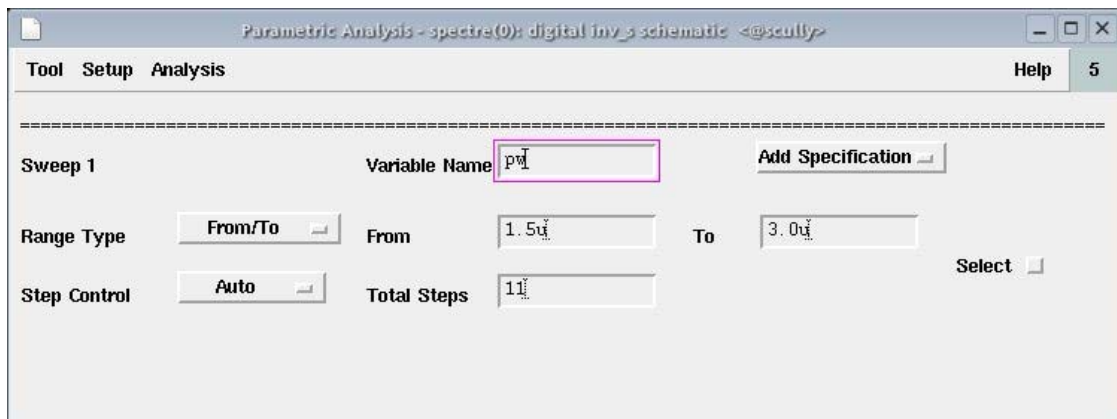


Figure 3: Parametric Analysis dialog window.

Passing (Localized) Parameters

Design variables as applied above are useful for setting parameters that are defined globally – such as the value of the minimum transistor width, or VDD. However, it is often necessary to have localized parameters that apply only to an instanced cell, and not to all cells in a design. For example, you may want to create a single inverter schematic with parameterized transistor sizes, but you want every instance of the inverter to have a custom size. This part of the tutorial describes how to use the `pPar()` function in order to pass parameter values from a parent cell to its constituent components.

1. Open the inverter schematic (or create a new one to preserve the functionality of previously-designed circuits)
2. Access the properties for PMOS by selecting it and pressing ‘q’.
3. Set the width value to `pPar("pw")`. This will specify the width of the PMOS as a “passed parameter”. Click **OK** to apply changes. This is shown in Figure 4.
4. Set the width of the NMOS to `pPar("nw")`. Click **OK** to apply changes.
5. Select **Design → Check and save**. Verify that there are no errors.
6. Create a new symbol for the inverter. If you have an existing symbol from a copy, delete it first. In addition, any time you change the number or name of passed parameters in your schematic, recreate the symbol for it. This is an important step; otherwise, you may not be able to use passed parameters properly in later steps. If a window appears and asks you if you want to overwrite base CDF information, click **Yes**.
7. Begin new schematic. Place an instance of the modified/new inverter. Select this instance and open its properties by pressing ‘q’. At the bottom of the dialog, you should see CDF parameters called `pw` and `nw` and a corresponding value box. This is shown in Figure 5.
8. Put any transistor sizes you wish in the value box and press **Apply** or **OK**. This will now pass the value to the parameterized inverter schematic.
9. You may complete the schematic in any way you see fit.
10. If you have specified the sizes explicitly for the inverter cell, simulation of the schematic is no different than described in Tutorial A. However, you can also use design variables to specify the values of passed parameters. This will allow you to set their value from [ADE](#), as well as enable the ability to do parametric analysis. In this case, follow the appropriate simulation procedures introduced in this document.

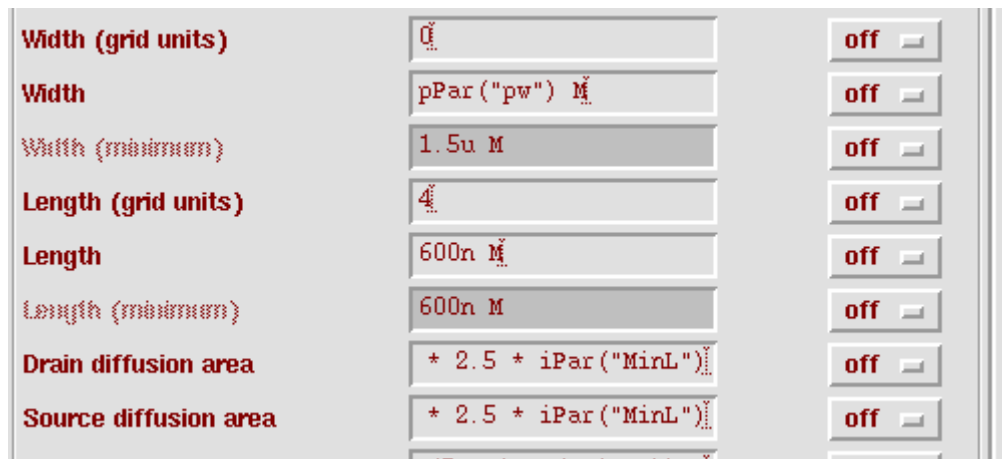


Figure 4: Setting the width of a PMOS transistor to a passed parameter value called “pw”.

View Name	symbol1	off <input type="checkbox"/>		
Instance Name	I0	value <input type="checkbox"/>		
<input type="button" value="Add"/> <input type="button" value="Delete"/> <input type="button" value="Modify"/>				
User Property	Master Value	Local Value	Display	
interfaceLastC..	20 16:20:09 2006		off <input type="checkbox"/>	
partName	inv_2		off <input type="checkbox"/>	
vendorName			off <input type="checkbox"/>	
CDF Parameter			Value	Display
nw			off <input type="checkbox"/>	
pw			off <input type="checkbox"/>	

Figure 5: Object properties for an instance that uses passed parameters.

Notes about Design Variables and Parametric Analysis

- Spectre allows the use of mathematical notation for specifying values in stimulus files and properties of objects in schematics. For instance, you can specify the width of a transistor as “width*2”, or “width+1.5u” from its properties dialog.
- You can express the properties of an object in a schematic in terms of more than one design variable. For instance, you can specify the width of a transistor as “width*size”, allowing you to set the size of the transistor through two design variables called “width” and “size”.
- You can mix and match passed parameters and design variables. For instance, you can specify the width of a transistor as “minWidth*pPar(“size”)”, which would allow you to specify the minimum transistor width globally, but adjust the size locally through a passed scale factor. This could be extended to include multiple design variables and passed parameters in a single expression.
- A limitation of the supplied NCSU transistor models is that you cannot parameterize the length of the transistor. Changing the length to a variable will not update the area formulas, and thus is not reflected through simulation.
- Avoid using design variables that begin with numbers. Also, if you parameterize the “width” property of a transistor, make sure that the value does not begin with a number, but rather a character. For instance, it is OK to specify the width as “pw*2”, but giving it a value of “2*pw” will cause a problem. When the value starts with a character, all area and perimeter values are updated upon a change to the width. If the value starts with a number, it is assumed to be a constant value and therefore the values for the areas and perimeters do not change. This is a flaw in the NCSU implementation.
- Schematics that use design variables can be used for LVS verification, but cannot be used to cross-check transistor sizes. Schematics that use passed parameters cannot be used for LVS (LVS will fail). The best solution is to find the optimal values for the design variables/passed parameters through simulation of the schematic, and then build a new schematic that implements those values. LVS can then be run to compare that new schematic with the extracted layout.