Tutorial 1.5:
The Design and Simulation of a D Flip-flop

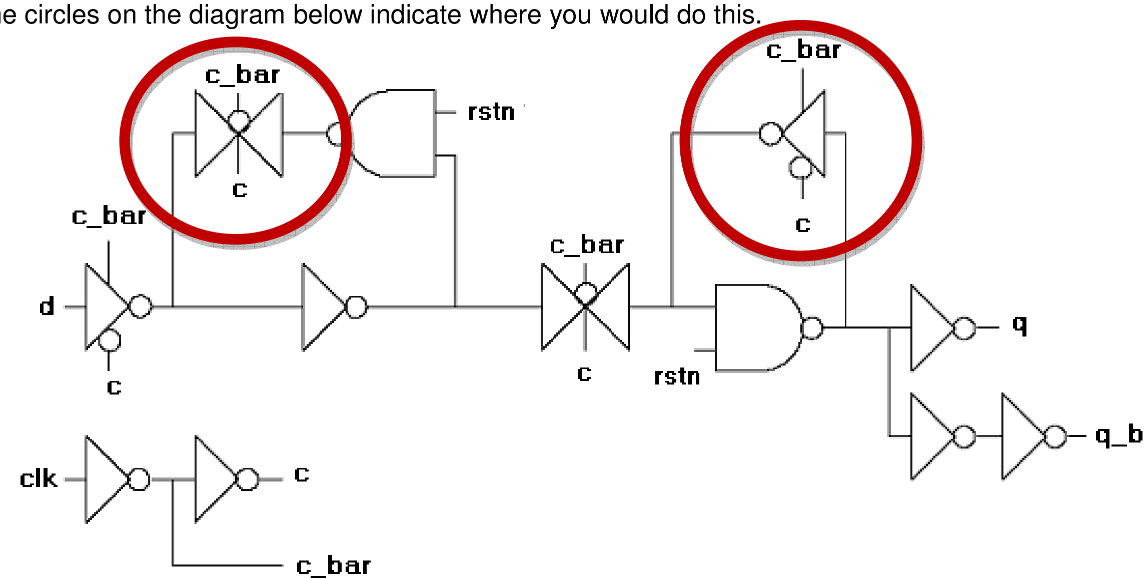
(Last updated: Jan. 15, 2009)

Introduction

Tutorial 1 described the design flow for combination logic. This tutorial describes a few additional details needed to get timing and weak inverters to work in digital simulation.

Adding a High Resistive Transistor

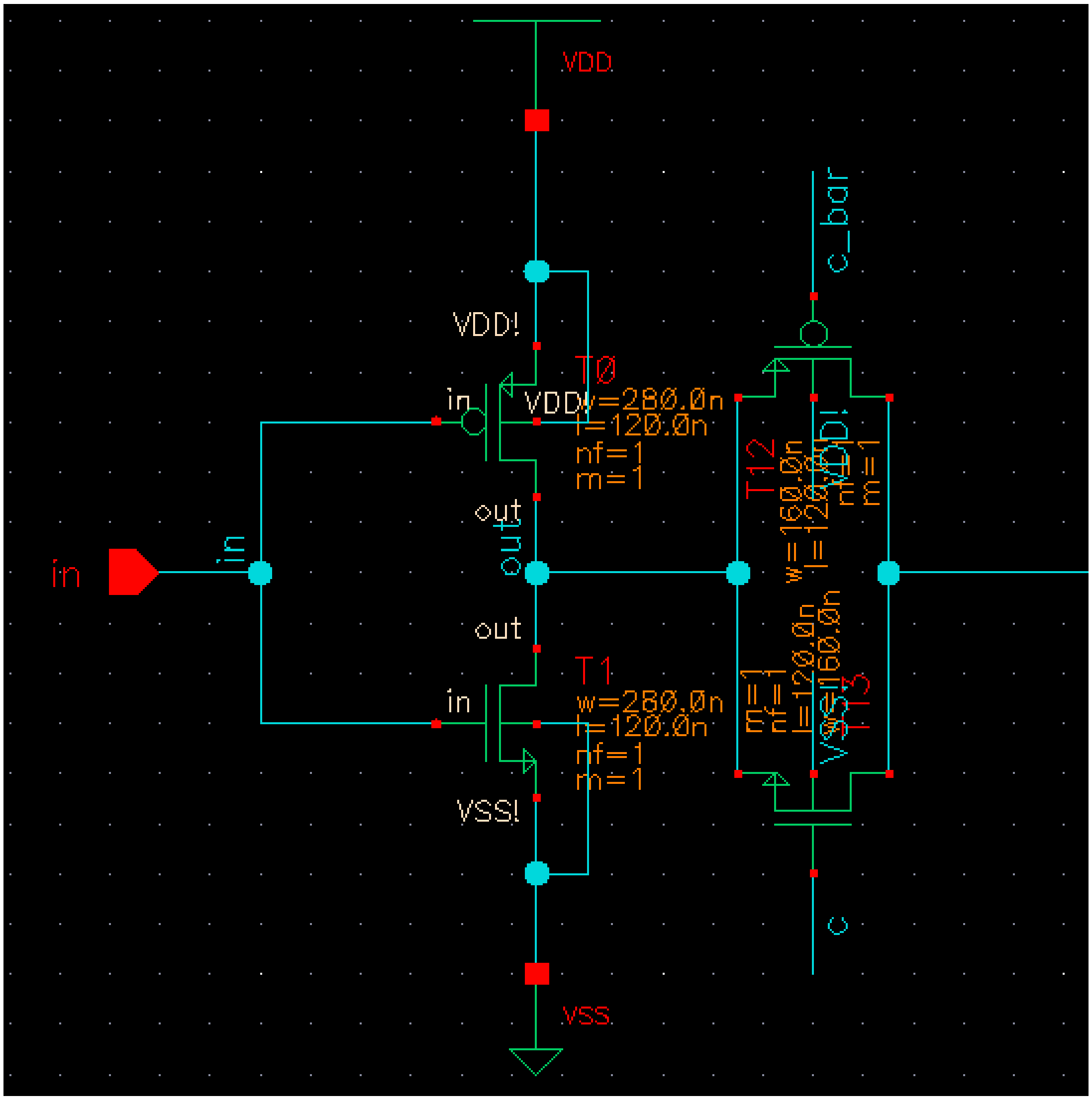
Any time there will be fighting where one transistor should lose; a high resistive transistor is required to ensure the digital simulation will represent this. The transistor should still be sized properly to make it weak and analog spice simulation is required to verify that it will work, but this will tell Verilog that it should lose to the non-weak transistor. Verilog simulations also often have problems with tristate feedback where the simulations hit all sorts of race conditions, 0-delay timing loops, and general screw-ups. The circuit below essentially works properly because any momentary flicker when both tristates are off will be short enough not to break anything. However, Verilog simulations often don't take that into account and the ordering of signals changing can be arbitrary. One simple solution that can fix "instantaneous" conflict issues is to use high resistive transistors in the feeback elements even though they aren't really. Sometimes you might need to add delays to the feedback elements as well. The circles on the diagram below indicate where you would do this.



When entering a schematic, instantiate "rnfet" for high resistive n-type transistors, and "rpfet" for high resistive p-type transistors. In the Verilog simulations there are actually a few drive strengths: "nfet" and "pfet" pass the supply line drive strength of "supply" through unchanged and the "rnfet" reduces that drive strength to "pull." When two devices are driving the same net to two different values, the device with the drive strength of "supply" will overpower the device with the drive strength of "pull." Thus, by using high resistive transistors, a designer can reduce the drive strength of the keeper or feedback device and thus mimic the real circuit in Verilog more closely.

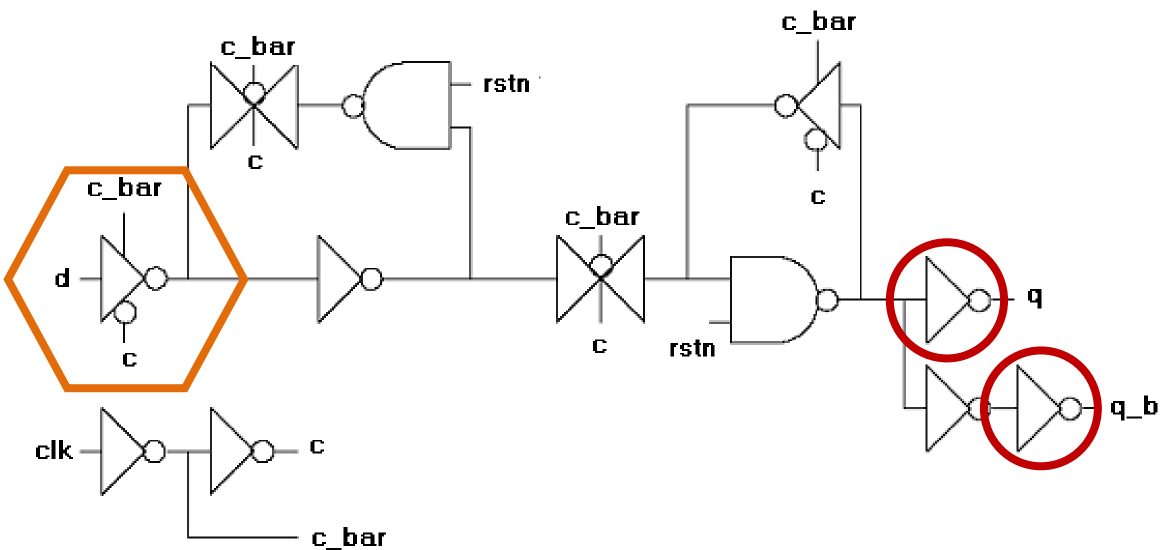
Warning: Watch your transistor direction

Even though a transistor's source and drain connection are supposed to be interchangeable, a design needs to specify the source and drain when entering the schematic. This requirement converts the transistors from bidirectional devices to uni-directional devices in the Verilog simulation resulting in a number of benefits. In addition to a simulation speedup, imagine the problems if every pass transistor suddenly became a potential feedback transistor. There is a way to get a bidirectional transistor in Verilog but getting it to work involves careful consideration of drive strengths of everything around it. To help you visualize the source and drain port of a device (information only flows from source to drain), a little triangle has been added to the symbol of the transistors next to the source connection. The following diagram illustrates the implementation of an inverter and passgate:



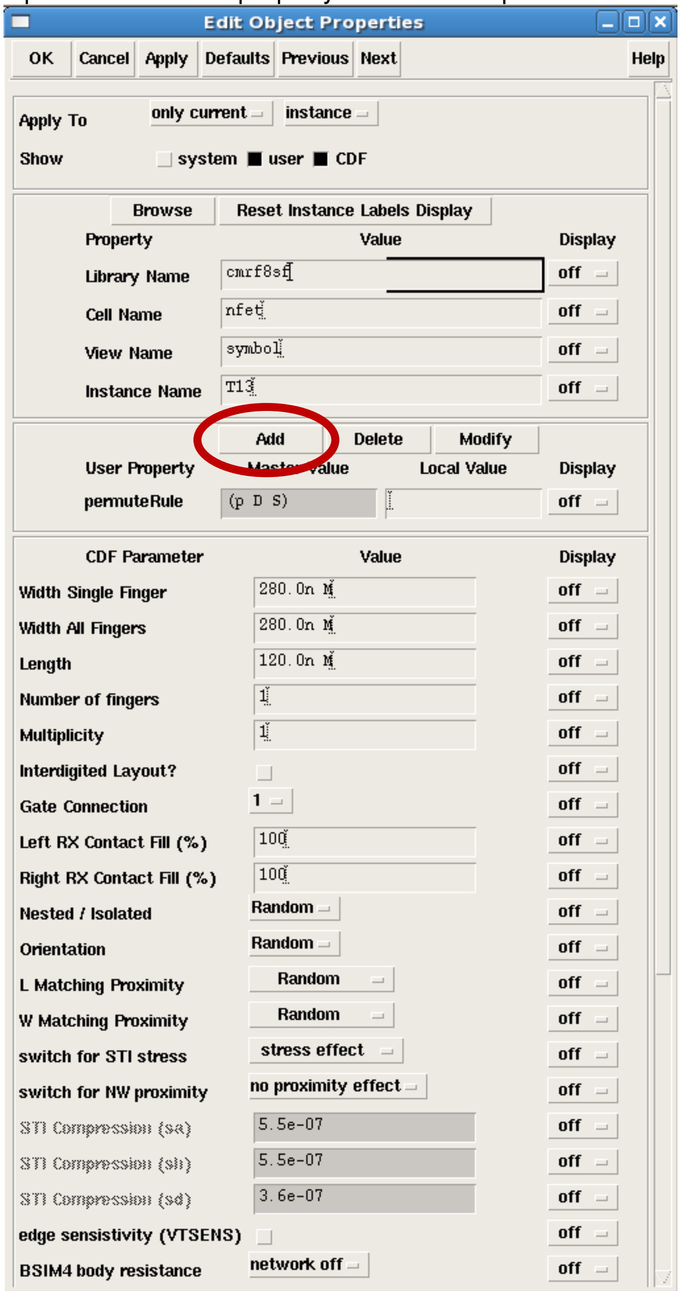
Adding Verilog Propagation Delay for Digital Simulation

To get a more accurate digital simulation, and ensure a trouble-free simulation with the standard cell behavior models, Verilog propagation delays must be added. This step should be after the post-extraction *HSpice* simulation, so you would have some idea about the setup time and clk-to-q delay. You should add the setup time to devices that are connected to input “d,” which are circled in the orange hexagon. Then you should add the clk-to-q or clk-to-q_b delay to the devices circled in red. Hold time is harder to model and, unless you design a very strange register, it should not be an issue so you need not add it.

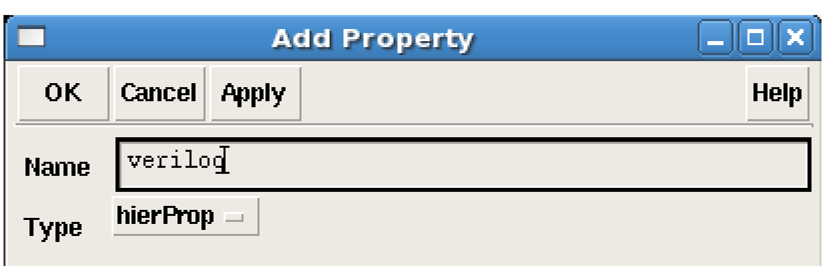


Here are the steps to add Verilog delay into one device. You will need to repeat this step until you have added all of the necessary delays.

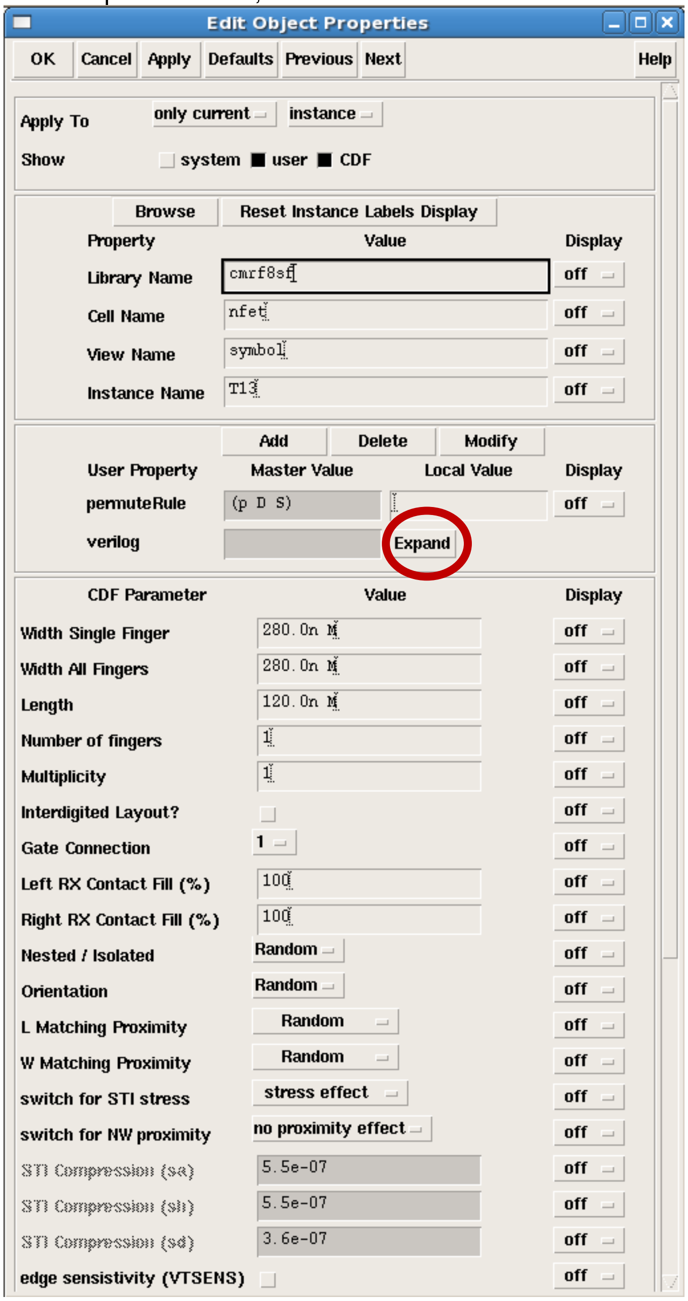
Step 1: Open the transistor property window. And press on "Add" which is circled in red.



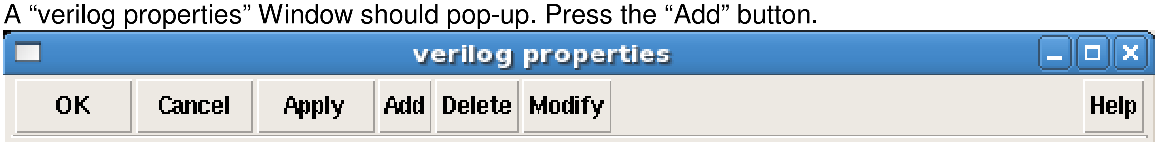
Step 2: A "Add Property" Window will pop-up. In the "Name" field, type "verilog". Change the "Type" field to "hierProp" and click on OK.



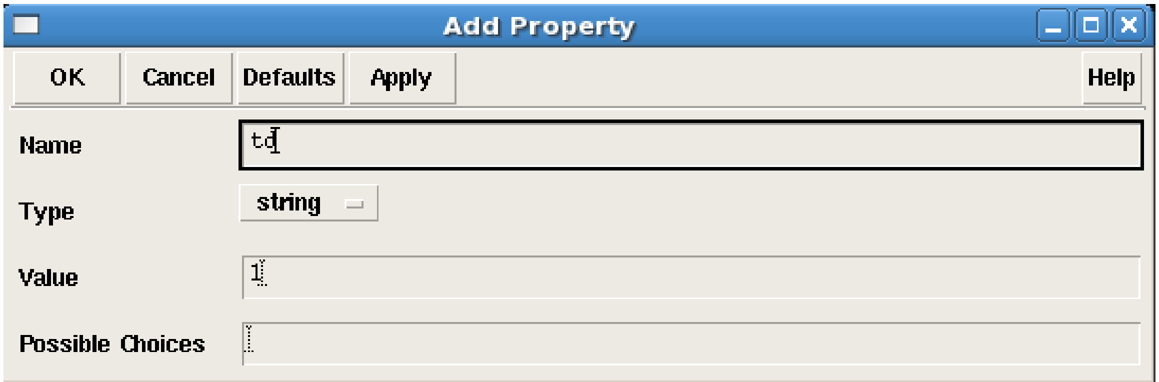
Step 3: Back in the "Edit Object Properties" Window, a new user property name "verilog" should appear. Press on the "Expand" button, circled in red.



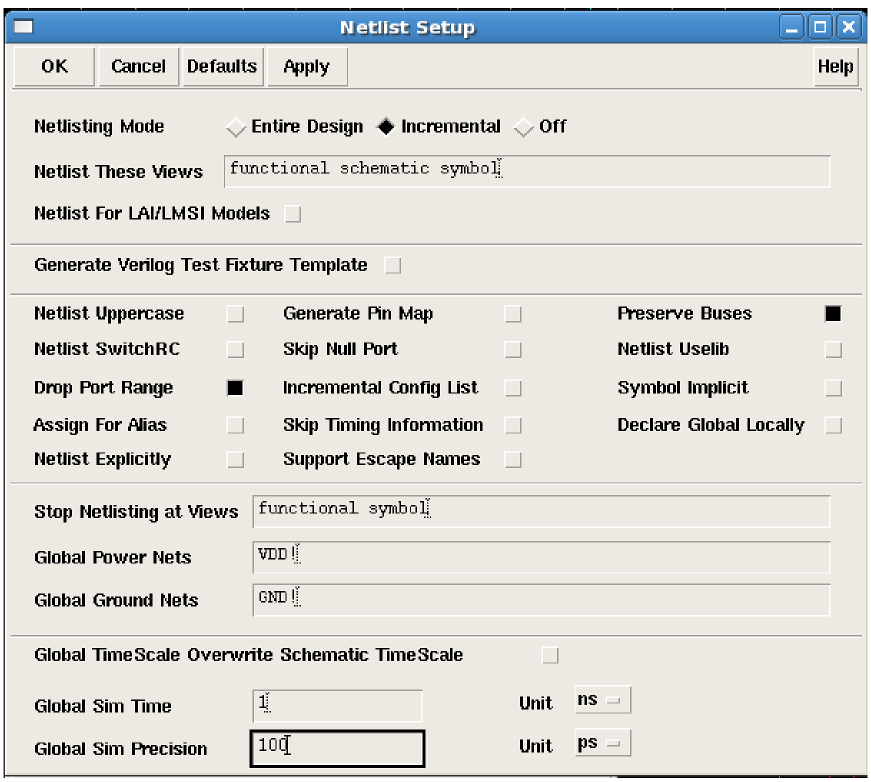
Step 4: A "verilog properties" Window should pop-up. Press the "Add" button.



<u>Step 5:</u> A new "Add Property" Window will pop-up. In the "Name" field type "td". In the "Type" field select type "string". In the Value field, type in the delay that is desired. In this example, a delay of 1 ns is used. Please round it off to the first decimal.



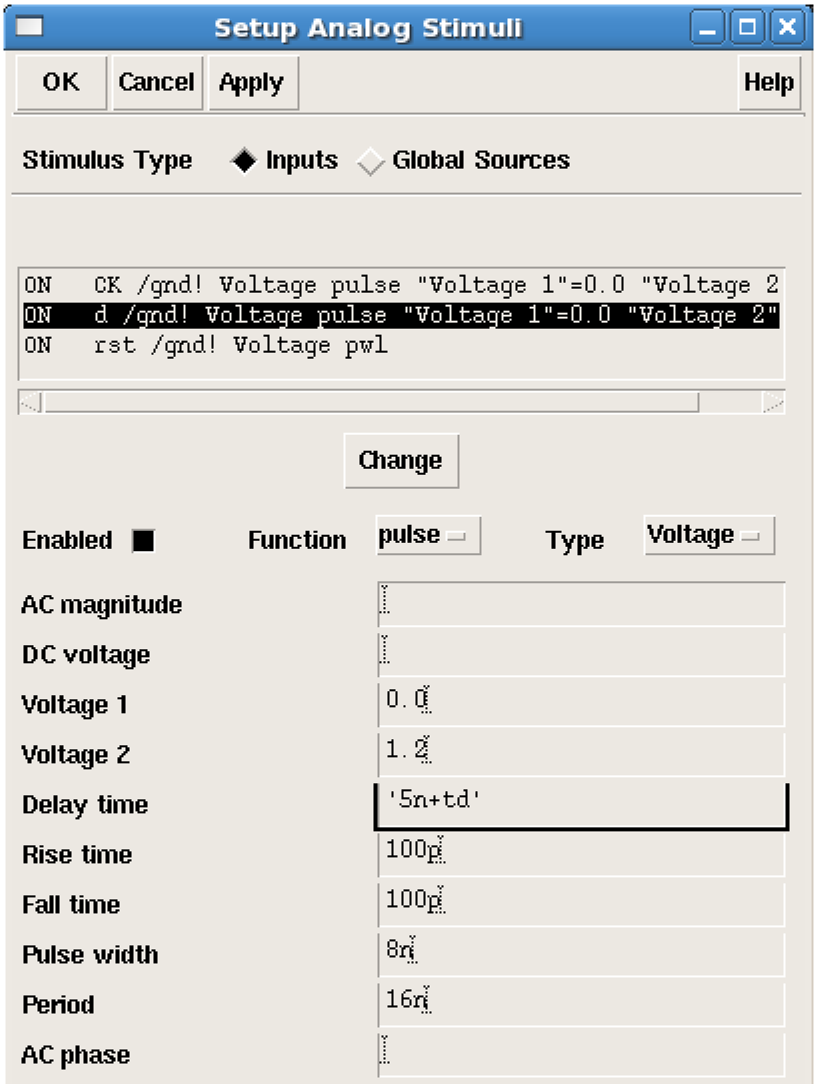
After adding all the delays, you need to modify the accuracy of the Verilog simulator so you will see the delay. To do this, you need to select ***Options->Netlist*** after you **initialize** your NC-Verilog environment. The "Netlist Setup" Window will pop-up. You need to change the Global Sim Precision from 1 ns to 100 ps before you simulate your design.



After modify your setting, start your simulation, and verify that you see a clk-to-q delay in your simulation.

Parametric Analysis Tutorial:

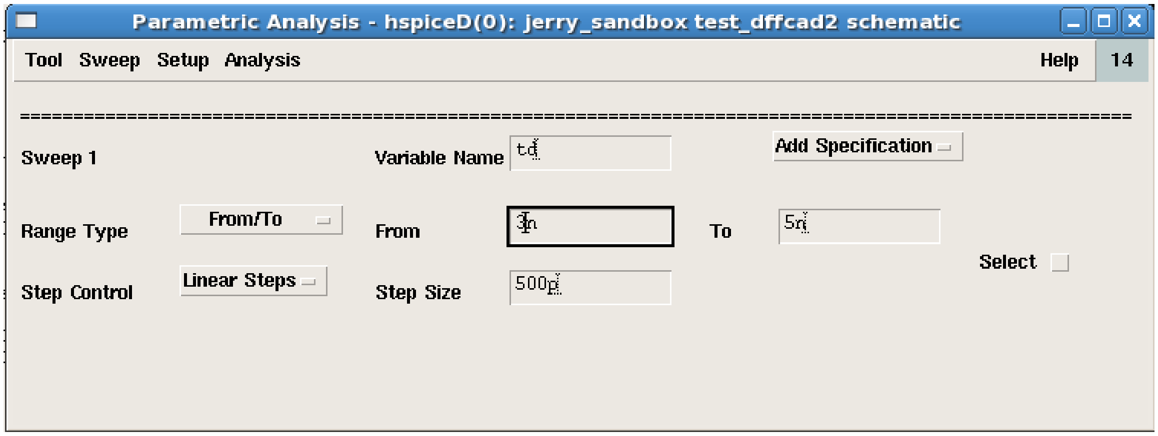
Parametric Analysis allows you to find your setup time and hold time with one simulation. Assuming that you are using a "pulse" as the stimulus of your input "d," you need to add a variable to your delay as shown in the following diagram:



After the stimulus has been setup open the Parametric Analysis window through the menu.

Tools->Parametric Analysis

The following window should open. Type in the variable name that you would like to sweep. Give it a starting point and stopping point and the desired step size. The numbers in the following figure are just an example.



To simulate in Parametric Mode, use the menu in the Parametric Analysis Window: ***Analysis->Start***.

When the simulation is all done, you should see your result from the sweep in the plotting area.