

S-Edit v13

Tutorial

Tanner EDA Division
Tanner Research, Inc.
825 South Myrtle Avenue
Monrovia, CA 91016-3424
Tel: (626) 471-9700

TABLE OF CONTENTS

1.	INTRODUCTION	3
1.1.	TYPOGRAPHIC CONVENTIONS	4
2.	INSTALLING THE TUTORIAL	5
3.	GETTING STARTED WITH S-EDIT	6
3.1.	VIEWING THE DESIGN	8
3.2.	NAVIGATING THE DESIGN HIERARCHY	9
3.3.	ELEMENTS OF A SYMBOL VIEW	11
3.4.	ELEMENTS OF A SCHEMATIC VIEW	15
4.	EVALUATED PROPERTIES	18
4.1.	ANNOTATE PORT PROPERTIES	20
5.	CALLBACKS	23
6.	SAVING DESIGNS	25
7.	CREATING A NEW CELL	26
7.1.	CREATING A NEW SCHEMATIC	26
7.1.1.	Placing Instances	26
7.1.2.	Making connections	29
7.1.3.	Placing Input and Output Ports	31
7.1.4.	Moving instances with rubberbanding	33
7.2.	CREATING A NEW SYMBOL	34
7.2.1.	Automatic symbol generation and update	34
7.2.2.	Editing the symbol	34
8.	CHECKING THE SCHEMATIC	36
9.	HIGHLIGHTING NETS	38
10.	SIMULATING THE DESIGN	40
10.1.	RUNNING SIMULATIONS	41
10.2.	PROBING VOLTAGES, CURRENTS, AND CHARGES	42
11.	VIEWING VOLTAGES, CURRENTS, AND CHARGES ON THE SCHEMATIC	47
12.	BUSSES AND ARRAYS	48
13.	CUSTOMIZING THE SETUP	50

RINGVCO TUTORIAL

1. Introduction

S-Edit is a schematic capture tool for use by circuit design engineers. In this tutorial you will learn about:

- ❑ Opening a design
- ❑ Viewing designs and navigating design hierarchy
- ❑ Elements of a symbol
- ❑ Elements of a schematic
- ❑ Saving designs
- ❑ Creating your own schematic and symbols
- ❑ Simulating Designs and probing results
- ❑ Highlighting nets

1.1. Typographic Conventions

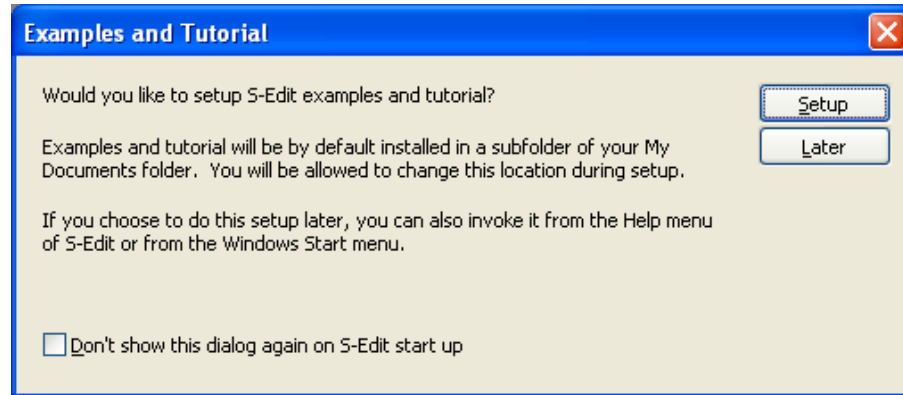
This section describes typographic conventions that are used in this document.

Bold Indicates elements that are a part of S-Edit, such as toolbars, menus, and buttons.

`Courier` Indicates elements that are part of the tutorial design, such as cell names, instance names, and properties.

2. Installing the tutorial

The first time you run S-Edit after installation, the following dialog appears, asking if you wish to Setup the examples and tutorial.



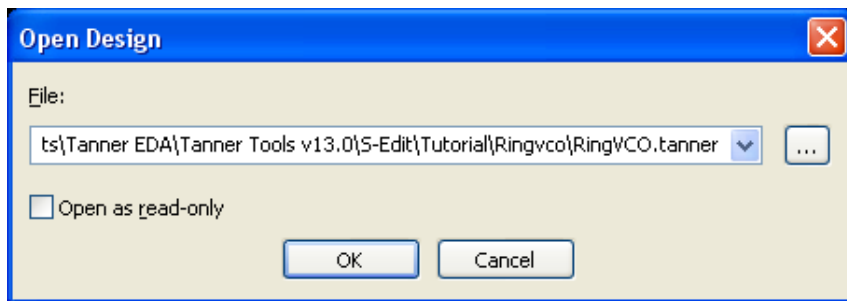
If you installed the tutorial when you first installed S-Edit, then the tutorial can be found in the location that you specified for the tutorial files. The default location for the tutorial is **My Documents\Tanner EDA\Tanner Tools v12.6\S-Edit\Tutorial**.

Tutorial and example files may be installed at any time by invoking **Help > Setup Examples and Tutorial** from S-Edit, or by invoking **Setup Examples and Tutorial** from the Windows start menu for S-Edit.

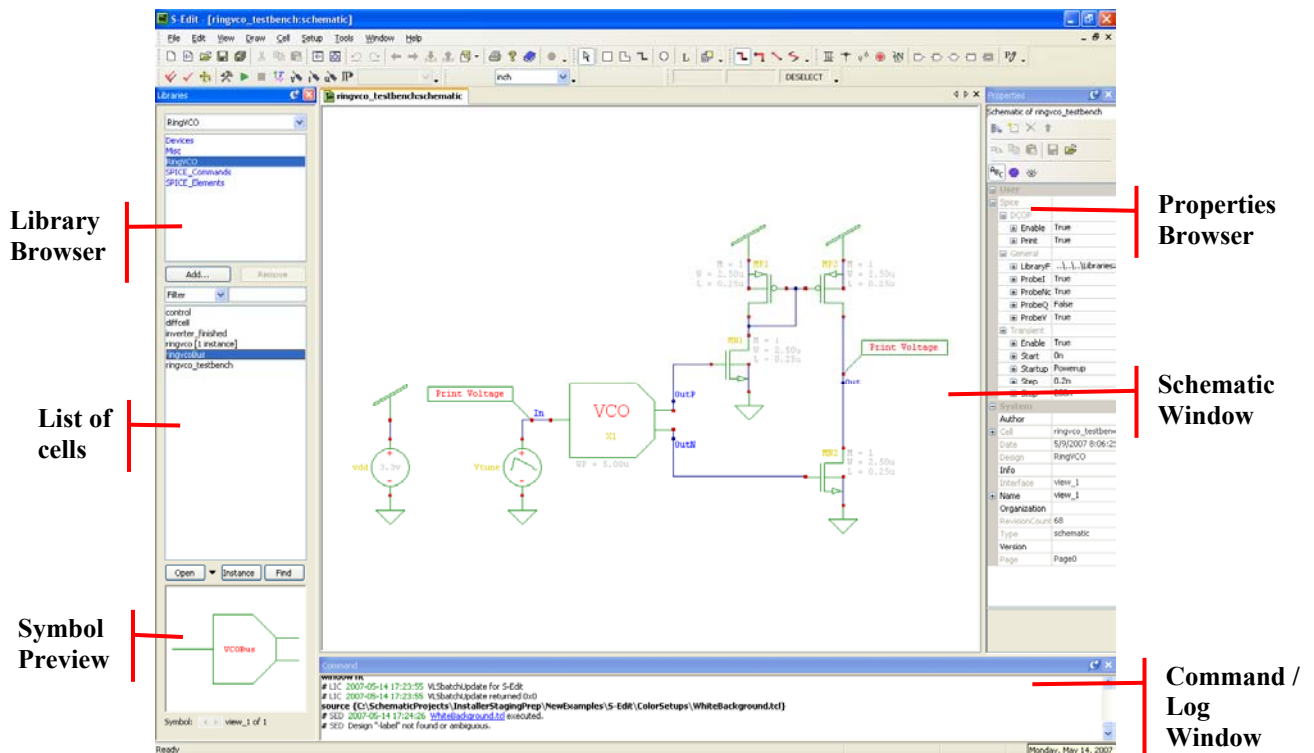
After running the tutorial, modified tutorial files can be replaced with a fresh copy of the tutorial files by invoking **Help > Setup Examples and Tutorial**, and choose **Repair**, when asked to select between **Modify**, **Repair**, and **Remove**.

3. Getting started with S-Edit

1. First, we need to open the design database for the tutorial. Invoke **File > Open > Open Design...**, press the Browse button, (...), in the **Open Design** dialog, and browse to the **RingVCO.tanner** file in the **RingVCO** folder of the tutorial folder. The default location for the tutorial is **My Documents\Tanner EDA\Tanner Tools v13.0\S-Edit\Tutorial**, although you may have installed it in another location. Press OK in the **Open Design** dialog.



2. The RingVCO tutorial will load into S-Edit, and look like this. On the left side of the application window, we have the **Library Browser**, on the bottom, we have the **Command/Log window**, and on the right, we have the **Properties Browser**. In the center is the toplevel schematic of our design.



3. This project consists of a design called Ringvco, which references several libraries. The libraries referenced are Devices, Misc, SPICE_Commands, and SPICE_Elements.

The design is shown in black, and libraries are shown in blue in the list of libraries in the top pane of the **Library Browser**.

3.1. Viewing the design

4. Let's examine the **Library Browser**, located on the left side of the application.

- ❑ Select the design or one of the libraries in the top pane, and you can see the list of cells contained in that library. You can select multiple libraries in the top pane and see all the cells in the selected libraries. Use Shift+Click or Ctrl+Click to select multiple libraries.
- ❑ Select all the libraries in the design in the top pane. You can filter the list of cells shown in the cell list by typing a substring match in the filter field. Type MOS into the filter field, and see now only those cells containing a substring MOS are shown in the list of cells.
- ❑ Click on the Filter dropdown in the library browser and change it to **Top-level**. Top-level filter will show only those cells that are not instanced anywhere in the design. You can type a substring match into the filter to further filter the results.
- ❑ Click on the filter dropdown and select **Leaves**. Leaves are cells that have no instances in their schematic. You can type a substring match here too, to further filter results. Return the Filter dropdown to **Filter** and clear out the filter substring.
- ❑ Select different cells in the cell list, and see a preview of their symbol in the **Preview Pane** below the cell list.
- ❑ You can right-click anywhere on the **Library Browser** and toggle on/off display of the **Preview Pane**.

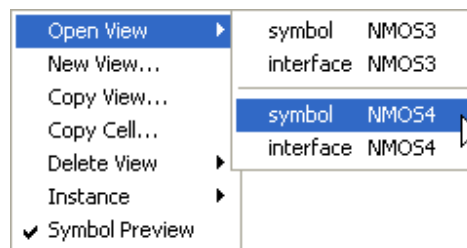
5. Let's pan and zoom around the design.

- ❑ Use the +/- keys to zoom in and out of the design.
- ❑ Scroll the mouse wheel to zoom in and out of the design, centered at the cursor.
- ❑ Use the **arrow** keys to pan up/down/left/right.
- ❑ Press the **Z** key then press the left mouse button and drag over a region to zoom into that region.
- ❑ Use the **Home** key to fit the contents to the window.

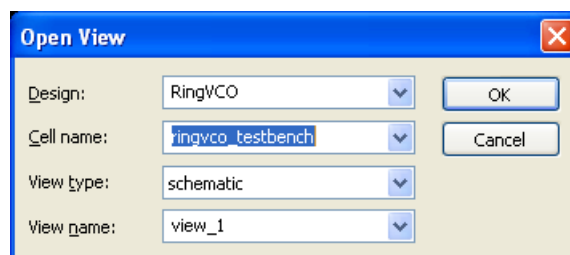
3.2. Navigating the design hierarchy





6. Now let's navigate around the design. There are several ways to open a schematic or symbol view of a cell.
7. Double click on the symbol in the center labeled VCO to open the schematic of RingVCO.

- ❑ Double click on one of the DiffCell symbols to view its schematic.
- ❑ You can also open symbol or schematic views by right clicking on the desired cell in the cell list of the **Library Browser**, selecting **Open View**, then choosing the view you want to open. The view type is indicated first, followed by the view name. The view name is only displayed if there are multiple views of the same type. Open the symbol view of the 4 terminal NMOS cell using this method. If you don't see NMOS in the cell list, then you need to select the Devices library in the list of libraries. You can also select all the libraries in the list to see all cells.



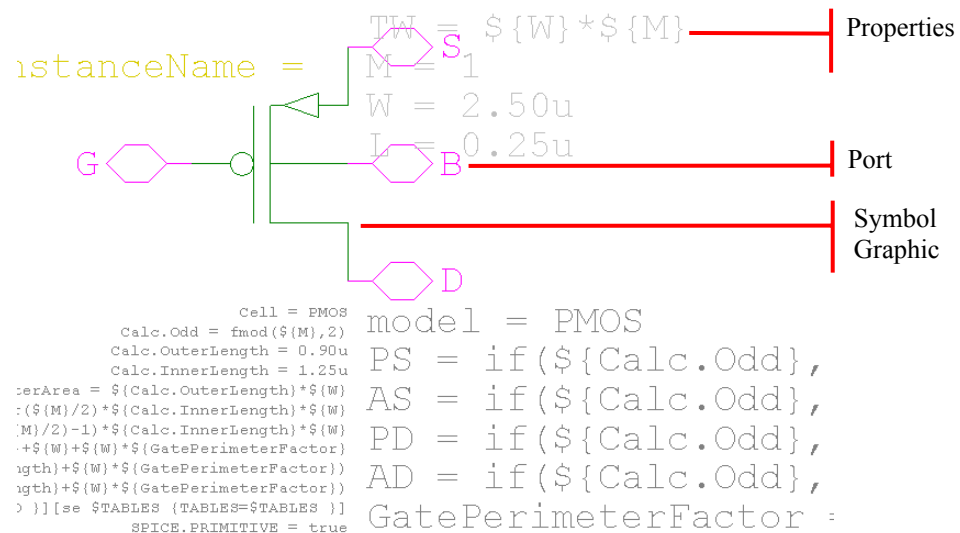
- ❑ Views can also be opened by double clicking on a cell in the cell list. The view opened (symbol, interface, or schematic) will be of the same type as the last opened view. Double click on cell NFET in the list of cells to open the symbol view of NFET. If multiple views exist, a list will be displayed to select the desired view.
- ❑ You can also select a cell and press the **Open** button in the Library browser to open a cell. This behaves the same way as double click. Select the DiffCell cell in the cell list, and press the **Open** button to open its symbol view.
- ❑ Pressing the drop down arrow next to the Open button lets you pick the specific view of the selected cell you wish to open. Open the schematic view of the control cell in this manner.
- ❑ Lastly, views may be opened using the **Cell > Open View** command. Use this method to open the schematic view of cell RingVCO_TestBench in this manner.



-
- ❑ When opening a view, you can force the new view to open in a new window by holding the **Ctrl** key down when opening the new view.
 - ❑ You can use the **View Symbol** () and **View Schematic** () toolbar buttons to open the Symbol or Schematic view that corresponds to the currently active window. Close all windows, and open the schematic view of DiffCell. Press the **View Symbol** button to see the symbol for DiffCell. Pressing the “?” key will toggle between symbol and schematic views of a cell. Close all open views.
 - ❑ The **Back** and **Forward** buttons ( ) on the Standard toolbar can be used to navigate to previous and next views. The forward button is only enabled after backing up using the back button. Open the Schematic view of RingVCO_TestBench. Double click on the symbol for RingVCO to open the schematic for RingVCO. Now double click on one of the symbols of DiffCell to open its schematic. Now use the **Back** and **Forward** buttons to navigate back and forward to previous views and to next views. Now close all views.

3.3. Elements of a Symbol View

8. Now let's examine a symbol. Open the symbol view of cell PMOS, a PMOS transistor, using any method.



- A symbol view contains the following elements:

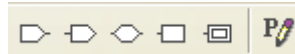
- **Symbol Graphics:** Symbol graphics comprise the graphical image of the symbol. This is the image that is seen when the symbol is instantiated. Symbol graphics can be boxes, polygons, paths, or circles, and are shown in green above.
- **Labels:** Text labels can be added to a symbol, and are visible when the symbol is instantiated.
- **Ports:** Ports define the connection points that can be made to the symbol when the symbol is instantiated. In the figure above, these are the S, D, G, B shown in purple. When instantiated, ports do not show their text, but appear as an open circle “hot sport” for connecting a wire.
- **Properties:** Properties are name-value pairs that are usually used to describe some characteristic of a device, such as a transistor length, width, or Source/Drain areas and perimeters. Properties can also be used for other purposes, such as controlling the spice statements written for a device. Properties on a symbol provide the default values when that symbol is instantiated, but may be overridden on an instance-by-instance basis.

- The toolbars used for creating the objects on a symbol are:

- The Draw toolbar for creating symbol graphics and labels.

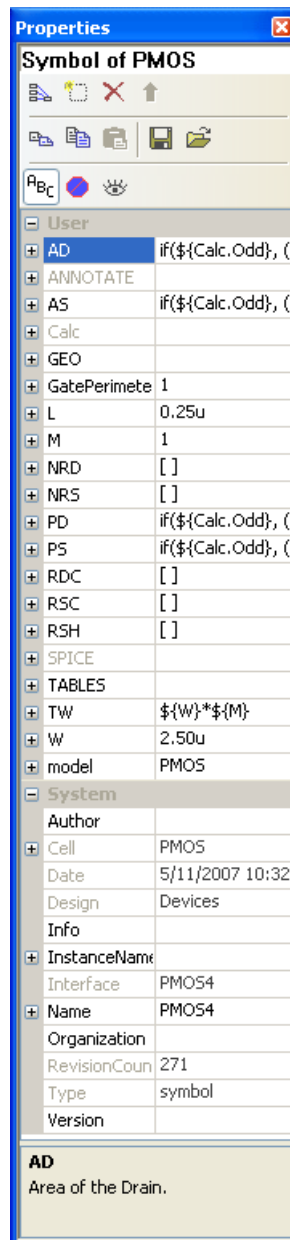


- The Electrical toolbar for creating Ports and Properties. Ports may be of type In, Out, In/Out, Other, and Global.

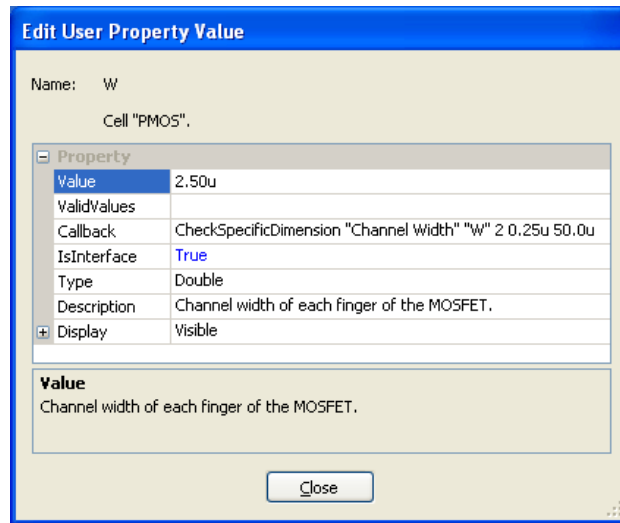


9. Lets examine symbol properties in more detail:

- Properties for the symbol are visible on the symbol itself, and also in the Properties browser. User properties such as Drain Area (AD), Source Area (AS), Length (L) etc are shown in the User area of the Properties Browser. System properties such as the Design/Library from which the symbol comes (Design), the cell name (Cell), etc are shown in the System area of the properties Browser.



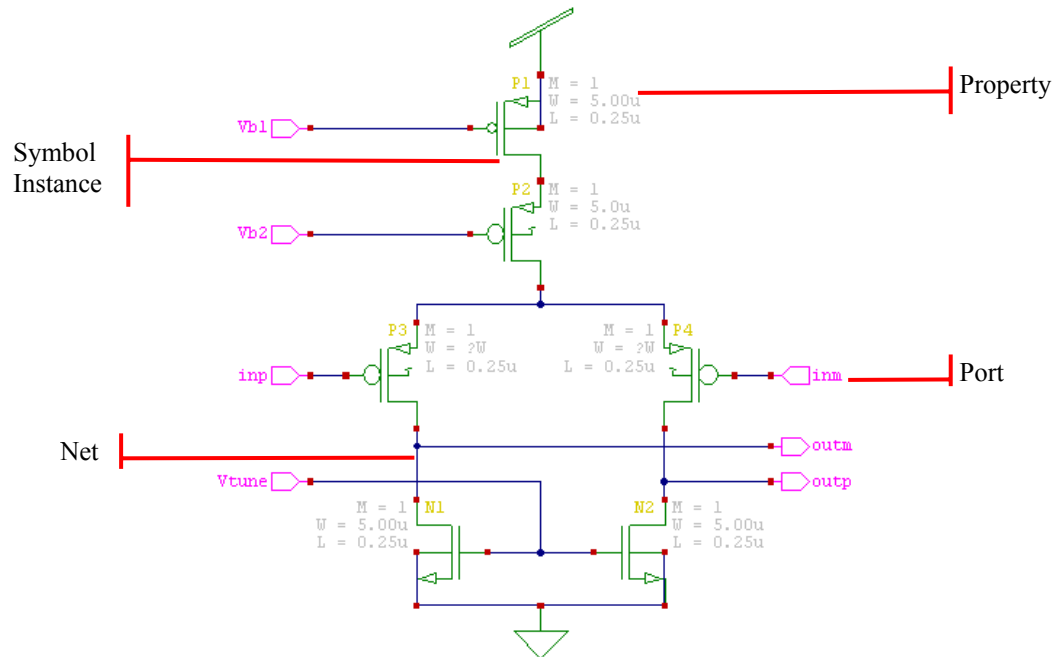
-
- ❑ The properties on a symbol are default properties that are used when the symbol is instantiated.
 - ❑ You can easily change the value of any property by placing the cursor in the value entry of the property grid and entering a new value. Properties can also be added using the **Add** button on the toolbar, and deleted using the **Delete** button.
 - ❑ Properties can be an explicit value, can reference another value, and can be an expression. For example, the value of L is 0.25u and the value of W is 2.5u. Property TW, the total width of the PMOS is an expression which references the value of properties M and W using the “\$” as a reference. When the PMOS is instantiated, the value of TW will get resolved to $TW = M * W$. If the values of M and W are not locally overridden on the instance, then $TW = 1 * 2.5u = 2.5u$.
 - ❑ Properties AD, AS, PD and PS use expressions with the “if (expr, expr, expr)” function to calculate the source and drain areas and perimeters based on the length “L”, the width “W”, the multiplicity “M”, and the GatePerimeterFactor. Note that Calc.Odd and Calc.OuterArea are sub-properties of property Calc. Sub-properties are referenced using a period as separator between levels.
 - ❑ A property on a symbol can be made visible, hidden, or only the value visible when the symbol is instantiated. Click on the (+) next to the AD property, and see its **Display** value is set to Hidden. Click on the grid on the word **Hidden** and Click the drop down control to see the possible choices **Hidden**, **Visible**, and **ValueOnly**. Close the properties for AD by pressing the (+) next to AD, and inspect the Display property for L, see its Display property is Visible. Hidden and Visible/Value Only properties are rendered in slightly different colors on the screen to easily distinguish which are visible and which are hidden. The Display property specifies how the property is displayed when in evaluated mode. The Display.WhenNotEvaluated property sets how the property is to be displayed when not in evaluated mode.
 - ❑ Inspecting the properties on the symbol for PMOS, we see that W, L, and M have their Display set to Visible, ANNOTATE.Drain, ANNOTATE.Gate, and ANNOTATE.Source have their Display set to ValueOnly, and all other properties have their display set to hidden.
 - ❑ Properties may also be edited in place, without the properties grid. Double click on property W to get the following dialog to edit its value. Press Cancel to dismiss the dialog.



- ❑ Properties may also be moved to different locations. Right click on a property to select it, and then hold the middle mouse button to move the property.

3.4. Elements of a Schematic View

10. Now let's examine a schematic. Open the schematic view of cell **DiffCell**.



□ A schematic view contains the following elements:

- **Instances of symbols:** Instances of symbols refer to a particular symbol in a cell. A schematic may contain many instances of the same symbol or of different symbols. Instances contain graphics, which provide the illustration of the symbol, and ports, which provide connection points for attaching nets. Ports on the symbol and ports on the instance are different in that ports on the symbol are shown as their port name, whereas ports on the instance do not show the port name but are simply a “hot spot” for connecting a net.
- **Nets:** A net is a wiring connection between two or more instance ports. A net can be a single wire, or can be a collection of wires called a bus or a bundle.
- **Properties:** Properties are name-value pairs that are usually used to describe some characteristic of a device, such as a transistor length, width, or Source/Drain areas and perimeters. Properties can be put on an instance to override symbol values, or new properties can be created on an instance.
- **Ports:** Ports on the schematic correspond to the ports on the symbol. Ports on the schematic define how connections made to ports on symbol instances connect to nets on the schematic for that symbol.

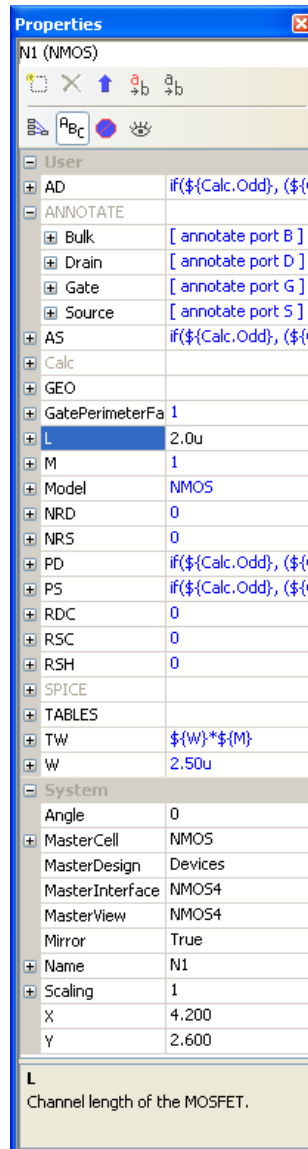
- **Annotation graphics:** Annotation graphics are non-electrical objects such as boxes, polygons, paths, and labels used to add comment or illustrations to the schematic.

11. Now let's examine the contents of **DiffCell**:

- ❑ Annotated in square brackets in the list of cells in the Library browser you can see what cells and how many of each cell are instantiated in the schematic. Make sure both Ringvco and Devices are selected in the library list. Cell DiffCell contains 1 instance of Gnd, 2 instances of NMOS, 4 instances of PMOS, and 1 instance of Vdd.

12. Let's examine some instance properties in more detail:



- ❑ Select instance N1 of cell NMOS located near the bottom of the schematic. You can see all the properties of the instance N1 in the properties browser.



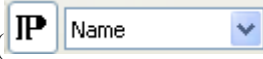
-
- ❑ Notice that only those properties that were designated as Visible or Value only on the symbol of NMOS are visible on the instance N1 in the schematic window.
 - ❑ Values may be overridden on an instance by instance basis. Property values that are only on an instance and not on the symbol, or are overrides over symbol values are shown in black in the properties browser, property values that are inherited are shown in blue. For instance N1 notice that property L has a value of 2.0u, which overrides the value of 0.25u on the symbol.
 - ❑ You can revert an overridden property value back to its symbol value by selecting that property and pressing the **Reset** button (blue up-arrow) on the property browser. Select the L property of instance N1 in the property browser, and press the reset button. See the value changes from 2.0u (and black, meaning local override) to 0.25u (and blue, meaning inherited). Press Ctrl-Z to undo the change.
 - ❑ You can modify the values of multiple instances at once. Select the instances N1 and N2, the bottom two NMOS instances in the schematic. The property browser displays those values that are the same for all selected instances, and leaves blank those property values that are different for the selected instances. Change the value of L to 10u. Select each instance individually and see that each now has the new values. Change the values back to 2.0u.
 - ❑ You can use the Library browser to easily select all instances of symbol and then change properties on all of them. Select an instance in the schematic. Note that the selection in the list of cells in the Library browser is updated to correspond to the selected symbol. Select an instance of cell PMOS on the schematic or select PMOS directly in the list of cells. Now press the **Find** button and all instances of cell PMOS will be selected in the schematic. You can now change properties on all the selected symbols, or do any other operation on the selections.
 - ❑ Properties may also be edited in place, without the properties grid. Hold down the Ctrl key, and double click on a property a dialog to edit its value. Press Cancel to dismiss the dialog.
 - ❑ Properties may also be moved to different locations. Hold down the Ctrl key and right click on a property to select it, and then hold the middle mouse button to move the property.
 - ❑ Close all views, and open the schematic view of RingVCO_TestBench.

4. Evaluated Properties

13. We saw in **Elements of a Symbol View**, properties can be an explicit value, or can be an expression. We will now explore the full capability of evaluated properties. Expressions support standard mathematical operators -, *, /, **, as well as standard functions sin(), cos(), etc. Expressions can also reference the values of other properties using the following prefix operators:

- %T (or %{T}) references the name of the node connected by the terminal T. When viewing a cell "in context" (see **Probing voltages**, currents, and charges) this name is the hierarchical name of the net.
 - \$P corresponds explicitly to another property on the same instance, or in symbol view to another property on the symbol. For example, $TW = \$W * \M , references the width and the multiplicity of the same device. This option corresponds to the Cadence iPar() function.
 - ?P references the parent cell's property, but only looks up one level. As usual, overridden properties on the instance of the parent have higher priority than the default value on the symbol. This option corresponds to the Cadence pPar() function.
 - @P (or @{P}) in a property value references the *highest-level* definition of P. If we have a cell TOP which contains an instance of MID, which contains an instance of BOT, then the priority order of a property P "inside" BOT is then (high to low): global, $MID_{INST.P}$, $MID_{SYM.P}$, $BOT_{INST.P}$, $BOT_{SYM.P}$.
-
- ❑ We have already seen the \$ prefix used to compute the value of the TW property as $TW = \$\{W\} * \$\{M\}$, in the symbol view of the 4 terminal PMOS.
 - ❑ Open the schematic view of DiffCell, Select P4 (the rightmost instance), and see in the properties browser that the value of W is ?W. To find the value of ?W we look up one level, i.e. on the instance of DiffCell to find the value of W. We see that all the instances of DiffCell have the same value of W, which is ?WP. Looking up one level at the instance of RingVCO, we see it as a property $WP = 5.00u$. So the value of W on P4 will be 5.00u.
 - ❑ To see the values of evaluated properties we must navigate through the hierarchy "in context". Viewing a schematic "in context" refers to viewing a specific hierarchical path of that schematic. Since evaluated properties refer to other properties on a hierarchical path, we must have a hierarchical path in order to evaluate properties.
 - ❑ To navigate the hierarchy in context, first select an instance, then press the **Push into context** button () on the toolbar. To pop back up the hierarchy, press the **Pop context** button, (). Select the VCO symbol and push into the VCO. You can also push into a cell by double clicking on its symbol with the cursor in selection mode. Push into the third from the left DiffCell instance, Xa3.
 - ❑ When you push in context into a cell, S-Edit will display the hierarchical path on the title bar of the schematic window. In our example we started in cell RingVCO_TestBench, pushed into instance X1 of cell RingVCO, and then into Xa3

of cell DiffCell, so the titlebar of the schematic window will display **RingVCO_TestBench/X1/Xa3**.

- ❑ To display the values of evaluated properties, press the **Display Evaluated Properties** button (). We can now see the evaluated value of W on P4 on the schematic view as 5u. Also, properties in the property browser display their evaluated values. Values are shown in green in the property browser to indicate we are showing evaluated values.
- ❑ Now select P1 (topmost instance of PMOS). See the values of AD, AS, PD, PS display their evaluated values.
- ❑ If you Display Evaluated Properties without a hierarchical path, property values will display as <property is not evaluated>.

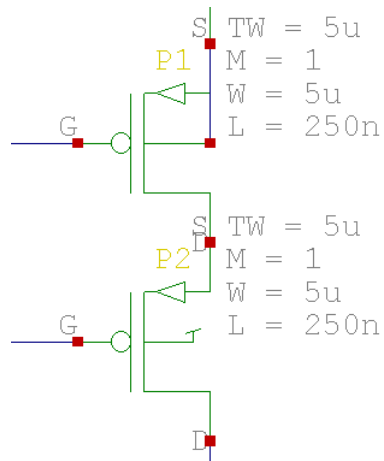
4.1. Annotate Port Properties

14. Using evaluated properties we can display several values of interest at the ports of an instance. These include:

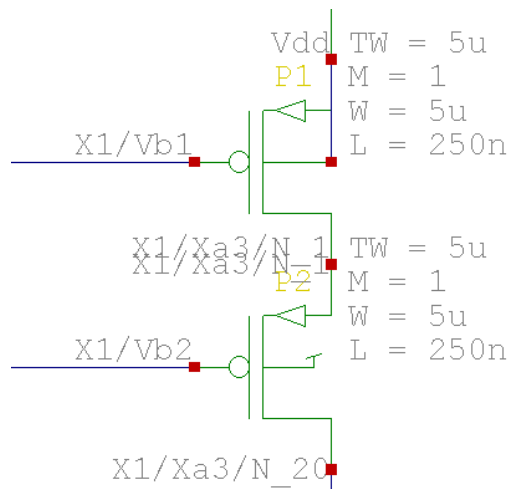
- Port name
- Net name
- DC Voltage
- DC Current
- DC Charge

The first two of these are always available to display, the next three depend on values being present from a DC simulation run.

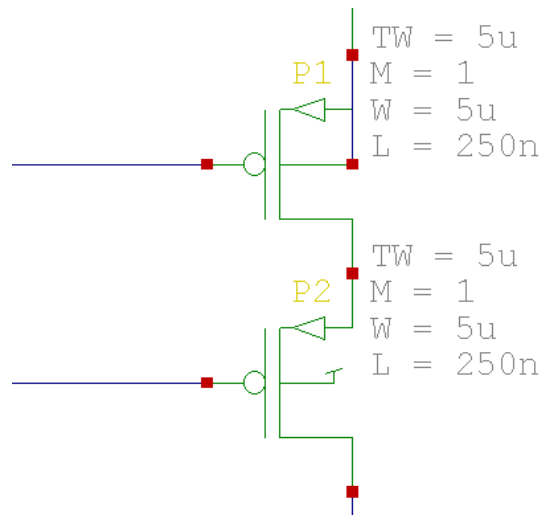
- ❑ We saw on the symbol view of the 3 or 4 terminal PMOS , there were some special properties on the symbol of our MOSFET:
 - ANNOTATE.Drain = [annotate port D]
 - ANNOTATE.Gate = [annotate port G]
 - ANNOTATE.Source = [annotate port S]
 - ANNOTATE.Bulk = [annotate port B] (4 terminal PMOS only)
- ❑ The key elements of these properties are i) only the property value is important, the property name can be anything, and, ii) the property value contains the string **annotate port *portname*** in square brackets, where *portname* is the name of a port on the symbol. Open the symbol view of the 4 terminal PMOS and see these properties.
- ❑ These annotate properties, when placed on a symbol (or on individual instances), can display the port name, net name, DC voltage, DC current, DC charge, or nothing. Which of these values is displayed is controlled by the **Display Evaluated Properties** button and dropdown on the Simulation toolbar.
- ❑ Next open RingVCO_TestBench, and run **Tools > Design Checks**. This is needed to extract the netlist, we will discuss more on design checks later. Open the VCO in context by double clicking on it, then open Xa3 in context. Press the **Display Evaluated Properties** button (selecting the **Annotate Port** dropdown as **Name** if not already selected). See the annotate properties now display the port names, S, G, D (B is not shown because it's Display property is set to hidden):



- Now select **Net** in the **Display Evaluated Properties** dropdown to make the annotate properties display the hierarchical net names, Vdd, X1/Vb1, X1/Vb2, X1/Xa3/N_1, and X1/Xa3/N_20:



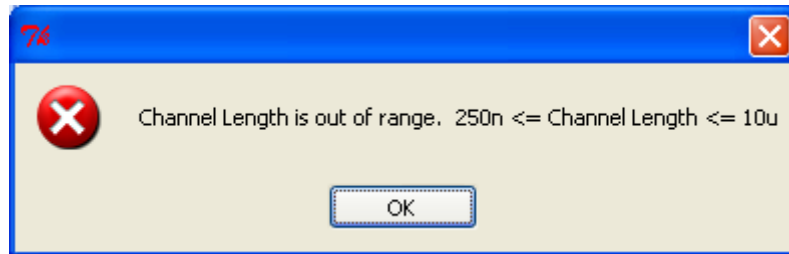
- You can also choose **None** to display nothing for the annotate properties.



5. Callbacks

15. Callbacks provide the ability to call a tcl command upon changing a property value. The callback command is usually a user written function. Typical uses of callbacks are to perform validity checking of the input or to modify other properties that should change in order to maintain consistency with the modified property.

- ❑ Open the symbol view of the 4 terminal PMOS, and in the property grid, expand property “L” and see that it has a callback function named `CheckSpecificDimension`. The function `CheckSpecificDimension` checks that the value that the user has entered is in a specified range and is snapped to a specified grid, and returns an error message these conditions are not met.
- ❑ Open the schematic view of cell `DiffCell`, and select P1 (near the top of the schematic). Note that it has values `L=0.25u`. Now change the value of L to 0.25 (remove the u) and see the following message:



This dialog is coming from the callback function.

- ❑ Functions `CheckSpecificDimension` are user written functions listed below:

```
proc CheckSpecificDimension { sDimensionName sPropertyName
    nMinMaxBoth sMinimumDimension sMaximumDimension
    { sProcName "" } } {
    global dMfgGrid
    global gdTOLERANCE
    set dMinimumDimension [ stod $sMinimumDimension ]
    set dMaximumDimension [ stod $sMaximumDimension ]
    set dDimension [ stod [ property get $sPropertyName ] ]

    set dSnappedDimension [ gNearestMultiple $dDimension
        $dMfgGrid ]
    if { [expr abs($dDimension-$dSnappedDimension) ] >
        $gdTOLERANCE } {
        set sMsg "$sDimensionName is not on the
            manufacturing grid. $sDimensionName has been
            snapped to [dtos $dSnappedDimension]"
        tk_messageBox -message $sMsg -type ok -icon error

        property set $sPropertyName -value [dtos
            $dSnappedDimension ]
        set dDimension $dSnappedDimension
    }
}
```



```

set sMsg ""
if { ($nMinMaxBoth == 0) && ($dDimension <
$dMinimumDimension) } {
    set sMsg "$sDimensionName is too small. [dtos
    $dMinimumDimension] <= $sDimensionName"
} elseif { ($nMinMaxBoth == 1) && ($dDimension >
$dMaximumDimension) } {
    set sMsg "$sDimensionName is too big.
    $sDimensionName <= [dtos $dMaximumDimension]"
} elseif { ($nMinMaxBoth == 2) && (($dDimension <
$dMinimumDimension) || ($dDimension >
$dMaximumDimension)) } {
    set sMsg "$sDimensionName is out of range.
    [dtos $dMinimumDimension] <= $sDimensionName
    <= [dtos $dMaximumDimension]"
}
if { $sMsg != "" } {
    tk_messageBox -message $sMsg -type ok -icon error
    return $sMsg
}

if { $sProcName != "" } {
    eval $sProcName
}
}

```

- ❑ The callback functions must be defined in S-Edit before they can be called. This is done either by dragging the file containing the callbacks into the command window, or by placing the file in one of the folder locations from which scripts get automatically loaded:

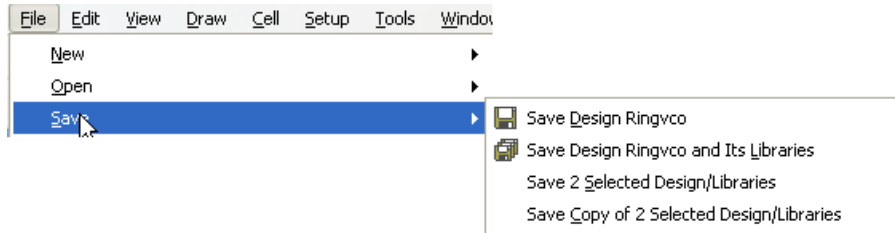
Scripts placed in a folder `scripts/open.design` in the design folder will get automatically loaded when the design is opened. We have placed the file `GeneralCallbacks.tcl` in this location in the Devices library.

Other locations from which scripts are automatically loaded are:

To load script when any design is opened, place script in:	C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\open.design
To load script when S-Edit is started, place script in:	C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\startup
To load script when S-Edit is shutdown, place script in:	C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\shutdown

6. Saving Designs

16. Before we proceed with the tutorial, let's learn how to save the design, so we can save the modifications we will be making in the next section. The **File > Save** menu is shown below:



There are four ways of saving a file:

- ❑ **File > Save > Save Design *DesignName*** saves modifications to the design, but does not save modifications to any libraries.
- ❑ **File > Save > Save Design *DesignName* and Its Libraries** saves modifications to the design and all libraries.
- ❑ **File > Save > Save # Selected Design/Libraries** saves those Design/Libraries that are selected in the list of libraries in the Library browser.
- ❑ The **Save Copy of ...** command saves a copy of the selected Design and Libraries in the Library Browser to a new location on disk. The Save Copy of ... command is not the same as a Save As command in that after the command is issued, the application is still editing the original design, not the newly saved design.

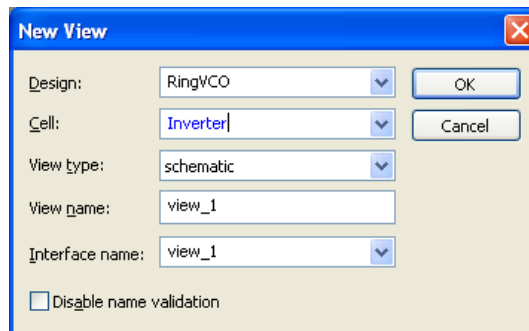
7. Creating a New Cell

7.1. Creating a new schematic

17. Now let's learn how to create our own schematic. If you wish to skip the steps to create the inverter cell, we have provided a completed cell called `InverterFinished`.

- ❑ First create a new cell called `Inverter`. Invoke **Cell > New View**. We want to create the new cell inside the design `Ringvco`, so enter the following into the **New View** dialog:

Design:	<code>Ringvco</code>
Cell:	<code>Inverter</code>
View type:	<code>schematic</code>
View name:	<code>view_1</code>
Interface name:	<code>view_1</code>

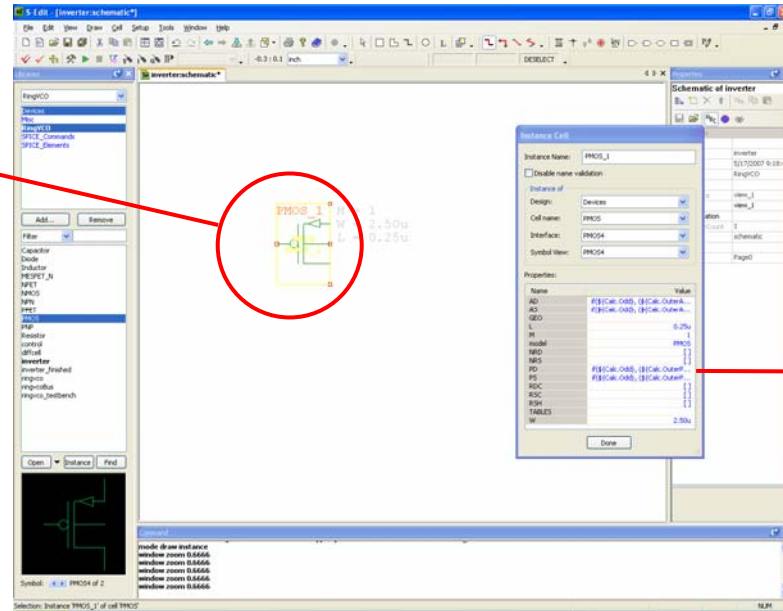


- ❑ After pressing OK, you will get an empty drawing area for a new schematic view.

7.1.1. Placing Instances

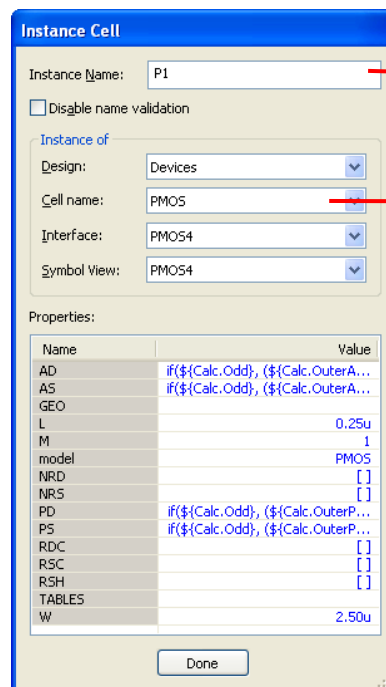
- ❑ Select `PMOS` in the cell list from the `Devices` library, and press the **Instance** button below the cell list. The **Instance Cell** dialog will appear, as shown below. You can zoom in and out using the `+/-` keys or the mouse scroll wheel, or pan using the arrow keys, before placing the instance.
- ❑ In the **Instance Cell** dialog, you can change the instance name or properties of the cell you are about to instance. Changing properties will make local overrides over the symbol default values. Change the instance name to `P1`. Make sure the `Interface` and `Symbol View` are `PMOS4`, for the 4-terminal `PMOS`.
- ❑ Place an instance of `PMOS` by clicking on the schematic drawing window.

Drag instance with mouse, click to place.



Change property values before placing an instance using the Instance Cell dialog.

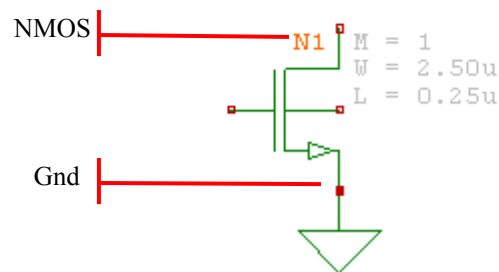
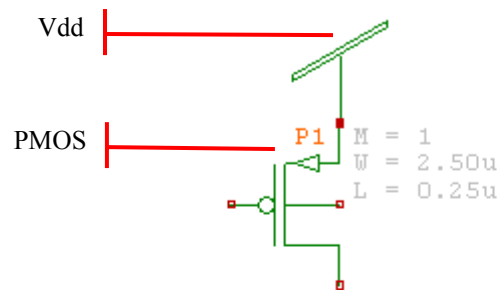
- Additional instances of PMOS can be placed by continuing to click the mouse to place instances. Instances can be flipped horizontally or vertically prior to placement by pressing the “H” or “V” key and may be rotated prior to placement by pressing the “R” key. Properties for each instance can be modified in the **Instance Cell** dialog prior to each placement. In the schematic for the inverter, we only need one PMOS, so we’ll proceed to change the cell we are instantiating.
- You can change the instance name by entering a new **Instance Name** in the Instance Cell dialog. You can also change the cell being instantiated by selecting a new **Cell name** in the Instance Cell dialog.




Change instance name.

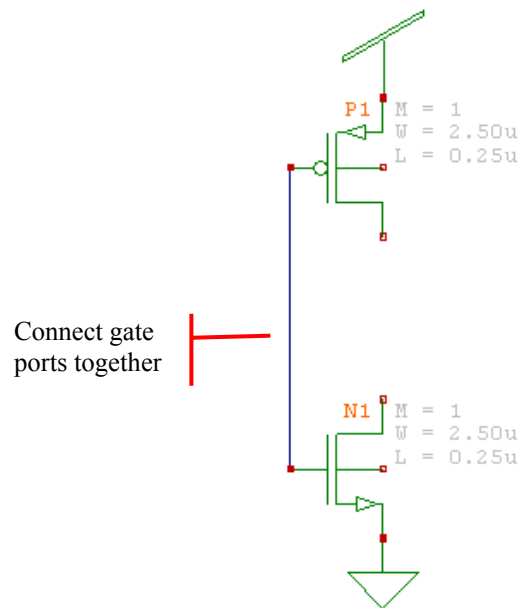
Choose a new cell to instance.

- ❑ Select cell NMOS in the Instance Cell dialog, change the instance name to N1, make sure the Interface and Symbol View are NMOS4, and place the NMOS below the PMOS by clicking on the schematic view at the location you wish to place the instance.
- ❑ Now place an instance of the Gnd connected to the bottom (source) port of the NMOS and an instance of Vdd connected to the top (source) port of the PMOS. The Gnd and Vdd symbols are in the Misc library.
- ❑ Notice that ports on symbols are displayed with a unfilled red box when they are not connected. When connected either directly to another symbol, or with a wire, the box becomes filled, indicating that a connection has been made.
- ❑ Exit instance mode by either pressing the right mouse button, pressing the ESC key, or pressing the Done button on the Instance Cell dialog.
- ❑ The placed instances should appear as follows:

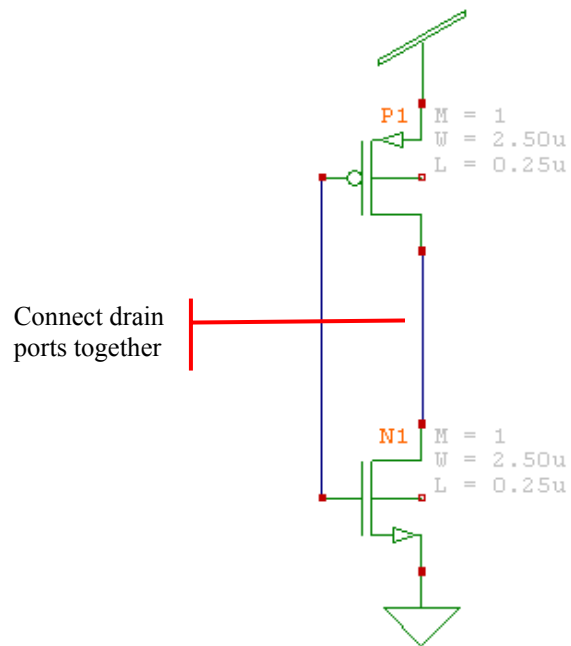


7.1.2. Making connections

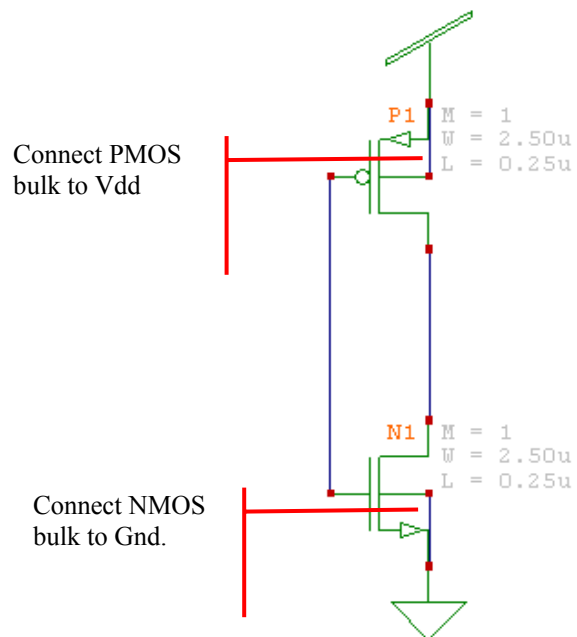
- ❑ Connect the gate (G) ports of the two MOSFETs together by drawing a wire from one to the other. To draw a wire, select the Wire drawing button on the toolbar, (), then click the left mouse button to place the first vertex, subsequent clicks with the left mouse button will place additional vertices. Clicking the right mouse button ends the wire without placing a vertex, and double clicking the left mouse button will end the wire with placing a vertex at the double click location. Pressing the ESC key will abandon the entire wire.
- ❑ Unconnected symbol ports and unconnected wire ends are shown as open circles. When a wire end is properly connected to a symbol port, the open circles disappear, indicating that they are properly connected. Similarly for two ports or two wires connecting directly to each other.
- ❑ Now you try connecting the two gate ports together. Click the left mouse button on the open circle of the gate port of the PMOS symbol, drag the mouse down to draw the wire, click the left mouse button on the open circle of the NMOS symbol, then click the right mouse button to end the wire.




- Now connect the two drains together.

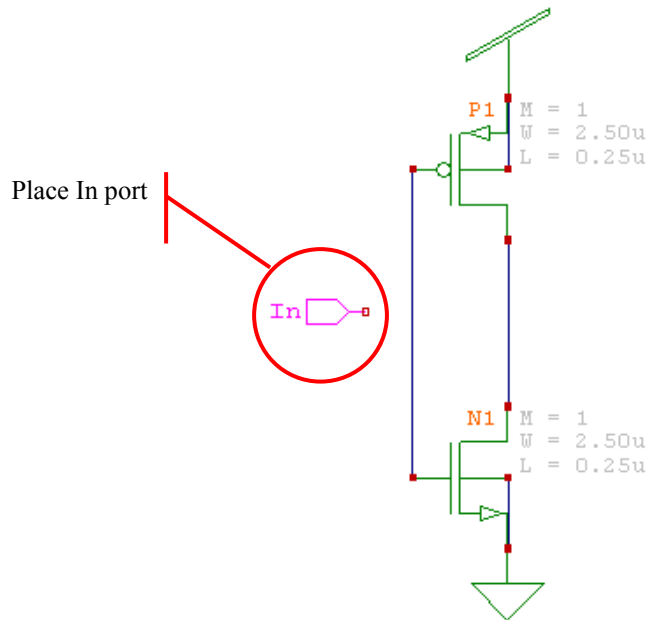


- Now connect the source and bulk ports of the PMOS to Vdd and connect the source and the bulk ports of the NMOS to Gnd.

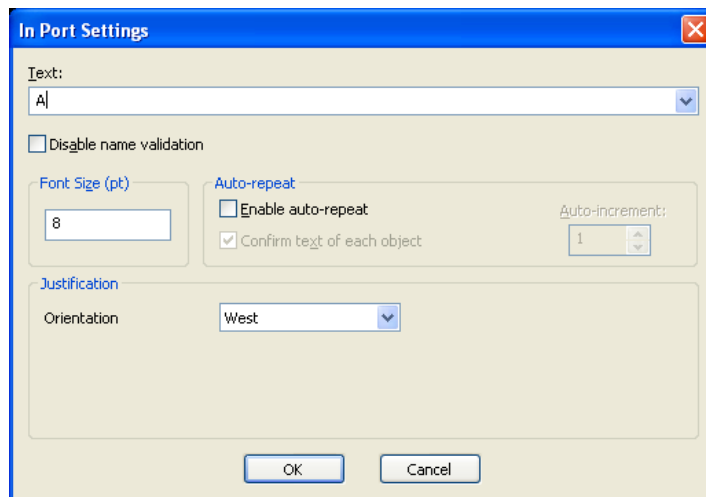


7.1.3. Placing Input and Output Ports

- Now let's place input and output ports on our inverter. Ports may be of type In, Out, In/Out, Other, and Global. To place an In port, click the **In Port** button on the Electrical toolbar (), drag the mouse over the schematic and place the port on the schematic by clicking the left mouse button. Place the port to the left of the schematic, approximately vertically centered.



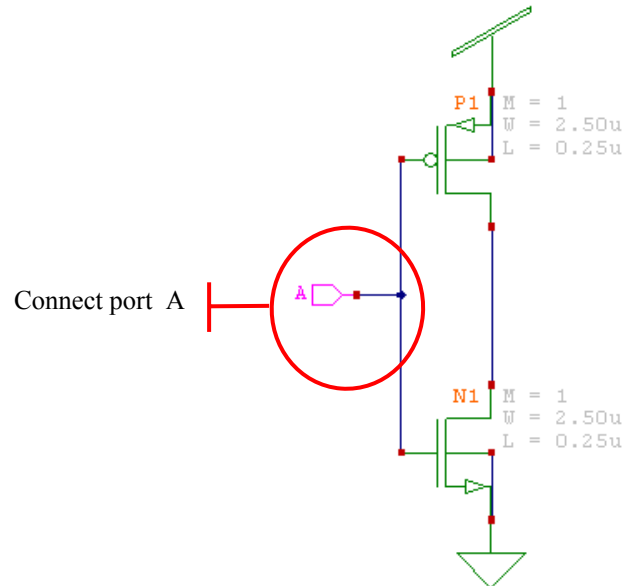
- After clicking the left mouse button to place the port, a dialog will appear to set the port name, and size and justification parameters. Give the port the name “A” and set Orientation to “West”, and press OK.




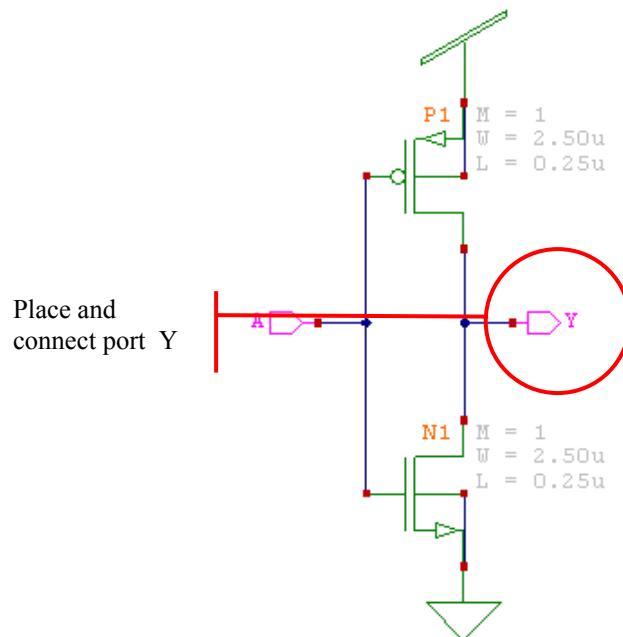
- Multiple ports can be consecutively placed by selecting **Enable Auto-repeat**. This will allow you to continue click-placing ports. If Confirm text of each object is checked, then the dialog appears after each click allowing you to change the port

name. If Confirm text of each object is not checked then the dialog will not subsequently appear, and you can place as many ports as you wish. Press the right mouse button or press the ESC key to exit port placing mode. If the port name ends in a digit, then when multiple ports are placed in succession, the number will increment by the **Auto-repeat** value in the dialog.

- ❑ Now connect port A to the wire that connects the gates.




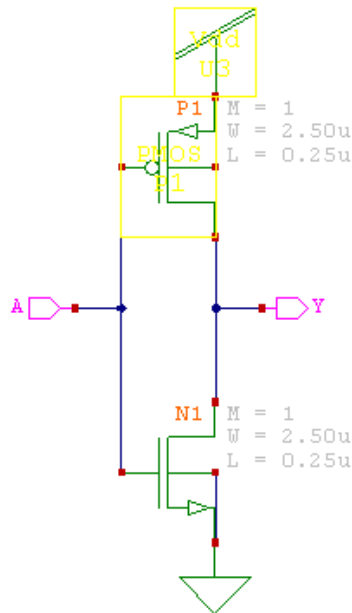
- ❑ Now place a port named “Y” of type Out as shown, and connect it to the wire that connects the two drains. Use the Out port button () to start. Give the Out port Right alignment.



7.1.4. Moving instances with rubberbanding

18. Now let's look at S-Edit's rubberbanding capability that keeps wires connected to instances when moving instances.

- ❑ Select the PMOS and Vdd instances at the top of the schematic, as well as the attached wires, by drawing a selection box as shown below. First enter selection mode by pressing the **Select** toolbar button () , then press and hold down the left mouse button, drag a rectangle as shown, then release the mouse button.

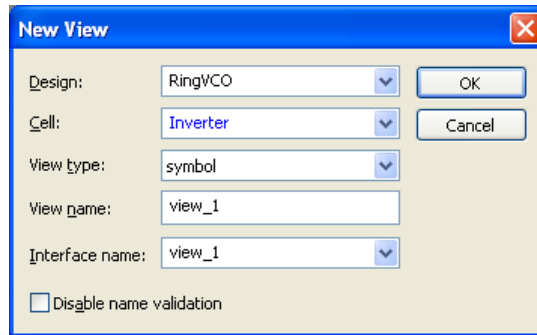


- ❑ Now press and hold the middle mouse button while dragging the mouse up or down to move the two instances without breaking any connections.
- ❑ You can move instances, forcing them to detach from connected wires by invoking **Draw > Force Move** (ALT-M shortcut key) before performing the move operation. Invoke force move, then select the PMOS instance by right clicking on it, then press and hold the middle mouse button and drag the mouse to move it. See that the instance moved, breaking away from its connections. Invoke **Edit > Undo** (Ctrl-Z) to return the instance to its original position.

7.2. Creating a new symbol

19. Now let's make a symbol for our inverter.

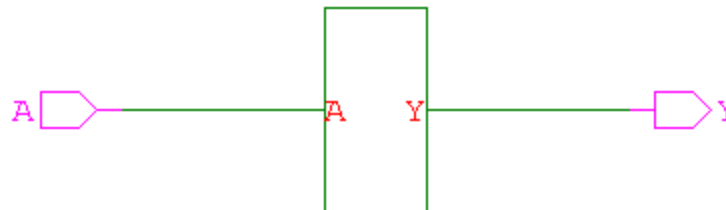
- ❑ Invoke **Cell > New View**, and select Ringvco for design, Inverter for the cell, and symbol for view type. Enter view_1 for View name and Interface name.



- ❑ After pressing OK, you will get an empty drawing area for a new symbol view



7.2.1. Automatic symbol generation and update


- ❑ S-Edit can auto-generate a symbol from the ports on the schematic. Invoke **Cell > Update Symbol**, and S-Edit will create a symbol as shown below:

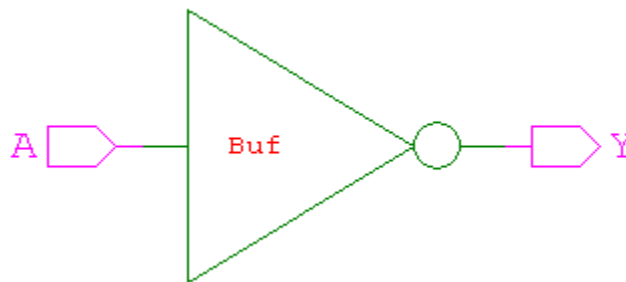


- ❑ If the symbol view is empty, Update Symbol will create simple graphics and place ports corresponding to those on the schematic. It will also place text labels corresponding to each port. If a symbol already exists with graphics or ports, Update Symbol will add the new ports from the schematic to the symbol view, but will not modify the graphics or remove any existing ports.

7.2.2. Editing the symbol

- ❑ We can now easily modify this symbol to make a symbol more suitable for an inverter. Draw a triangle for our symbol using the path drawing tool. To draw a path, select the **Path** drawing button on the toolbar, () , then select the **All Angle** button on the segment toolbar, () , then click the left mouse button to place the first vertex, subsequent clicks with the left mouse button will place additional vertices. Clicking the right mouse button ends the wire without placing a vertex, and double clicking the left mouse button will end the path with placing a vertex at the double click location.

-
- ❑ Draw a circle at the right vertex of the triangle by pressing the circle drawing button on the toolbar. Press the left mouse button at the center of the desired circle, and drag the mouse to create the circle.
 - ❑ Delete the box that was placed by **Update Symbol**. Press the **Select** button on the toolbar () , click on the box to select it (it will highlight when selected) and press the delete key to delete the box.
 - ❑ Move the ports and labels onto the new graphics you have drawn, as shown below. When in selection mode, you can drag a region with the mouse to select all objects in the region. Then hold the middle mouse button and move the mouse to move objects. You can also change the text size of the ports and labels using the properties browser.
 - ❑ Your completed symbol should look like this:



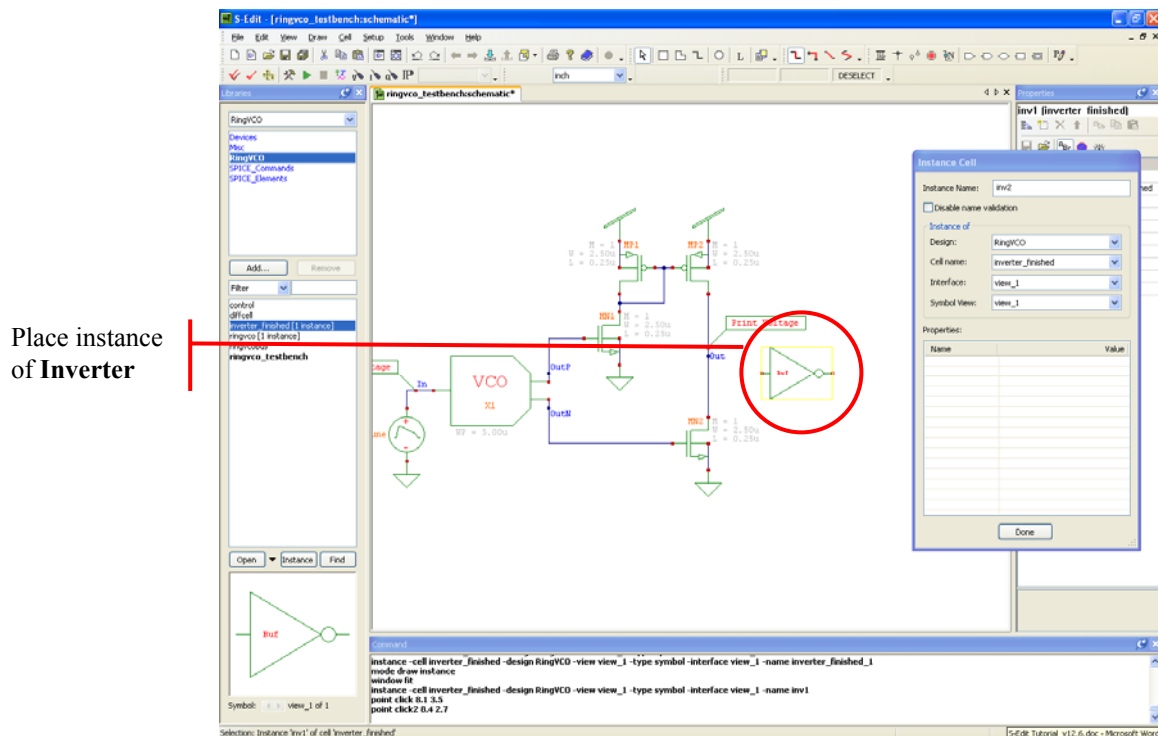
8. Checking the Schematic

20. S-Edit's Design Check tool can check the schematic for many of the common mistakes made during the schematic creation process. These are categorized into errors, which will prevent a proper connectivity from being formed, and warnings which do not prevent connectivity extraction by may be unintended mistakes by the user. Some of the common items the design checker will check for include:

- Dangling wires – wires with unconnected ends.
- Dangling ports – ports on instances with no connection
- Instances with no name or non-unique name.
- Nets have at most one output port reference connected to them.
- Ports with the same name have the same type.

21. Now let's place our inverter in the RingVCO_TestBench and check for errors. If you skipped the steps to draw the inverter, you can use the inverter_finished cell provided.

- ❑ Open the schematic of RingVCO_TestBench and place an instance of Inverter to the right of the schematic.



- [illegible]

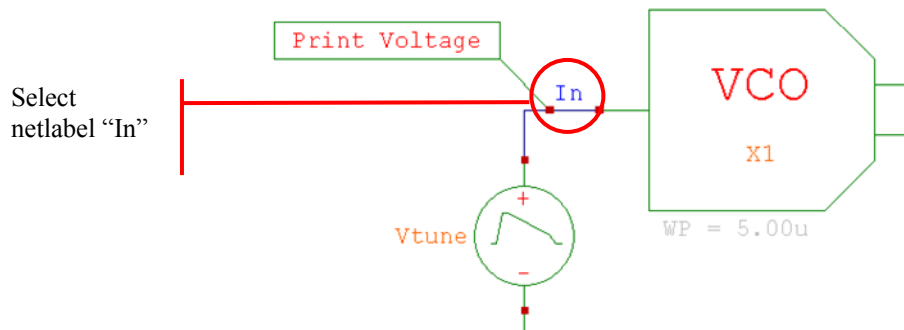
- ```
CHK Warning: Cell: RingVCO_TestBench, Port "A" on instance "inv1" is dangling.
CHK Warning: Cell: RingVCO_TestBench, Wire: Out is dangling.
CHK Warning: Cell: RingVCO_TestBench, netlabel: Buffered is dangling.
CHK Warning: Cell: RingVCO_TestBench, Net: "Buffered" in schematic: is not connected to any port.
SED Design check complete. Design: RingVCO and 0 libraries, 5 cells and 10 views have been checked.
0 errors and 4 warnings were found.
```

- 
- 37

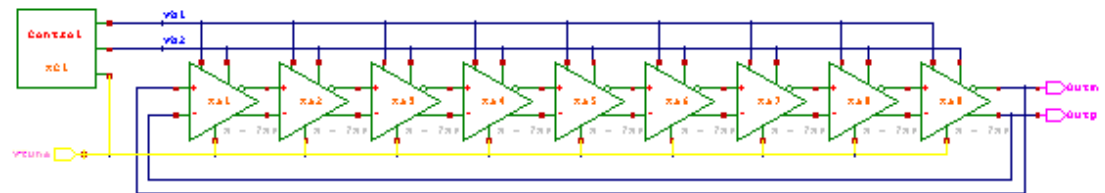
## 9. Highlighting Nets

22. Now let's highlight some nets.

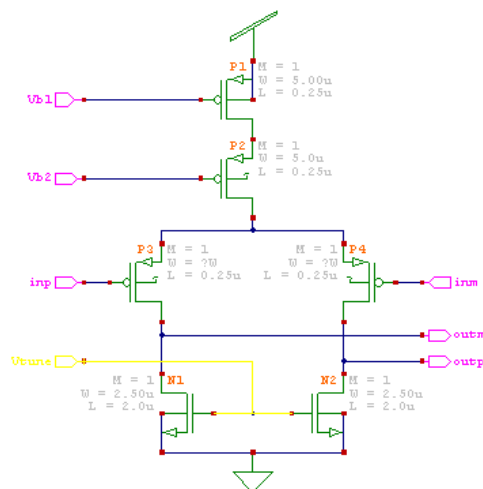
- ❑ Close all windows, and open the schematic view of cell RingVCO\_TestBench. Select any object on the net that is the input to the VCO. Select the netlabel “In”, for example.





- ❑ Double click on the VCO, and see that the net is highlighted as you push into an instance.



- ❑ Double click on the rightmost DiffCell, instance Xa9, and see that the net continues to be highlighted as you navigate down the hierarchy.



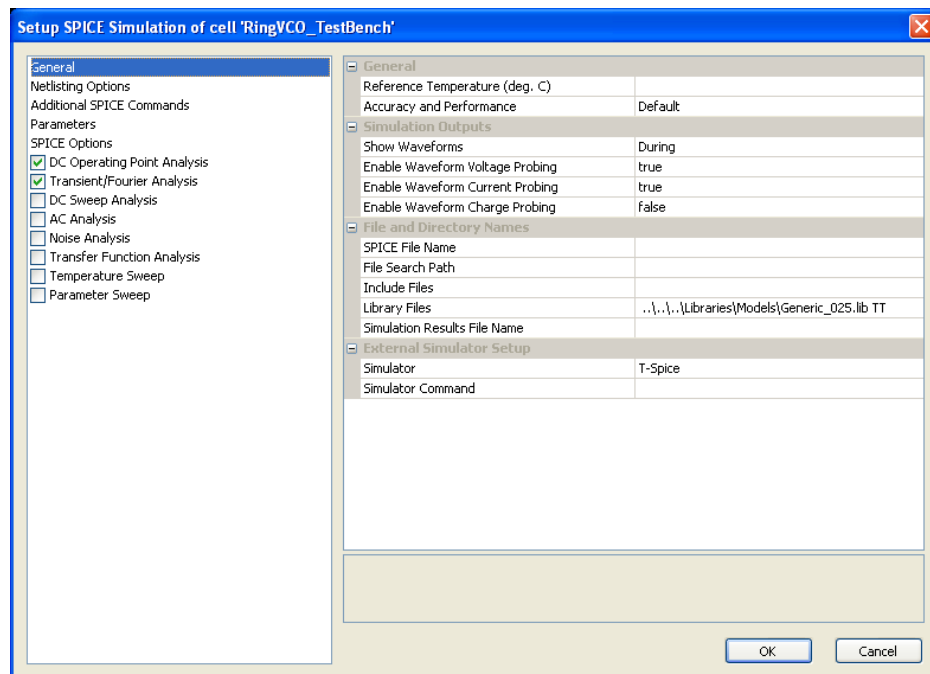
- 
- ❑ Now click on port Outp, and pop up a level of hierarchy using the **Pop context** button () , and see that nets can be traced both up and down hierarchy.
  - ❑ In schematic of RingVCO, select port Outm on the right. Highlight the entire net Outm within the current schematic view by invoking **Tools > Highlight Net**, or press the **Highlight Net** toolbar button () . See the selected net gets highlighted. You can select any object on a net and invoke **Highlight Net** to highlight a net.
  - ❑ You can also highlight nets by name. In the **Command Window**, type `find net VTune`, and see the **VTune** net gets highlighted.



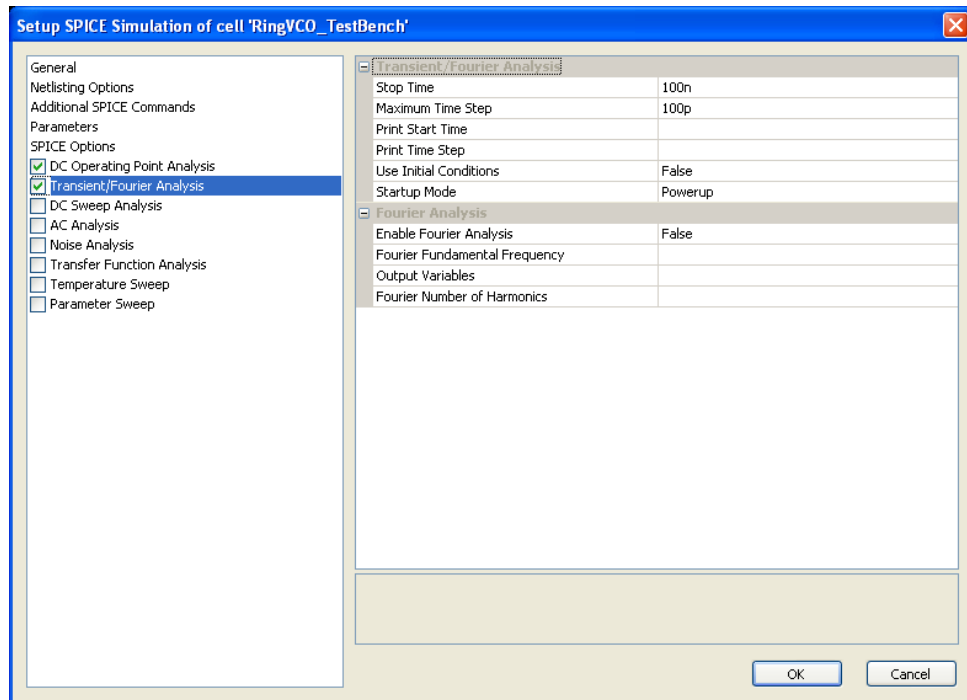
## 10. Simulating the design

23. Now let's run Spice simulation using the T-Spice engine. First let's examine the simulation setup.

- ❑ Close all windows and open the schematic view of RingVCO\_TestBench.
- ❑ The schematic view of RingVCO\_TestBench has been setup with a DC voltage source and an AC voltage source as inputs to the circuit. Print commands have been placed on the input net, and on the nets before and after the inverter. These designate the nets for which voltages will be plotted in the waveform viewer during simulation.
- ❑ Press the **Simulation Setup** button on the toolbar to invoke the **Simulation Setup** dialog. We have selected to run a DC Operating Point Analysis and a Transient Analysis.



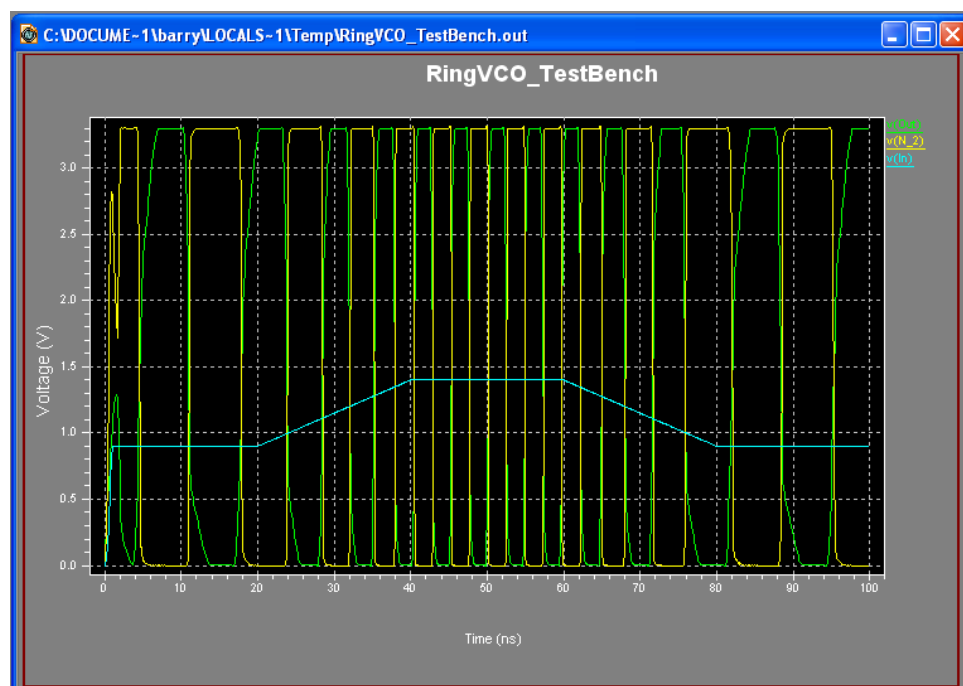
- ❑ Arguments for each type of simulation may be set by selecting that simulation type in the list. Click on different types to see their parameters. For our transient analysis, we have set a Start time of 0s, a Stop time of 100n, and a step time of 100p. We also selected Powerup as our Startup mode.




- ❑ Press OK in the **Simulation Setup** dialog.

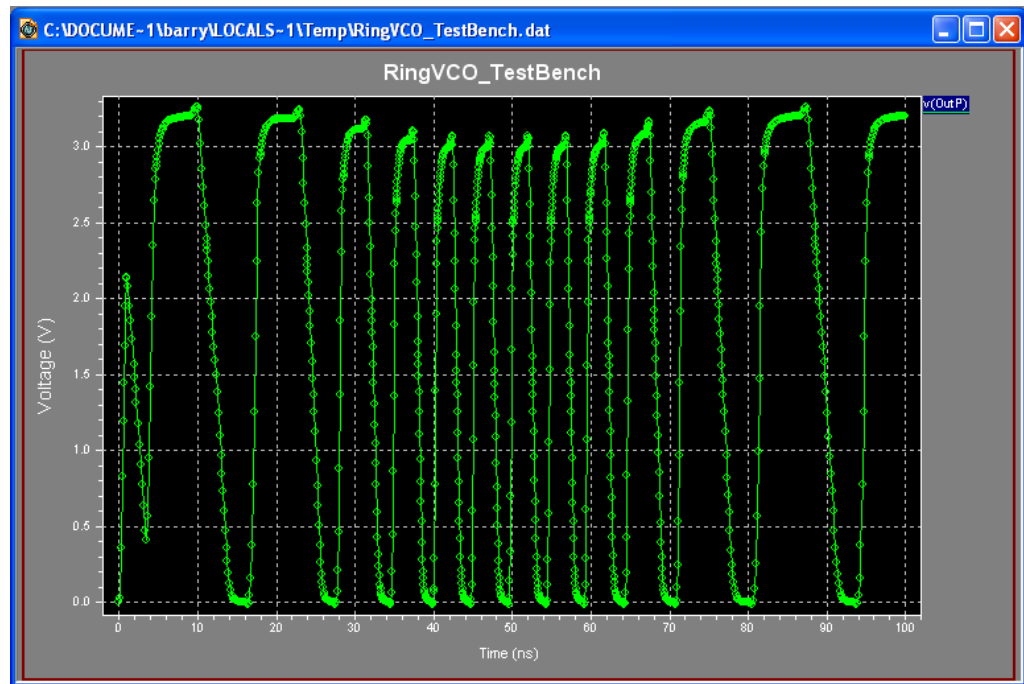
## 10.1. Running simulations

- ❑ Now press the Simulate button, (▶), to run the simulation. Results will appear in W-Edit, displaying the voltage waveforms for the nodes we indicated with the PrintVoltage commands.

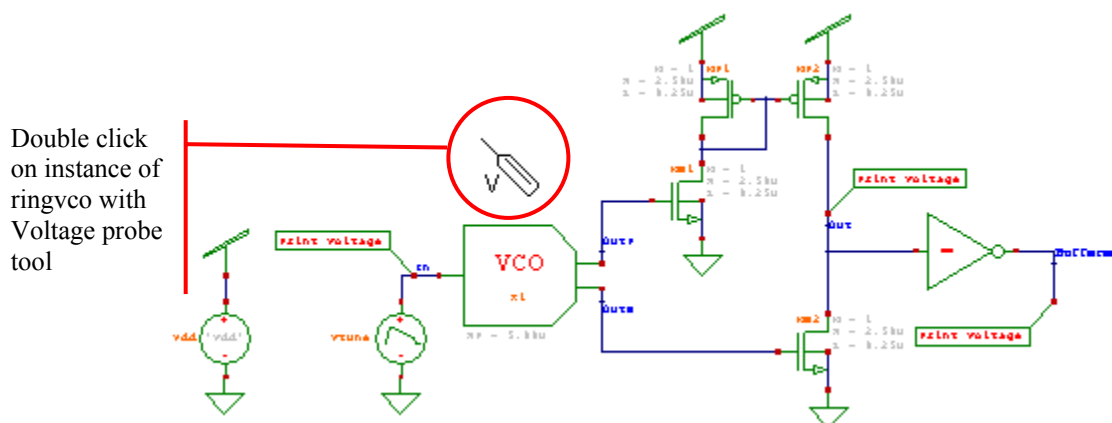


## 10.2. Probing voltages, currents, and charges

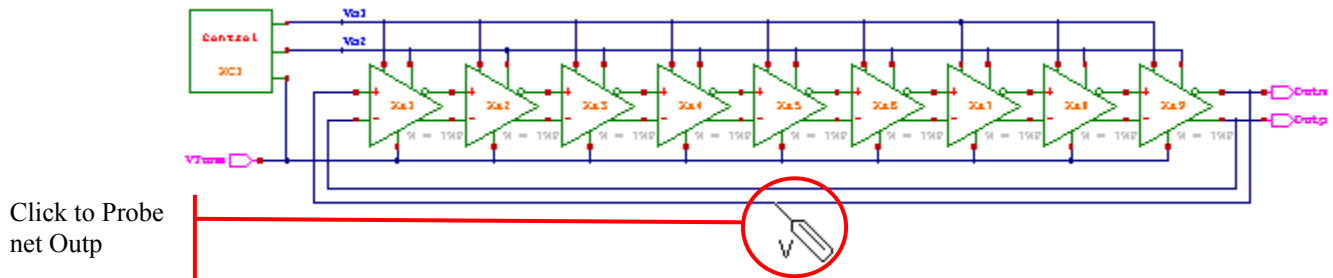
- ❑ We can probe voltages on any net in the circuit by first selecting the **Probe Voltage** button, () , then clicking on the node we wish to probe. The voltage of the probed node will then display in W-Edit.
- ❑ Select the **Probe Voltage** tool, and then probe net OutP in the schematic of RingVCO\_TestBench. The waveform of the voltage of OutP will be displayed in W-Edit, as shown below.



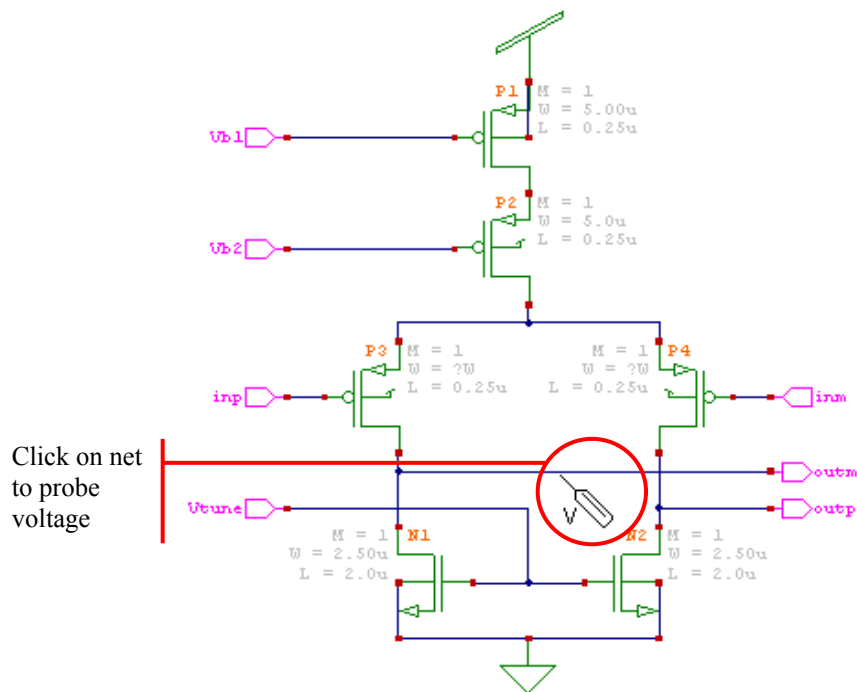
- ❑ We can also probe results down the hierarchy. Using the voltage probe tool, double click on the vco symbol (instance named x1) to push in context into the instance of that symbol.



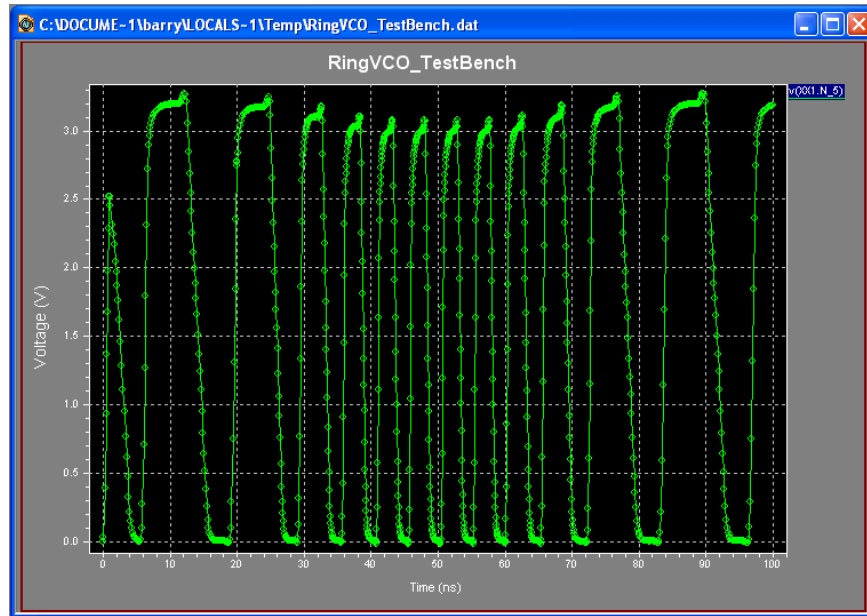
- Click on the bottom net (labeled Outp) to probe the voltage on that net.




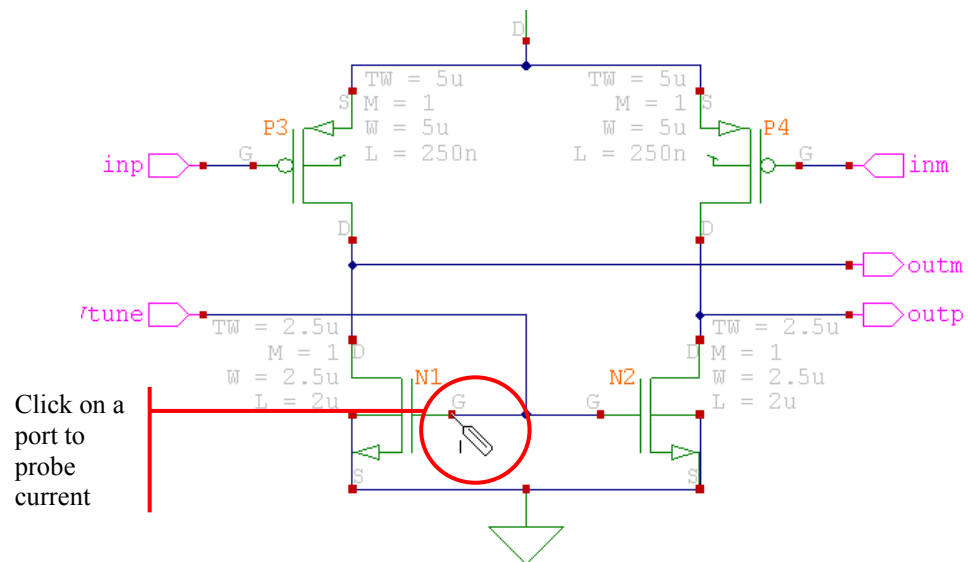
- Using the probe tool again, push into the third instance from the left of DiffCell (instance named Xa3). To push into an instance, you can also select the instance with the probe tool, then press the **Push into context** button (↓) on the toolbar. To pop back up the hierarchy, press the **Pop context** button (↑).



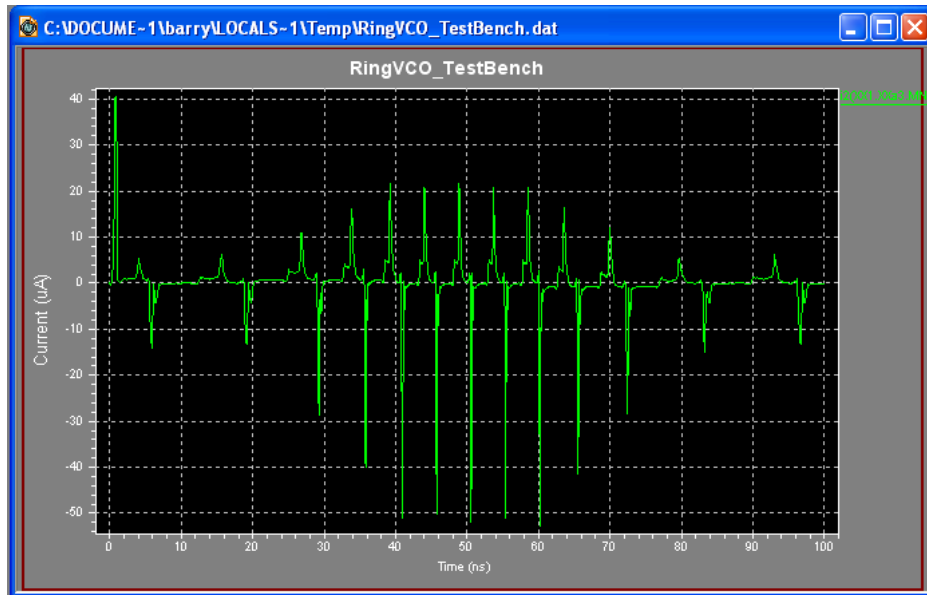
- Probe the voltage on the internal node as shown above. Results will display in W-Edit.




- ❑ You can probe Currents into the port of a symbol by selecting the **Probe Current** button, () , then clicking on the port you wish to probe.
- ❑ Click on the red square showing the connection to the gate of instance N1, as shown below, to probe the current into that gate.



- ❑ The current of the probed port will then display in W-Edit.




- ❑ Similarly, if the simulation is setup to do so, charges can be probed by selecting the **Probe Charge** button, () , then clicking on the terminal you wish to probe. The simulation in this tutorial is not setup for probing charges.
- ❑ You can also display the small signal parameters of a primitive device by probing that device. Double-click on the top instance of PMOS to see its small signal parameters.

**Small-signal Parameters**

Device Name:

| Name   | Value      |
|--------|------------|
| MODEL  | PMOS       |
| TYPE   | PMOS       |
| REGION | Cutoff     |
| ID     | -1.85e-011 |
| IBS    | 0          |
| IBD    | 0          |
| VGS    | -0.0175    |
| VDS    | -0.0298    |
| VBS    | 0          |
| VTH    | -0.453     |
| VDSAT  | -0.0418    |
| BETA   | 0.000703   |
| RS     | 0          |
| RD     | 0          |
| GM     | -5.28e-010 |
| GDS    | -3.4e-010  |
| GMB    | -1.67e-010 |
| GBD    | 0          |
| GBS    | 0          |
| CDTOT  | 1.93e-014  |
| CGTOT  | 7.42e-015  |
| CSTOT  | 2.88e-014  |
| CBTOT  | 4.45e-014  |
| CGS    | 2.66e-015  |
| CGD    | 2.75e-015  |
| CGB    | 2.01e-015  |
| CBD    | 1.65e-014  |
| CBS    | 2.6e-014   |

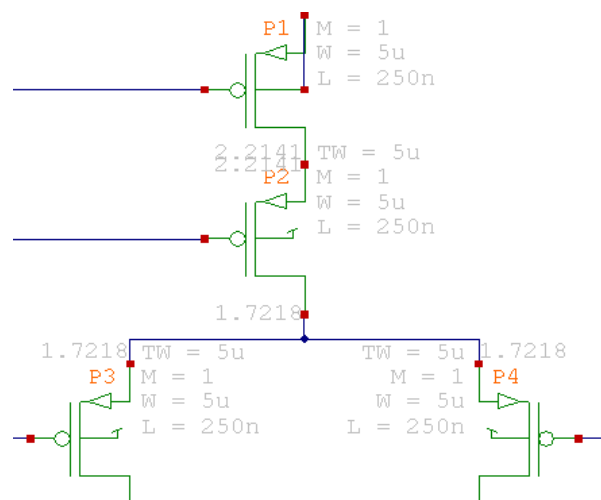
OK

- 
- ❑ Try navigating the hierarchy in context and probe different nodes.
  - ❑ You can also export the Spice netlist for the currently active schematic view into **T-Spice**, and then in **T-Spice** you can add additional commands for simulation. Open the schematic view of `RingVCO_TestBench`, and press the **T-Spice** button, () on the toolbar.

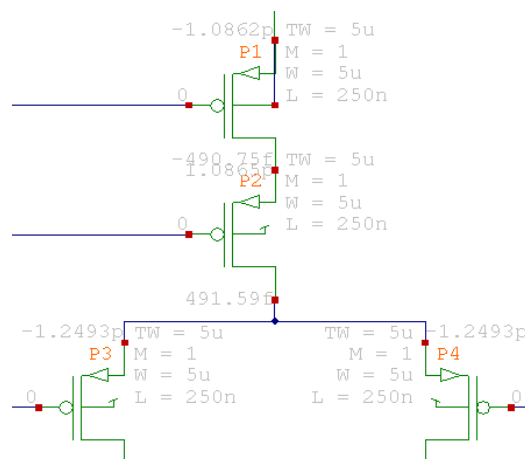
## 11. Viewing Voltages, Currents, and Charges on the Schematic

24. If you have not already done so, then begin this section of the tutorial by opening the schematic view of `RingVCO_TestBench` and running simulation. Probe in context into the VCO, then into the third `DiffCell` from the left, `Xa3`.

- DC operating point Voltages, Currents, and Charges can be displayed at the ports of primitive elements using **annotate port** properties, as discussed in the section **Annotate Port Properties**.
- Enable **Display Evaluated Properties** and select **Voltage** in the dropdown to make the annotate properties display the DC operating point voltages at the corresponding port locations:



- Select **Current** in the **Display Evaluated Properties** dropdown to make the annotate properties display the DC operating point currents at the corresponding port locations:





---

## 12. Busses and Arrays

25. S-Edit supports arrays, busses and net bundles. First let's describe nets, buses, bundles, and arrays, then we'll look at some examples. A net is the fundamental single unit of connection. A bus is a set of connections with the same name and a numerical identifier and increment, and a bundle is collection of nets and busses.

- An array is created by assigning an instance with an instance name containing array syntax.

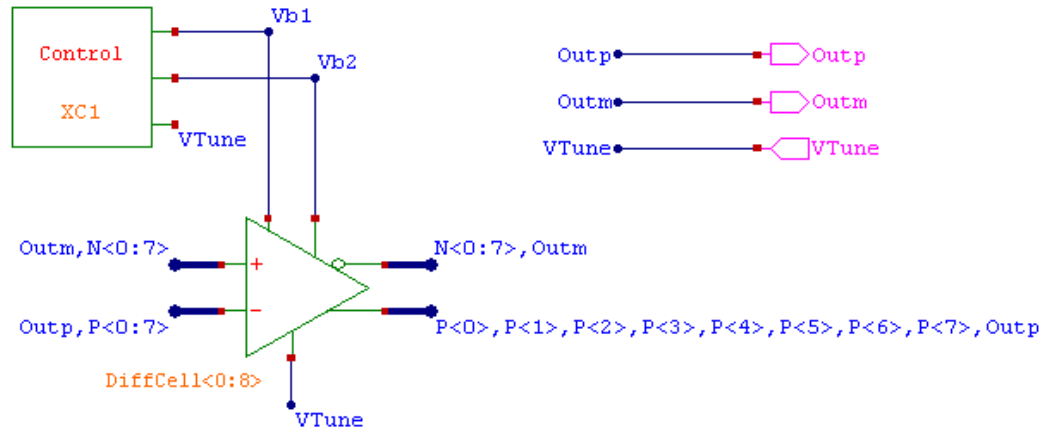
- Naming an instance `array_name<n1:n2:step>` creates an array of instances named `array_name<n>`, where `n` starts at `n1`, ends at `n2`, and increments by `step`. `Step` is 1 by default, and may be omitted. The instance name `U<0:7>` defines an array of instances, with elements `U<1>`, `U<2>`,...`U<7>`. Two dimensional arrays may be created by naming an instance `U<n1:n2:step1><n3:n4:step2>`. The second range increments first. The instance name `U<0:7><0:3>` creates an array of instances named


`U<0><0>`, `U<0><1>`, `U<0><2>`, `U<0><3>`,  
`U<1><0>`, `U<1><1>`, `U<1><2>`, `U<1><3>`,  
`U<2><0>`, `U<2><1>`, `U<2><2>`, `U<2><3>`,  
...  
`U<7><0>`, `U<7><1>`, `U<7><2>`, `U<7><3>`

- A bus is created by assigning a net a name using bus syntax.
- Similarly to arrays, naming a wire, using the net label tool, `bus_name<n1:n2:step>` creates a bus of nets named `bus_name<n>`, where `n` starts at `n1`, ends at `n2`, and increments by `step`. `Step` is 1 by default and may be omitted. The name `A<0:7>` creates an 8 bit wide bus, with nets `A<1>`, `A<2>`, ... `A<7>`. Two dimensional buses may be created by naming a wire `bus_name <n1:n2:step1><n3:n4:step2>`. As with arrays, the second range increments first.

26. Now let's look at our tutorial example.

- Cell `RingVCO_ArrayBus` is a redesigned copy of cell `RingVCO`, using an array and buses. Close all windows and open the schematic view of cell `RingVCO_ArrayBus`. See how we have redesigned the `ringvco` using an array of diffcells.



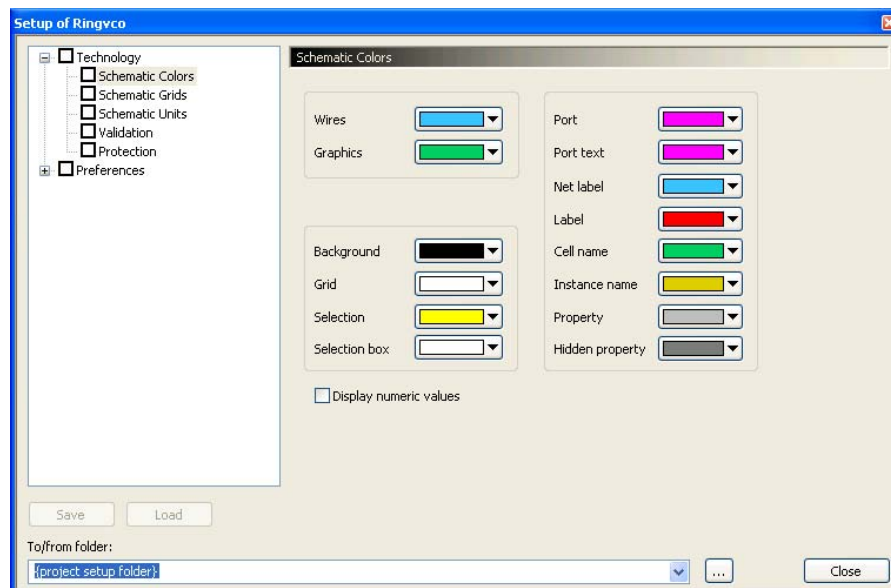
- ❑ We have created an array of 9 diffcells by giving the diffcell instance the name `DiffCell<0:8>`. The `<0:8>` component of the name designates the instance as an array. Note the bundle `Outm, N<0:7>` connected to the `inp` port of the diffcell and the bundle `N<0:7>, Outm` connected to the `Outm` port of the diffcell. Also observe that buses can be named in the compact form `bus_name<n1:n2, step>`, or can be expanded explicitly, naming each component.
  - ❑ Open cell `RingVCO_TestBench`, and replace the instance of `RingVCO` with an instance of `ringvcoBus`.
  - ❑ Export the Spice netlist for `RingVCO_TestBench`, by pressing the **T-Spice** button, (  ), on the toolbar. Note the arrays have been expanded into individual subcircuits and the buses have been expanded into individual connections, as Spice does not support arrays and buses.
  - ❑ Close T-Spice and press the **Simulate** button to run T-Spice simulation.
27. Additional examples of buses and arrays are available in the Examples folder that can be installed from S-Edit, `C:\Documents and Settings\<username>\My Documents\Tanner EDA\Tanner Tools v12.6\S-Edit\Examples`

---

## 13. Customizing the setup

28. Settings for various configuration parameters are available in the **Setup** menu.

- ❑ The following is the list of settings that are available:
  - Colors – Sets the colors of objects in the schematic, such as wires, symbol graphics, ports, labels, etc.
  - Grids – Sets the Major, Minor, and Snap grids.
  - Units – Sets the Units (mm, cm, meter, inch) and the size of an internal unit.
  - Validation – Sets the tcl procedures to call for validating cell names, instance names, view names, port text and net labels.
  - Protection – Set the Allow Edit option.
  - General – Sets windowing behavior, to reuse windows or open a new window each time a view is opened.
  - Selection – Sets Selection options
  - Text Editor and style – Sets Text Editor options.
- ❑ Invoke **Setup > Technology > Schematic Colors**



- ❑ Select different colors for Wires, Graphics, and various other elements. Note that your changes take effect immediately. To save the new settings to the design folder so they will be automatically present next time you load the design, select **{project setup folder}** in the **To/From folder**, and press the **Save** button. If you do not save the settings, they will be in effect until you close the design, but will not be there when you reload.
- ❑ Try changing some other settings on other pages in the Setup dialog.