SUE Design Manager Tutorial

Micro Magic Tools

Version 4.3

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.micromagic.com

Part Number: SUE4.3, Revision 1

Table of Contents

Part 1
Getting Started With SUE Design Manager

Part 2
Schematics And Icons

Part 3
Simulating Circuits With SPICE And IRSIM

Part 4
Higher Level Schematics And Verilog Simulation

Part 5
Cross-probing With MAX

About this Document

This chapter provides an overview of the *Micro Magic Tools SUE Design Manager T utorial*:

- Objectives on this page.
- Getting set up on page vii.

Objectives

This tutorial is a intended to be a step-by-step introductory training in the use of the Micro Magic Tools (TM) SUE Design Manager. By following the steps in this tutorial, you will learn how to run SUE, and how to use SUE to optimize your designs. Once you are familiar with SUE, refer to the Micro Magic Tools SUE User Guide for more detailed information about the tool and its options.

Audience - SUE

This document is designed for chip designers and architects. It assumes that you have a strong foundation in IC design and development, and schematic design. This document also assumes that you are familiar with the following programming languages and operating systems:Tcl/Tk, and UNIX or Linux.

Organization

This tutorial is divided into parts. Each part leads you through a different aspect of the Micro Magic Tools SUE Design Manager.

This document contains the following parts:

- Preface, “About This Document” (this chapter), provides a brief description of the content and organization of this document.
- Part 1, “Getting Started With SUE Design Manager”, gives a brief overview of what you can accomplish using SUE, and the requirements necessary for running the program.
- Part 2, “Schematics and Icons”, provides instructions on using SUE to draw schematics utilizing icons and generators. In addition, this section details methods for wiring up these icons into a circuit.
- Part 3, “Simulating Circuits Using SPICE And IRSIM”, covers how to use SUE to netlist, simulate, and verify the schematics you have created.

- Part 4, “Higher Level Schematics And Verilog Simulation”, demonstrates the SUE Design Manager’s capability of creating hierarchical schematics and simulating those schematics with Verilog.

- Part 5, “Cross-probing With MAX”, introduces the inter-process communication between SUE and MAX, the MMT full-custom layout editor. With cross-probing, you will be able to select any net in a schematic in SUE and have MAX highlight that net in the layout, or vice-versa

Chapter Organization

Each part contains descriptions of the commands and methodology necessary to create schematics using SUE, with step-by-step, illustrative examples describing the usage of commands and procedures.

Documentation Conventions

General Conventions

This guide uses the following text conventions:

- Statements, commands, command output, filenames, directory names, and configurations are shown in a boldface, fixed-width font. The following example shows a full path name:

  `~/mmi_private/sue/.suerc`

- In examples, text that you type literally is shown in bold. In the following example, you type the word *max*:

  To launch sue, type **sue**

- Menus, menu options, and pop-up menus are generally shown in a boldface, sans serif font. For example:

  Select **Save As** from the **File** Menu.

Conventions for Software Commands and Statements

When describing the Micro Magic Tools (TM) software, this guide uses the following type and presentation conventions:

- Statement or command names that you type literally are shown nonitalicized. In the following example, the statement name is *set MC(default_generator)*:

  set MC(default_generator) sram

- Variables for which you substitute appropriate values, are shown in itallics. Optional commands within a string of code are often enclosed within square brackets ([ ]). Variables are enclosed within angle brackets (< >). When you type the **`setMACRO`** statement, you substitute a value for *name*, *row1* and *row2*. *CELL* is an optional command.

set MACRO(*<name>*){[*CELL*]*<row1><row2>*...}

Conventions for Mousing

Throughout this document, we assume you are using a 3-button mouse. When we refer to using the mouse, the following settings apply:

- Mouse-Button-1, or Button-1, refers to the left-most button (index finger) when you are using a right-handed mouse.
- Mouse-Button-2, or Button-2, refers to the middle, or center button (middle finger).
- Mouse-Button-3, or Button-3, refers to the right-most button (ring finger) when using a right-handed mouse.

  Reverse these settings if you are using a left-handed mouse, using your mouse control program, so that you click Button-1 with your index finger, Button-2 with your middle finger, and Button-3 with your ring finger.
- "Click" means to depress the designated mouse button once.
- "Double-click" means to depress the designated mouse button twice, quickly.
- "Drag" means to depress and hold down the designated mouse button while moving the mouse.

Documentation Feedback

We are always interested in hearing from our users. Please let us know what you like and do not like about the Micro Magic Tools documentation, and let us know of any suggestions you have for improving the documentation. If you find any mistakes or out of date information, please send email to *support@micromagic.com*

Support

For product problems or technical support issues, contact Micro Magic Tools at *support@micromagic.com*.

Getting set up

SUE runs independently from any of the Micro Magic Tools suite.

First, ensure that the Micro Magic Tools software has been installed, and that you have a valid SUE License.

To run some parts of the tutorial you will need to have programs like HSPICE, Verilog, and IRSIM (IRSIM switch level simulator is included with MMI-SUE). To run these simulators, SUE must have been properly set up to access the HSPICE or Verilog simulator.

If you do not have one or more of these simulators, simply skip that section and go on to the next.

Part 1
Getting Started With SUE Design Manager

Introduction

Welcome to SUE Design Manager, from Micro Magic Tools. SUE is more than just a schematic capture program. It is a complete graphical user environment.

SUE Features

With SUE, you will be able to:

- Draw, view, and edit schematics, icons, random graphics and text.
- Use the built-in netlister to Verilog, HSPICE, Berkeley SPICE, and IRSIM.
- Cross-probe directly with Verilog, IRSIM, HSPICE, and MAX (the Custom Layout Editor from Micro Magic Tools).
- Interactively run Verilog and IRSIM, with logic values displayed directly on the schematic.
- Automatically attach documentation, Verilog models, etc. to schematics.
- Automatic version control.

This tutorial will carefully walk you through these features and many more.

Getting Started

Before we get started make sure someone has already installed SUE at your site.

> **Note**
> To run some parts of the tutorial you will need to have programs like HSPICE, Verilog, and IRSIM (IRSIM switch level simulator is included with MMI-SUE). If you do not have one or more of these simulators, simply skip that section and go on to the next.

Starting Up SUE

To run the tutorials, you need to make a personal copy of the SUE tutorial directory.

Step 1. To install the SUE tutorial, type:

```
mmi_tutorial
```

at the UNIX prompt. Make sure that the appropriate tutorial is selected and then click on **Install Tutorial**. The default installation directory is "`mmi_private/tutorial`" in your home directory. You can also select a different directory for installation. Don't worry if the directory doesn't exist, the script will make it if it can. You can also use this to reinstall a clean copy of the tutorial.

Step 2. To start the tutorial, you must first "**cd**" (change directory) to the directory that you installed the tutorial into. If you used the default directory, type:

```
cd ~/mmi_private/tutorial/sue
```

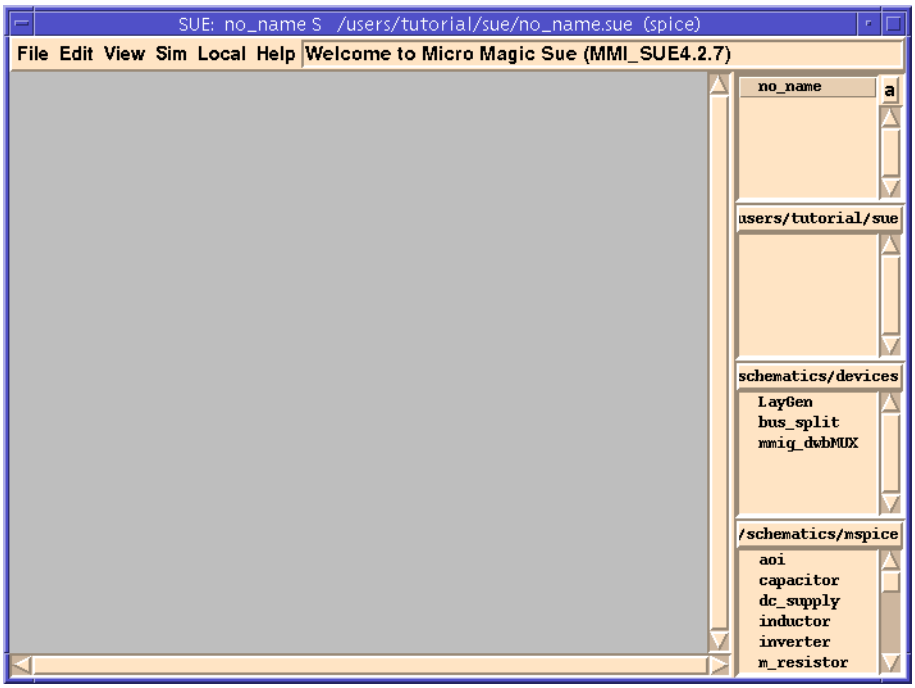
Otherwise, replace the above directory with the directory you selected. This directory contains files that you will need to run the tutorial.

Step 3. Then to start SUE type:

```
sue
```

A window should come up that looks something like this:

Figure 1: SUE Window



SUE Screen Layout

On the very top of the window the title bar should say:

SUE: no_name S <path_to_cell> (spice)

"**no_name**" means that you have not specified the file (schematic) you wish to edit. The "**S**" means that you are editing a *schematic*, as opposed to an icon. The "**(spice)**" means that you are currently in SPICE simulation mode.

Below the title bar you should see the menu bar which contains the following menu items: **File**, **Edit**, **View**, **Sim**, **Local** and **Help**. These are pull down menus much like any window-based application using click and drag.

Directly to the right of the **Help** menu is the **SUE Message Area**. It currently says "Welcome to Micro Magic SUE (MMI_SUE4.3.3)." Note that the 4.3.3 (or whatever the number is) refers to the version of SUE you are running. If you want to see more information about the version, go to **About SUE** in the **Help** menu.

Library List Boxes

Down the right side of the window are several small Library List Boxes.

The top one is the **Schematic List** box. This lists all current schematics that have been loaded into SUE. Currently only "**no_name**" should be listed.

All of the rest of the List Boxes are for icons. (In some tools, *Icons* are called *Symbols* -- they are the same thing). Each **Icon List Box** displays the loaded icons in a given library, which is also a UNIX directory.

The UNIX directory of the Library in each **Icon List** Box is shown at the top of the List Box, clipped from the left. You can change what Library goes in which List Box by holding down Button-1 on this directory name and selecting a different library from the list. Libraries are automatically added to this list when you load any element from them. You can also load all of the icons in the Library by selecting **`Autoload directory,`** and add or subtract List Boxes with this menu.

The top **Icon List** box typically contains the library that you are working in. When you create new icons, they will show up there. The other List boxes contain other libraries. In this example, the **mspice** and **devices** libraries are shown. The **devices Library** is necessary since it contains the all-important I/O icons: *input*, *output*, *inout*, and *global*. It also contains other handy general-purpose icons. The **mspice Library** is useful for low level circuit design since it contains parameterized transistors, simple gates, and the like.

SUE Pull Down Menus and Hotkeys

SUE pull down menus work just as in most other window-based applications. Each command is listed with its hotkey shortcut next to it (if it has one). In addition, when you drag the cursor over an entry, a description of the command is displayed in the **SUE Message Area**.

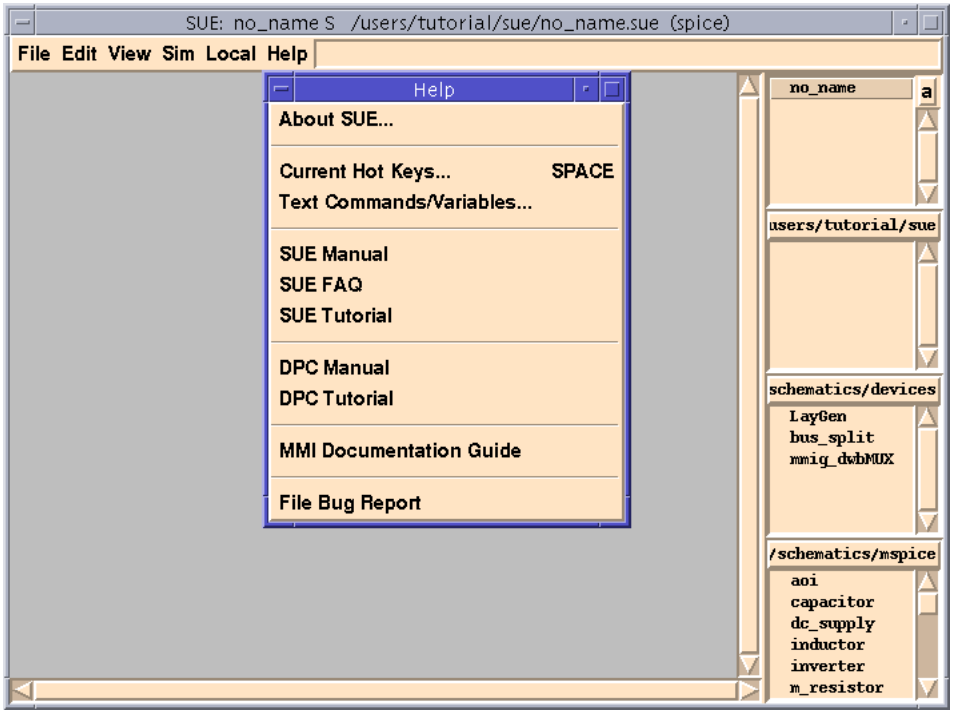
Step 1. Try this out. Go to the **Edit** menu and move the cursor from command to command. Can you find the help description? What is the hotkey to add a wire?

SUE menus can also be torn off -- that is, turned into top level windows that don't roll back up when you're done with them. This can be useful if you need to go to them frequently and they don't have a hotkey (or you can't remember it).

Step 2. To tear off a menu, simple click on the "--------" (dotted line) at the top of the exposed menu. Try tearing off the **Help** menu now.

You should now see something like Figure -2. You can manipulate this window like any other Xwindow (window manager dependent). For example, you can move it by clicking and holding down the left mouse button (Button-1) over the top title bar and dragging it around. Or you can close the window by holding down the right mouse button (Button-3) over the top title bar and selecting **Close** from the menu that pops up.

Figure 2: Help Menu



Note

You can also just click on the menu items and the pull down will stay down until you click on the desired item.

Scroll bars are shown across the bottom and right side of SUE (and are found in the List boxes). These work like the scroll bars in most other applications.

Step 3. To view the complete SUE on-line manual, select **SUE Manual** from the **Help** menu. In addition to providing detailed information on all of SUE's commands, modes, and features, the SUE manual explains the philosophy behind SUE. After finishing this tutorial you should read through the manual. In fact, it would be very useful to read the sections on SUE Philosophy and Selection at this time.

Step 4. To learn about all the Micro Magic Tools software in this distribution, select **MMI Documentation Guide** in the **Help** menu.

Step 5. Now hit the Space key (or **Current Hot Keys** in the **Help** menu). You will now see all of the hot keys in alphabetical order, plus a description of what they do. This window is context specific, so when you are in a mode like **Add Wire** and hit the Space key, you will see just those hotkeys which are active in that mode. Hit Space again to close the hotkey window.

Step 6. Now close the **Help** menu.

Part 2
Schematics And Icons

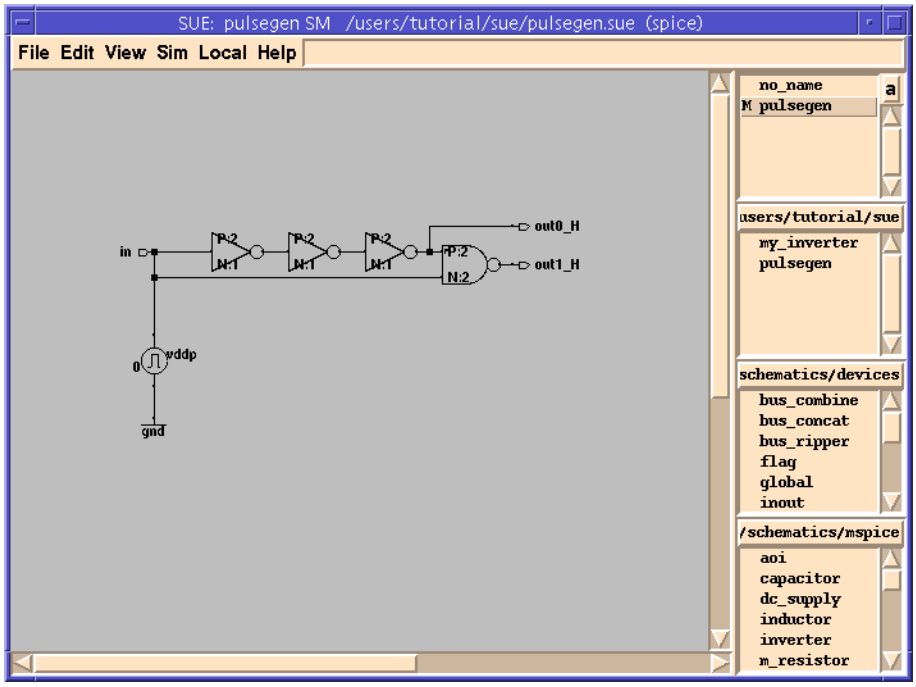
Introduction

Drawing A Schematic

OK, we are now ready to draw a schematic using SUE.

We are going to draw a very simple (and not very useful) circuit that simply generates a pulse. The circuit we are going to draw is shown in Figure 3.

Figure 3: Simple SUE Schematic



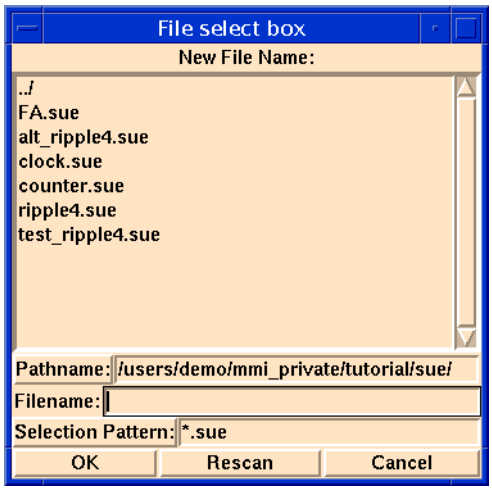
Creating a New File

First we will create a new schematic with a given name.

Step **1.** Pull down the **File** menu and select **New Schematic**. This will bring up the **File Select** box, as shown in Figure 4. Note that you could have also used the hotkey Ctrl-n to bring up this menu as well. That is, while holding down the Control key, type “n”.

> **Note**
>
> Using the hotkeys for the most common commands will make you much more productive when using SUE.

Figure 4: The File Select Box



The **File Select** box lets you select a directory along with the schematic name. This is the directory where the schematic will be saved.

Step **2.** Type the name of the schematic “**pulsegen**” (or any other name you like) near the bottom where it says “**Filename**:”. (If the cursor isn't blinking to the right of “**Filename**:”, click the mouse button there first.) Now either hit Return, or click the **OK** button.

You should now see “**pulsegen**” in the **Schematic List** box and also in the title bar at the top of the window. The title bar tells you that you are now editing **pulsegen**.

Now to draw our schematic.

Beginning the Schematic

To draw a schematic we are going to do 3 things:

- Add icons to the schematic,
- Modify properties of those icons, and

- Draw wires.

First we add icons from the **Icon List** boxes:

Step 1. Find the "**inverter**" icon in the **mspice Library Icon List** box. You might have to scroll to find it. Next, click Button-1 over "**inverter**" and then move the cursor over to the schematic window where you should see the inverter icon appear. Drop the inverter where you want it by clicking Button-1 again.

Notice that an "**M**" has appeared to the left of the "**pulsegen**" name in the **Schematic List** box, at the top right of the SUE window, and also in the title bar of the window. The "M" tells you that the cell is *modified*.

SUE doesn't let you exit if you have any modified schematics or icons without verification. Let's try it to see what happens.

Step 2. Select **Exit** from the **File** menu or hit Ctrl-d. The **Cells Modified** dialog box will come up and tell you that you have modified cells.

Step 3. Click on **Cancel** since we don't want to exit yet. If you had clicked **Exit and Lose Changes**, SUE would have quit WITHOUT saving your modifications.

Let's add another icon, but this time we'll try something fancier.

Step 4. Click Button-1 over the "**inverter**" in the **Schematic List** box again and move the cursor to the schematic.

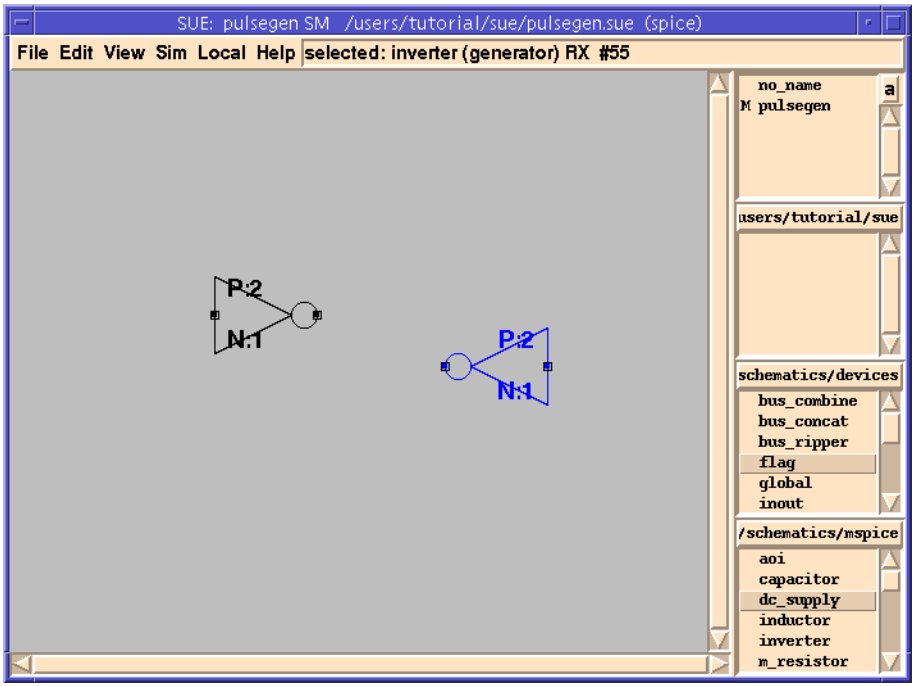
Step 5. Hit the Space bar on the keyboard. This brings up a list of all of the things you can do before dropping the inverter. An abbreviated list is also displayed in the **SUE Message Area**. Hit the Space bar again to make the window go away.

Step 6. Hit "x" to flip the inverter in the X direction.

Step 7. Hit "z" to zoom in around the inverter. Hit Shift-z to zoom out around the inverter.

Step 8. Click on Button-1 to drop the inverter. Your schematic should now look something like Figure 5.

Figure 5: Two Inverters



Selecting Icons

In SUE, most editing operations apply to the selected object. **Objects**, either *icons*, *wires*, *text*, *arcs*, or *lines*, change color to blue if they are *selected* (the colors referred to here are the default colors). You probably noticed that if you put the cursor over an icon, its color changes to white. This is known as the *active object*.

Step 1. Move your cursor over one of the inverters. It should change to white. This inverter is ready to be *selected*.

Step 2. Move your cursor slowly past the center of the icon. Notice that the icon will turn black momentarily. What happened? The icon is defined as what is black and thus the inside is not part of the icon. This may seem weird at first.

Step 3. Click Button-1 when the inverter is white. It should change to yellow which means it is both *selected and active*.

Step 4. Move your cursor away from the inverter. It should change to blue which means it is *selected but not active*.

SUE lets you do lots of interesting things to the selected object. For example, let's take the inverter you just dropped and rotate it

Step 5. Select the rotated (flipped in the X direction) inverter.

Step **6.** Hit the r hotkey twice to rotate it back to the original orientation.

> **Note**
> If two objects are close together, move the cursor around a little bit to get the desired object to become active. Also, by making an object active you can quickly tell what is included in that object since only that object will change color.

Selected objects are also described in the **SUE Message Area**.

Step **7.** Select the inverter you just rotated, and you should see something like:

selected: inverter (generator) RY #23

This is telling you that you have selected an inverter which is a generator (more about that later) and which is flipped in the Y direction and has the unique identifier of "23" in this schematic. The first inverter should have an orientation of **R0** (no flipping or rotating).

To recap, objects can be in one of 4 states:

- black -- default (inactive, unselected)
- white -- active
- blue -- selected
- yellow -- active, selected

The two main ways to select objects with the cursor are by clicking Button-1 on an active object as described above, or by dragging a select box around a region. By default, the previous selection will be cleared when making a new selection. To prevent this, simply hold down the Shift key when selecting, and the new selection will add to the old selection. Also, clicking Button-1 over a blank part of the schematic (that is, with nothing active), clears the selection.

Now you try.

Step **1.** First, add a few more icons to the schematic.

Step **2.** Click and hold Button-1 down and drag a red selection box around a few objects and then release Button-1. All of the objects completely inside of the red selection box should turn blue and be selected.

Step **3.** Draw a box around another portion of the schematic. The icons that were only in the first box will deselect and the new ones will select.

Step **4.** Hold down the Shift key and draw another box. The previously selected icons should stay selected.

Step **5.** With the Shift key still down, click Button-1 over an active, selected (yellow) icon. Only that icon should deselect.

Now let's do something to the selected object.

Step **6.** Select one inverter. Remember, you can either click Button-1 when it's active to select it or drag a box around it with Button-1.

Step **7.** Rotate the inverter by hitting the r hotkey. You could also use the menu command **Rotate** in the **Edit** menu. The inverter should now be facing down.

Step **8.** Rotate the inverter three more times by hitting r three more times (this is why the hotkeys are good to learn). The inverter should return to its original orientation.

> **Note**
> You can also type u for "**undo**" to undo your last action. SUE gives you 100 levels of undo!

Step **9.** Hit the u hotkey to undo the last rotation.

Step **10.** Hit the u hotkey 3 more times. Is that what you expected?

You can also **Move**, **Flip**, **Duplicate**, **Delete**, and **Modify** the inverter. All of these features and more are found in the **Edit** menu. **Edit** menu features work on the selected object or objects. Once again, you can either use the menu or the hotkeys.

Let's try duplicating an icon:

Step **11.** Select an inverter and type d. This will make a *duplicate* copy of the inverter and offset it slightly so you can see it. You are still in duplication mode which you can tell because the cursor is a "pointing hand".

Step **12.** Now click and hold down any mouse button. This allows you to drag the duplicate to where you want it to go.

Step **13.** Finally, release the mouse button to drop the icon in its new resting place.

> **Note**
> Once you are in any mode, such as duplication mode, the **SUE Message Area** tells you what to do next. You can also hit the Space bar to get a full list of available hotkeys and more information.

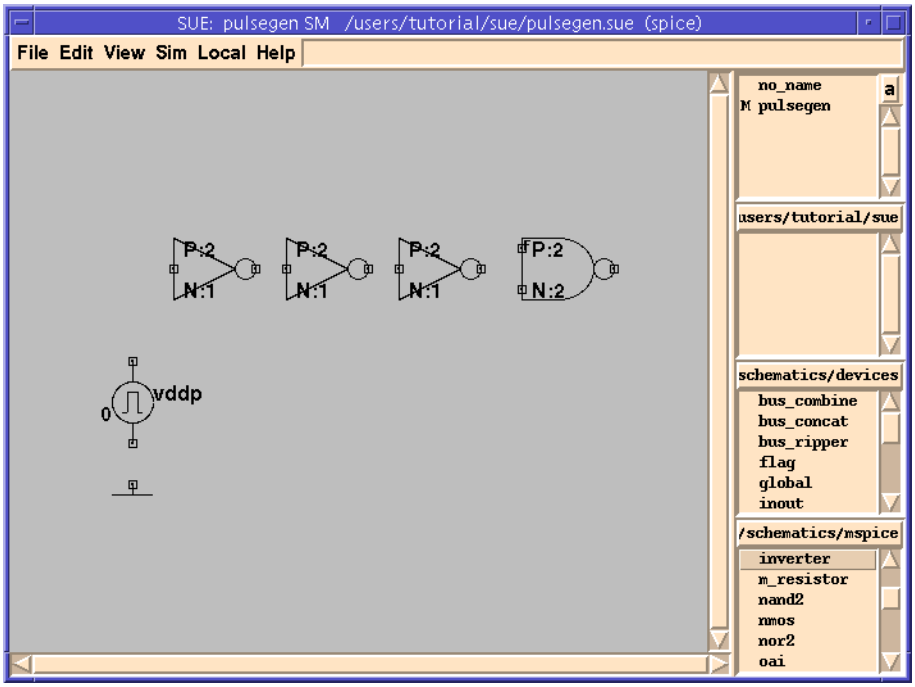
It's time to get back to making our pulse generator.

Step **1.** Remove all the icons except for three inverters. Move them if they aren't already lined up in a row.

Step **2.** Add a "**nand2**" gate (also in the **mspice Library**) to the right of the inverters.

Step **3.** We now need two more icons, the "**pulse**" from the **mspice Library**, and the "**global**" from the **devices Library**. Place them to the left of everything else as in Figure 6.

Figure 6: Icons Placed in Window



Editing Icons

Every icon has **properties** (also known as *parameters*). At the very least, an icon usually has a **name** property so you can specify its name for netlisting. The icons that come from the **mspice Library** are for transistor-level design and have parameters for transistor widths and lengths, also.

Let's edit the transistor sizes of the third inverter to make them larger.

Step 1. Make the right-most inverter *active* (white). It should have a P/N ratio of 2/1.

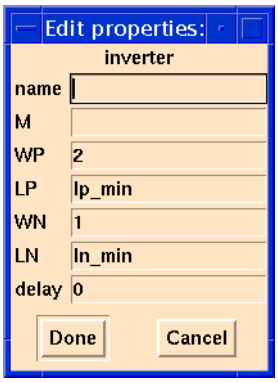
Step 2. Double-click with the left mouse button (Button-1) on the active inverter. This brings up an **Edit Properties** dialog box as shown in Figure 7 and selects the inverter.

> **Note**
>
> If the cursor is a "?" or the **SUE Message Area** displays **"? <message>"** then SUE is waiting for you to close a popup window. You cannot continue with SUE until you close that window, usually by clicking on either the **Done** or **Cancel** button. If you don't see the popup, look at all of your iconified windows or see if the popup got moved behind some other window.

Step 3. Change the value of **WP** (Width of P) from 2 to *8* and the value of **WN** (Width of N) from 1 to *4*. This will quadruple the size of this inverter.

Step **4.** Click the **Done** button. The inverter should now display a P/N ratio of 8/4.

Figure 7: Edit Properties Pop-up for Inverter



Now let's assign the correct name to the **global** icon. The **global** icon turns its attached wire into a global signal like a power supply. For our **pulsegen** circuit we want the **global** to be 'ground' to provide a reference for our pulse generator.

Step **5.** Make the **global** icon *active* and then double-click on it. This brings up a popup.

Step **6.** Type **gnd** into the **Edit Properties** popup for the name property.

Step **7.** Hit **Done**. The **global** icon should now have the label **gnd** on it.

> **Note**
>
> vdd and gnd will be automatically translated by SUE to the correct levels (1 or 0) for IRSIM and Verilog.

The **pulse** icon will generate a string of pulses for SPICE simulation. You can change the duration, amplitude, delay, period, etc. of the pulses by editing the **pulse** icon properties.

Step **8.** Make the "**pulse**" icon active and then double-click on it. This brings up a popup.

Step **9.** Hit **Done** when you are finished reviewing the popup.

Since you have done some work you should now save your circuit.

Step **10.** Do this by selecting **Save** from the **File** menu, or type the hotkey Ctrl-s.

Notice that the **M** disappeared from next to **pulsegen** in the schematic list box. The title bar of the window now has **SW** next to **pulsegen**. This means that this is a **S**chematic, and it has been **W**ritten.

Generators in SUE

Many of the sample gates provided with SUE are generators. A SUE generator takes arguments that can structurally change its icon and/or schematic (unlike parameterized icons) and creates a new icon/schematic. Hence, for every generator, a multitude of cells can be generated, depending on the combination of its arguments.

For example, the "**nand2**" icon is actually a generator. You can edit the number of inputs and/or specify that you want to use a demorgan equivalent icon. The schematics under the icon will also change based on the number of inputs.

Step 1. Select the "**nand2**" gate by clicking on it with the left mouse button.

Step 2. Push into the schematic (hotkey: **e** or from the **View** menu, select **Push Into**.)

Step 3. Look at the schematic. Notice that it has 2 pfets and 2 nfets.

Step 4. Look at the icon view by using the **c** hotkey, or selecting **Display Other View** from the **View** menu.

Step 5. Notice the Verilog property line which should say:

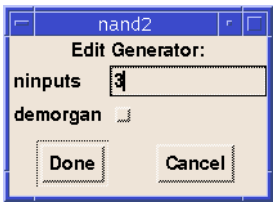
"-type fixed -name verilog -text {assign \#$delay out=!($In0&&$In1)\;}"

This is the Verilog model for a *2 input nand gate*.

Step 6. Pop back up to the **pulsegen** schematic (hotkey: **Ctrl-e** or from the **View** menu, select **Pop Out Of.**)

Step 7. To change the generator properties, hold down the Shift key and double-click on the **nand2** gate with the left mouse button. The pop-up for the **nand2** in will appear, as shown in Figure 8.

Figure 8: Edit Generator Form for nand2



Step 8. Change "**ninputs**" (number of inputs) to ***3*** and click on **Done**. Notice that the icon updates to have 3 inputs.

Step 9. Push into the icon again (hotkey: **e**) and notice that the underlying schematic has changed. There are now a total of 6 transistors.

Step 10. Look at the **icon view** (hotkey: **c**). Notice that the Verilog model is now for a *3 input nand gate*.

Step **11.** Pop back up to the to the **Pulse_generator** schematic (hotkey: Ctrl-e) and undo the change (hotkey: u). The nand gate should now have only 2 inputs.

Wiring and Connectivity

We are now ready to wire up the circuit. First, look closely at the ports of each of the icons. Do you see a hollow square surrounding each port? If not, zoom in until you do. The hollow square, called an '*open*', signifies that the port is *unconnected*.

Ports and wire ends show connectivity with one of three symbols:

- hollow square ('*open*') signifies unconnected
- < nothing > signifies one connection
- solid square ('*solder dot*' or just '*dot*') signifies two or more connections.

Remember to pay attention to this when you wire up the circuit. If you think that you have connected a wire to a port but you missed, the '*open*' won't go away.

Now let's add some wires.

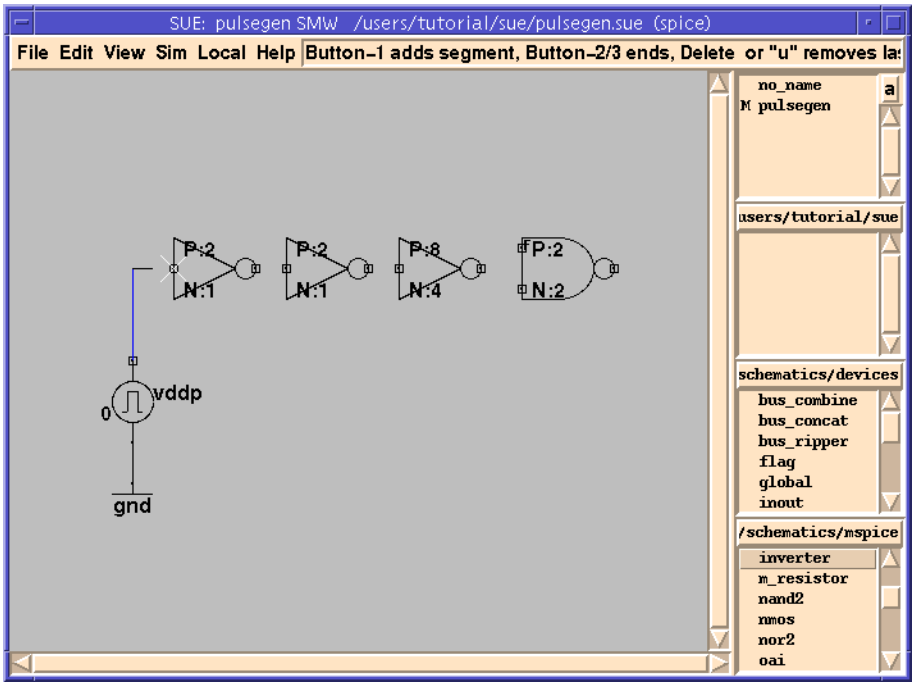
Step **1.** Select **Add Wire** from the **Edit** menu or type the w hotkey. This puts you into "wiring mode".

Step **2.** Look at the **SUE Message Area**. It tells you what the buttons do, or you can hit the Space bar to see what all of the buttons and hotkeys do.

Step **3.** Move the cursor to the port on the top of the **pulse** icon. Click the left mouse button (Button-1) on top of the open. This starts a new wire at that point.

Step **4.** Move the cursor up (vertically) away from the port on top of the **pulse** icon. A wire should follow your cursor. When you get up to about the level of the inverter input port click the left mouse button (Button-1) again and move the cursor to the right. The mouse click should have created an anchor point which allows you to put a bend into the wire. See Figure 9.

Figure 9: Drawing A Wire



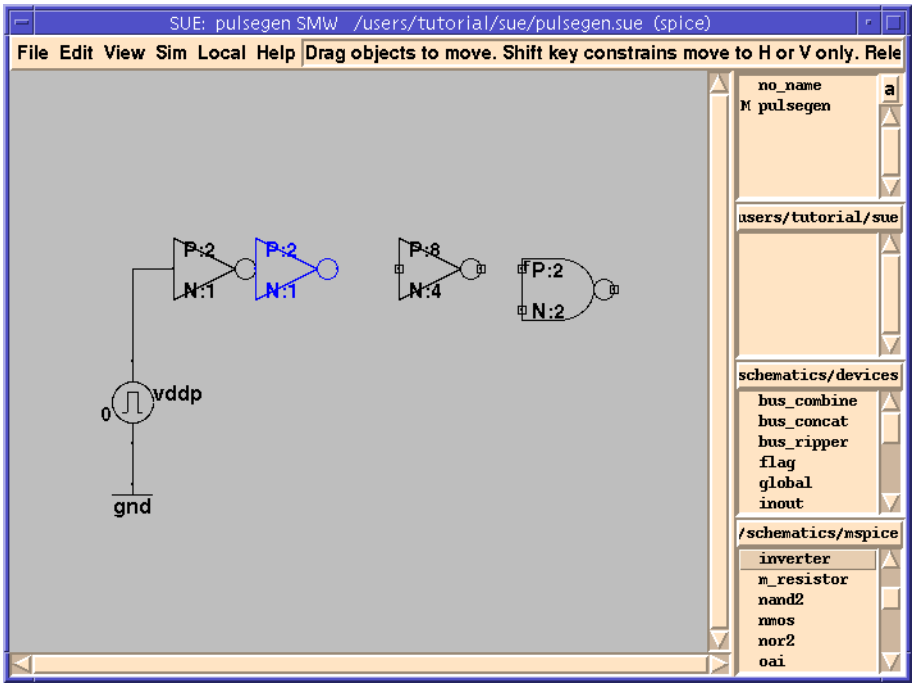
Step 5. When you get the mouse over the left (input) port of the inverter hit the middle mouse button (Button-2) to end the wire. Did the opens go away? If not, you missed. You can either start over with undo or add another wire to finish the connection.

> **Note**
>
> SUE tries to help you wire by placing a white 'X' at the nearest open port to the cursor, as shown in Figure 9. If you like that location then simply double-click with the left mouse button (Button-1) and SUE will wire it up for you.

When you move icons that are connected to wires, the wires stay connected. You can use this feature to add wires by abutting ports from two different icons you want to connect.

Step 6. Move the middle inverter left so that its input port overlaps with the output port of the inverter to its left as shown in Figure 10. To move the inverter, move the cursor until the inverter is *active* (white) and then press and hold Button-2. You are now in move mode and the inverter will follow the cursor until you release the button. Once you release the button, the opens between the two inverters should go away. If not, repeat until they do.

Figure 10: Wiring By Abutment



Step 7. Now, hold down the middle mouse button again and move the middle inverter to the right. Notice that the inverters are wired up. You can use this method of wiring instances as you are placing or duplicating instances.

If you draw a wire that crosses another wire, the two won't connect. In order for them to connect they have to overlap at least one of the endpoints of the wire or at a port. To get two wires to connect, just start or stop the new wire over the old wire.

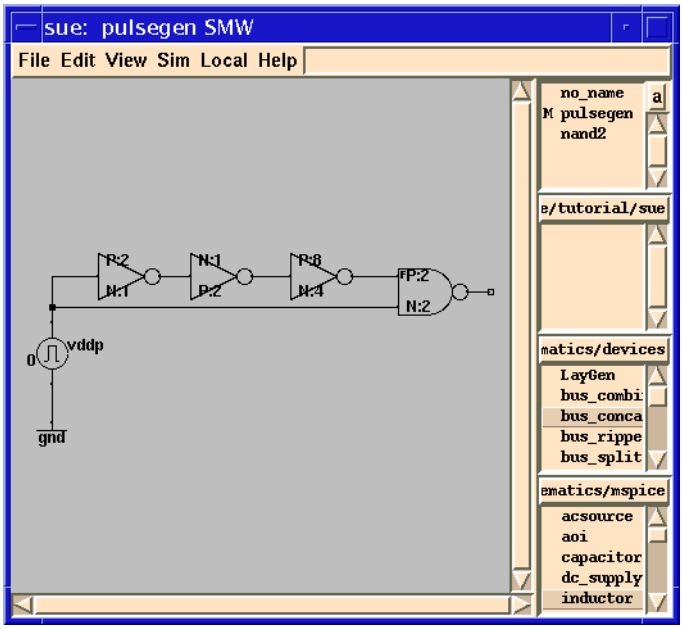
Step 8. Draw a wire starting from the center of an existing wire and draw it perpendicular to the existing wire. Did SUE add a dot to the “T” intersection? If not, try again until a “T” intersection is formed with a dot indicating the connection.

Step 9. **Undo** (hotkey: u) this wire segment.

> **Note**
>
> When you are drawing wires or moving objects, the connection information (dots, opens, etc.) is not recomputed until you are done. For example, when you move an inverter to make a connection, the dot doesn't go away until you release the button.

Step 10. Now wire up the other pieces so that your schematic looks like Figure 11. Don’t worry about the ports yet; we will add them later in the tutorial.

Figure 11: Sample SPICE Schematic



Note

If at anytime you are in a mode and something goes wrong, you can type Ctrl-c to abort that mode. For example, if your wire isn't going the way you want it to, typing Ctrl-c will abort that wire.

Step **11.** Before going on to the next section, save your schematic (hotkey: Ctrl-s).

Part 3
Simulating Circuits With SPICE And IRSIM

Simulating Circuits

The main purpose of creating schematics is to netlist, simulate and verify them. In SUE this is a snap! In the next few sections you will get a chance to simulate the same circuit in three different simulators: HSPICE, IRSIM, and Verilog.

Running HSPICE

If you don't have HSPICE installed, skip to the “Running IRSIM” section. If you have never run HSPICE before, don’t worry.

First, make sure the following two things are set up:

1. You have a circuit loaded into SUE that you want to simulate. The “**pulsegen**” circuit you just entered will work fine. If you just started SUE again, load the circuit with the **Open** command in the **File** menu. Your schematic should look like Figure 12-a.

2. Insure that you are in SPICE mode. On the title bar of SUE you should see ‘**(spice)**’ which means you are in SPICE mode. Otherwise, go to **Change Simulation Mode** in the **Sim** menu and click on **spice** and then **Done**.

OK, pay close attention. To run HSPICE you have to do the following set of complex tasks:

3. Type h

   That's it!

This single command (**SPICE it** in the **Sim** menu) does the following:

- First, it creates an HSPICE netlist of the current schematic which is the same as the hotkey Shift-n or **SPICE Netlist** in the **Sim** menu.

- Second, it runs a script that starts HSPICE either on your local machine or on a remote server depending on the configuration (assuming the script is set up).

- When the HSPICE job is finished, it starts the Micro Magic Tools NST Waveform Viewer (included with SUE) and loads the SPICE data file (the `.tr0` file) into it. This can also be done by Ctrl-i or **Init Probe** in the **Sim** menu.

> **Note**
>
> If you have two monitors on your machine, you can have SUE start NST on the other monitor by typing the line "`setenv PROBE_DISPLAY other`" at the UNIX prompt before starting SUE.

Displaying HSPICE Waveforms From SUE

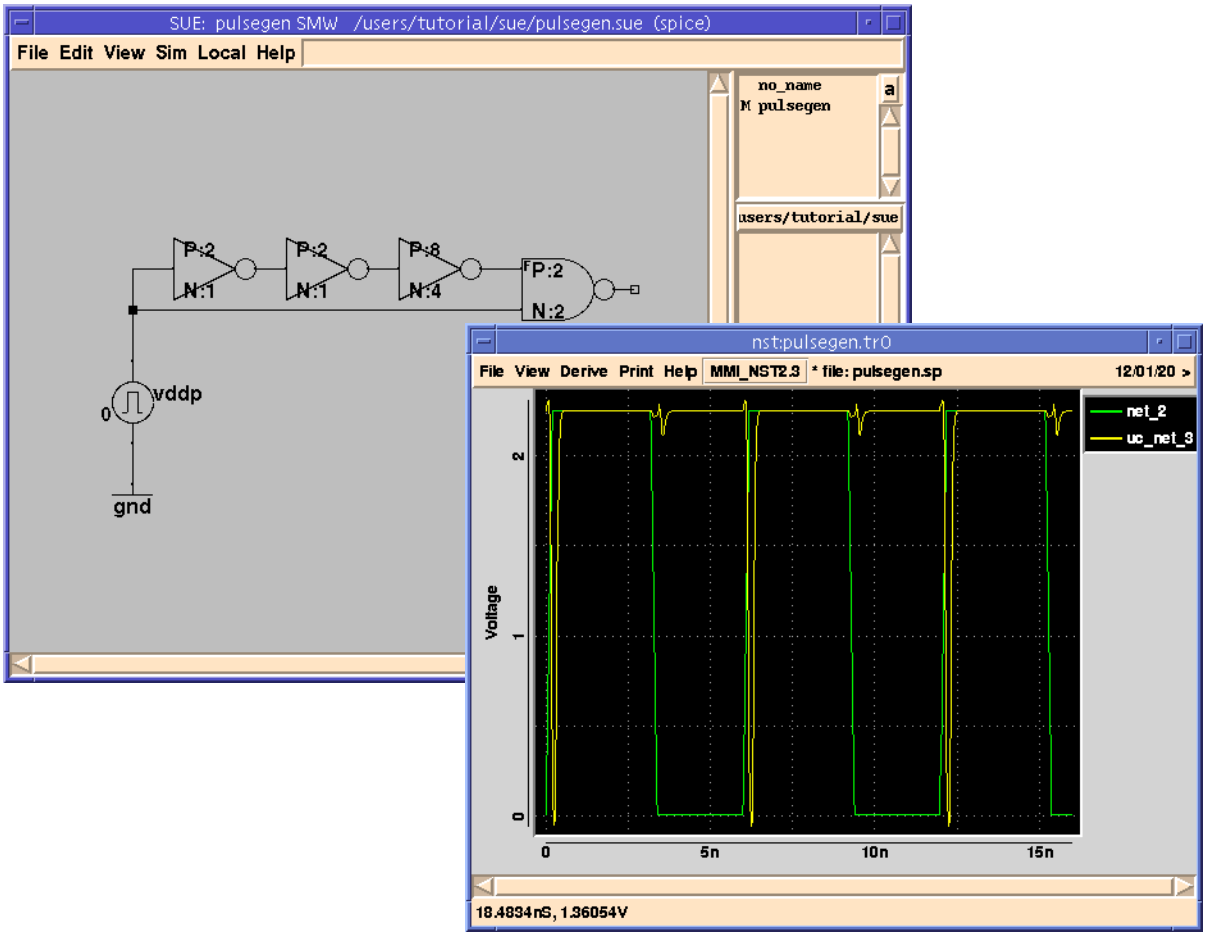
Now you will start to see some of the real power of SUE.

To see how your circuit simulated, you would simply select one of the wires in your schematic and then type p or **Plot Net** in the **Sim** menu. The waveform for that wire should show up in the NST window.

Step 1. Select the wire at the top of the **pulse** icon and hit **p**.

Step 2. Select the wire at the output of the **nand2** gate and hit **p**. You might have to carefully select the 'open' on the output of the **nand2** gate if you didn't add a wire there. Do you see the pulse output? Remember the pulse is low going because we have a **nand2** instead of an **and2**. Your waveforms should look like Figure 12-b.

Figure 12: Displaying HSPICE Waveforms: a) “pulsegen” Schematic; b) NST With Two Waveforms



Note that if you changed any of the parameters for the **pulse_icon**, your waveforms will look different.

There are lots of plotting options in the **Sim** menu such as **Plot Old Net, Unplot Net, Plot Net & Remember**, etc. **Plot Old Net** will plot the same net from the last HSPICE run. So you can tweak a device size or change a gate, rerun HSPICE with h, and compare the old and the new waveforms!

> **Note**
>
> NST can do a lot more than just plot waveforms. You can add panels, print, compute power, and do arithmetic derivations on waveforms! (See the NST manual for details).

You can plot net waveforms at any level of the hierarchy just as easily as at the top. SUE does all of the messy translations for you.

Step **3.** In this circuit there are only two levels of hierarchy: the one we are looking at, and the transistor level circuits of the inverters and nand2. Push into the **nand2** gate (select it and hit **e** or **Push Into** in the **View** menu).

Step **4.** Now select the net between the 2 series nfets and hit **p** to plot its waveform.

Step **5.** Pop back out to the **pulsegen** circuit with Ctrl-**e** or **Pop Out Of** in the **View** menu.

> **Note**
> You can always go directly to a schematic without pushing or popping by clicking on its schematic name in the **Schematic List** box. However, SUE will assume that the schematic you went to is at the top of the hierarchy so you should use push and pop whenever you can.

Running IRSIM

We now are going to run IRSIM on your same circuit.

IRSIM is a switch level simulator from Stanford University. We have included it with SUE. In any other schematic editor you would have to draw a different schematic to run IRSIM, but not with SUE. SUE can run SPICE, IRSIM, and Verilog all from the same schematic!

To run IRSIM, we first need to switch simulation modes:

Step **1.** Select **Change Simulation Mode** in the **Sim** menu and then click **sim** and then click **Done** or hit Return.

If you had an NST window up from the previous section, SUE closes it since IRSIM comes with it's own waveform viewer (called Analyzer).

Step **2.** Pull down the **Sim** menu. You will see that it has changed since you changed the simulation mode. But only the **Sim** menu changed. Note that the first entry used to be **SPICE Netlist** and it is now **SIM Netlist**, but the hotkey is the same.

Step **3.** Now type **h** to run IRSIM (or **SIM it** in the **Sim** menu).

SUE creates a SIM netlist for your schematic (**SIM Netlist**) and then starts IRSIM in interactive mode with the SIM netlist and brings up the waveform analyzer (**Init Probe**).

Interacting With IRSIM

You are now running IRSIM and the time is “0ns”. Furthermore, no inputs are set. In SPICE we set up a driving function with the **pulse** icon. In IRSIM (or Verilog) mode, this type of simple waveform is usually too simplistic and thus doesn't netlist to anything.

Let's setup an input:

Step **1.** Select the wire at the top of the **pulse** icon. This is the one we want to drive.

Step **2.** Hit hotkey **0** (zero) or **Irsim Set Low** in the **Sim** menu. This will tell IRSIM to set that net to a ‘logical 0’.

Since this is the only input to the circuit we can now “step time” forward.

Step **3.** Hit hotkey s or **Irsim Step** in the **Sim** menu.

Step **4.** Plot the selected net with the p hotkey, just like in SPICE mode. The timing waveform should appear in the analyzer window.

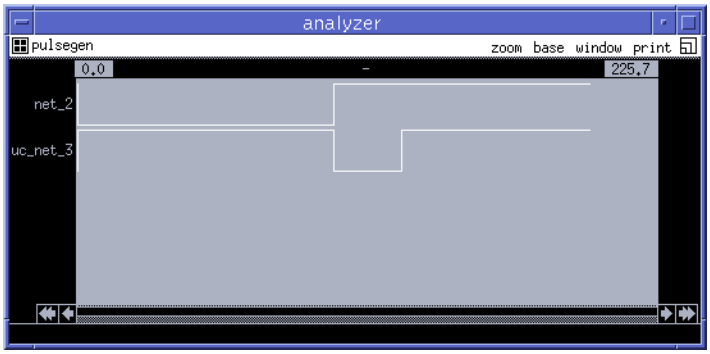
Let's look at another node:

Step **5.** Select the **nand2** output and type p. Notice that the **nand2** output went high after the input went low.

Step **6.** Now go back to the input and set it high by selecting it and typing **9** (or choose **Irsim Set Hi** from the **Sim** menu). Step time again by typing s.

Step **7.** Notice that the **nand2** output went low, and then some time later back high. Yep, we actually have a pulse generator. Your waveforms should look something like Figure 13.

Figure 13: IRSIM Waveform With Two Nets



Net Names When Netlisting

Did you notice that you haven't named a single net, however, you just finished simulating a schematic in two different types of simulators? In SUE, you only need to name ports and globals. SUE will create unique names for all other unnamed nets.

There are three ways to name a net in SUE:

- By connecting the net to an I/O port with either the “**input**”, “**output**”, or “**inout**” icons.
- By connecting the net to a global signal name with the “**global**” icon.
- By adding a “**name_net_s**” icon to a net and labeling it with a name. (Do this by double-clicking Button-1 on the icon, and then entering a net name.)

Otherwise, SUE makes up a name for the net of the form: "**net_#**". SUE will also make up unique names for devices and icons if you don't name them.

> **Note**
>
> If you have trouble double-clicking on the small "**name_net_s**" icons to name them, you might use the **Edit Selected** command in the **Edit** menu, or first name them and then move them onto the desired net.

If you want to see what SUE has named a wire, just select it and look at the **SUE Message Area**. Note that you have to netlist first to see the net name.

Part 4
Higher Level Schematics And Verilog Simulation

Higher Level Schematics

Verilog

We have already simulated our little pulse generator in IRSIM and in HSPICE. Now we want to run a Verilog simulation. Unfortunately, you can't simply set nodes in Verilog like you can in IRSIM. You need to have a test file to drive them. To accomplish this we will first make an *icon* for our cell, and then wire our icon up to a clock generator that has been provided for you.

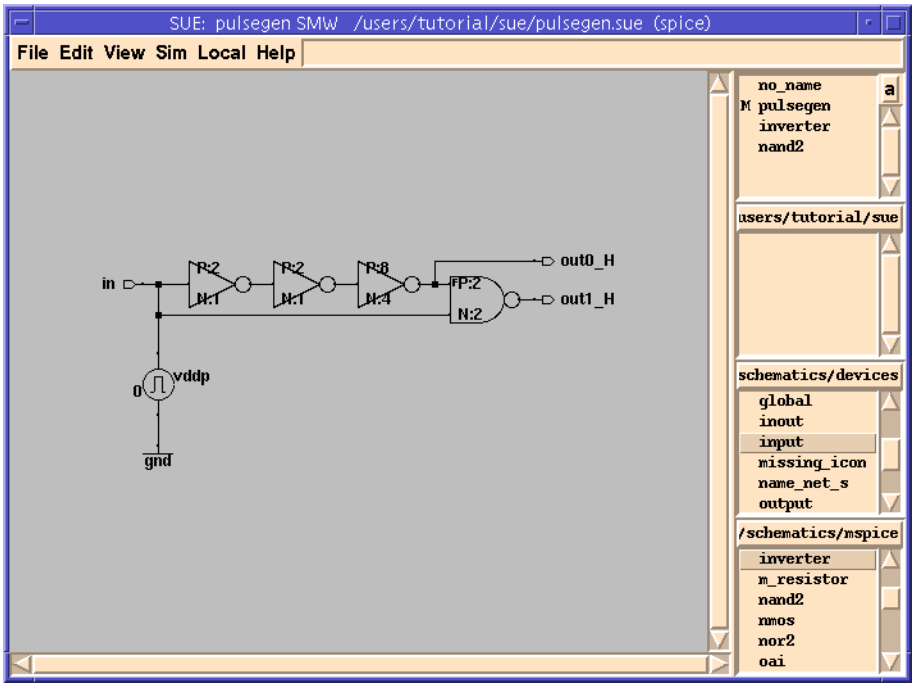
Adding Ports

Before we make our icon we need to add some I/O ports to our circuit. We are going to place and name them as shown in Figure 14.

Step 1. To attach the I/O ports, start by getting an **input** from the **devices Library** and attach it to the wire above the **pulse**. Remember, click with the left mouse button (Button-1) to select or get an icon from an **Icon List** box, and use the middle mouse button (Button-2) to move an instantiated icon.

Step 2. Next, attach an **output** icon to the output of the **nand2**. Since we would like to look at more than one output, duplicate the output port (**select**, then d) and wire it up to the output of the third inverter as shown in Figure 14.

Figure 14: Adding Ports to a Schematic



Step 3. Now name the I/O ports by double-clicking on the input port to call up the **Edit Properties** dialog box.

Step 4. Type "`in`" into the name box, then click **Done** or hit Return.

For the outputs we will use a little trick that allows us to name multiple signals. It's not much for only two I/O's, but when you have 32, it helps a lot.

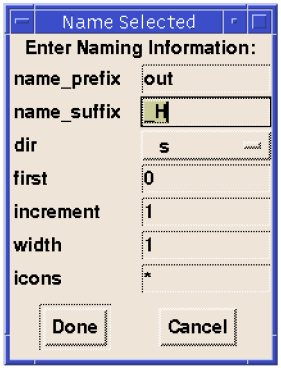
Naming Multiple Signals

Step 1. Select both of the `output` icons. You can do this by clicking the left mouse button on one of the outputs and Shift-clicking (hold down the Shift button while clicking with the left mouse button) on the other output. Or you can simply hold down the left mouse button (Button-1) and drag the selection box around both icons. Don't get any other icons (like the **nand2**) or you will name those also.

Step 2. Go to **Name Objects** in the **Edit** menu. The **Name Selected** dialog box (see Figure 15) will appear. Type "`out`" in the **name_prefix** box. Type "`_H`" for **name_suffix**. Change the direction from north (n) to south, "`s`". When you are finished click **Done** or hit Return.

Step 3. Save your circuit (hotkey: Ctrl-s).

Figure 15: Name Selected Popup

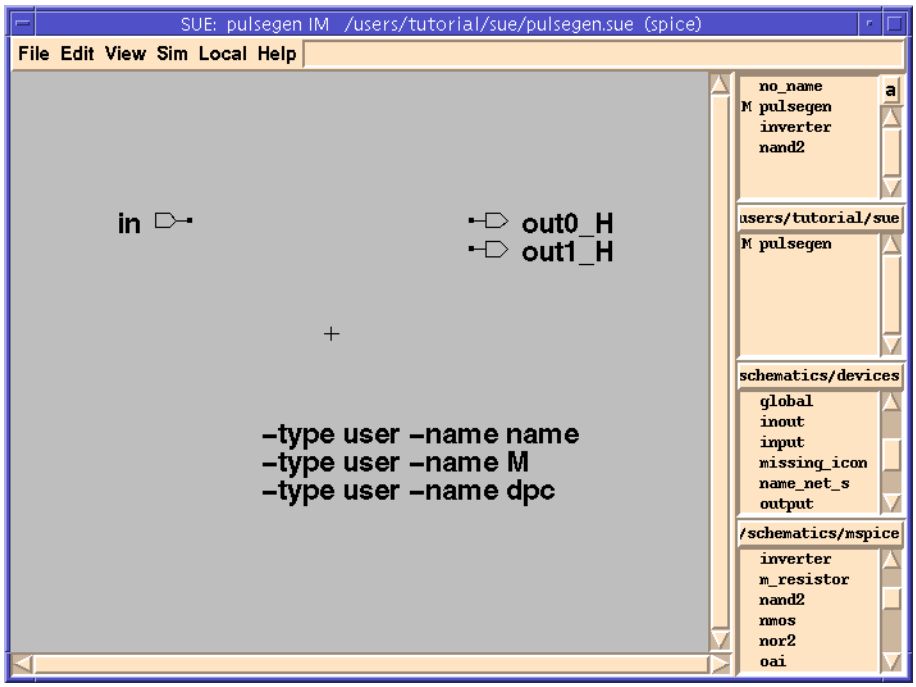


Making An ICON For Our Schematic

We are now ready to create an icon for the pulsegen schematic. Schematics only need icons when you want to instantiate them into other schematics -- which is what we are going to do.

Step **1.** To make an icon, select **Make Other View** from the **View** menu (or type Shift-c). This creates an icon for **pulsegen** and puts you into that icon view. It should look like Figure 16.

Figure 16: Icon View



You should now be looking at the icon view of **pulsegen**. You can tell it's an icon since the title bar of the SUE window now has as '**I**' instead of an '**S**' in it. Also, since **pulsegen** now has an icon, it appears in the **Icon List** box for this directory.

Only the following elements can be placed in icons, which are essentially just pictures:

- I/O's: *inputs*, *outputs*, and *inouts*. At the locations of the origins of these icons, small solid dots will be displayed when the icon is placed into a schematic. No other icons can be placed into an icon (except a **title_bar** which is for documentation only).

- The origin of the icon, which is notated with a "**+**". All placement, rotation, flipping, etc. of the instantiated icon is relative to this origin. Schematics don't have origins, only icons do. The only thing you can do to the origin is move it around.

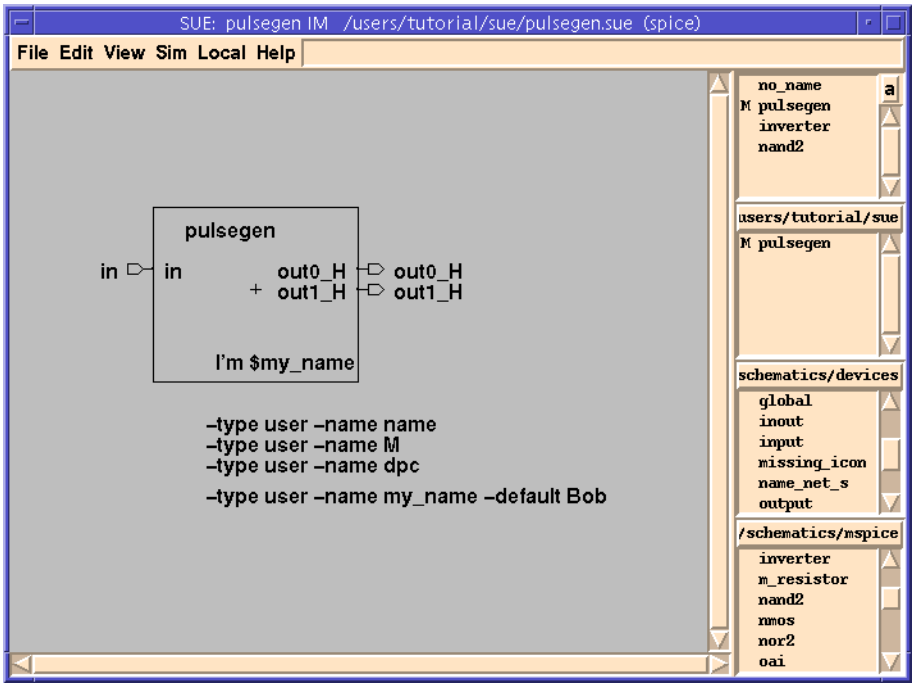
- Property Definitions, which are text lines that begin with a dash ("-"). Don't add text to an icon starting with a dash unless it is a **`Property Definition`**. The *`Name Property`* for the icon, for example, is defined by the line "**-type user -name name**" and will then appear in the **Edit Properties** dialog box if you double-click on an instantiation of this icon (discussed more fully below).

- Picture elements, like *lines*, *arcs*, and *text*. Don't add wires to icons.

When SUE makes an icon, it places all of the I/O's from the schematic into the icon, with inputs in alphabetical order on the left, and outputs on the right. It also adds some default **`Property Definitions`** (**name, M,** and **dpc** in this case).

Now we want to draw a picture of what we want our pulsegen icon to look like when we place it into another schematic. For this example, we will just draw a box with some text in it. We are going to make our icon look like Figure 17:

Figure 17: Icon for Pulse Generator



Drawing the ICON

Step 2. Start by drawing a box. You can make a box out of multiple lines, or by drawing a rectangle. Both generate the same thing in the SUE database, so use the one you find easiest.

a. *To draw a box using lines*, use the `l` hotkey, or select **Add Line/Rect** from the **Edit** menu. Click with the *left* mouse button (Button-1) to start your line and to add new line segments. Click the middle mouse button (Button-2) to end it.

b. *To draw a rectangle*, use the `l` hotkey or **Add Line/Rect** from the **Edit** menu. Then click and hold down the *middle* mouse button (Button-2) and drag out the area you want for your rectangle.

Notice the instructions which appear in the **SUE Message Area** (to the right of the **Help** menu) when you begin to draw.

Step **3.** You can resize your rectangle by clicking and holding down the left mouse button (Button-1) on one of the squares (called *edit markers*) in the corners of your rectangle, then dragging it to where you want the vertex to be. If you hold down the Shift key, the entire side will move.

> **Note**
> If you need to resize the rectangle later, double-click with the left mouse button on the line or rectangle, and the edit markers will reappear.

Adding Text

Let's add some text to our icon. First, we will type in the name of the cell, so we know what the cell is when we look at a schematic with this icon instantiated in it.

Step **4.** Add text with the t hotkey or by selecting **Add Text** from the **Edit** menu. Move the cursor to where you want the text to start and click the left mouse button (Button-1). Notice that directions are provided in the **SUE Message Area** to the right of the **Help** menu.

Step **5.** Now, simply type in "**`pulsegen`**" and hit Return. You can use standard editing commands like the arrow keys, backspace, delete, etc. Emacs users will find that simple Emacs commands also work.

> **Note**
> To change the font size of the text you must be editing the text, which you can return to by double-clicking (Button-1) on the text. Button-2 then brings up a menu of text sizes. You can also change the text anchor with Button-3.

As mentioned earlier, the I/O icons get replaced with small solid squares. To help identify these ports, we should add text to them. You can add text to each of them individually, which can get tedious. To make this easier, SUE has a special feature for this called "**Duplicate Text**".

Step **6.** Select the **input** icon labeled "**in**".

Step **7.** Hit the hotkey Shift-t or **Duplicate Text** in the **Edit** menu. The text from the selected icon is duplicated out of the icon, and you now use Button-1 to move it to where you want it just like in normal duplication mode. Move the "in" text to the inside of the icon rectangle.

Step **8.** Now select both of the `output` icons together and hit Shift-t. Move the text inside the rectangle as shown in Figure 17.

In order to personalize our icon, let's add a new property called "**`my_name`**":

Step **9.** Select the text "**-type user -name name**" and duplicate it (d). This is just to save some typing. Move the duplicated text underneath the other text.

Step **10.** Double-click on the copy and change it to read:

-type user -name **my_name** -default **Bob**

You just defined a new property called "**my_name**" and set its default value to "**Bob**". The "**my_name**" property will now show up in addition to the other properties when you double-click on the instantiated icon.

Once you've defined a property, you want to use it. The simplest way to use a property is to display it. The way you display a property is by preceding it with a "**$**" in a text line.

Step **11.** Hit **t** to **add text**, and click Button-1 inside the icon rectangle. Now type the line:

I'm $my_name

and hit Return.

Step **12.** Now type c or select **Swap Views** in the **View** menu to get back to the schematic.

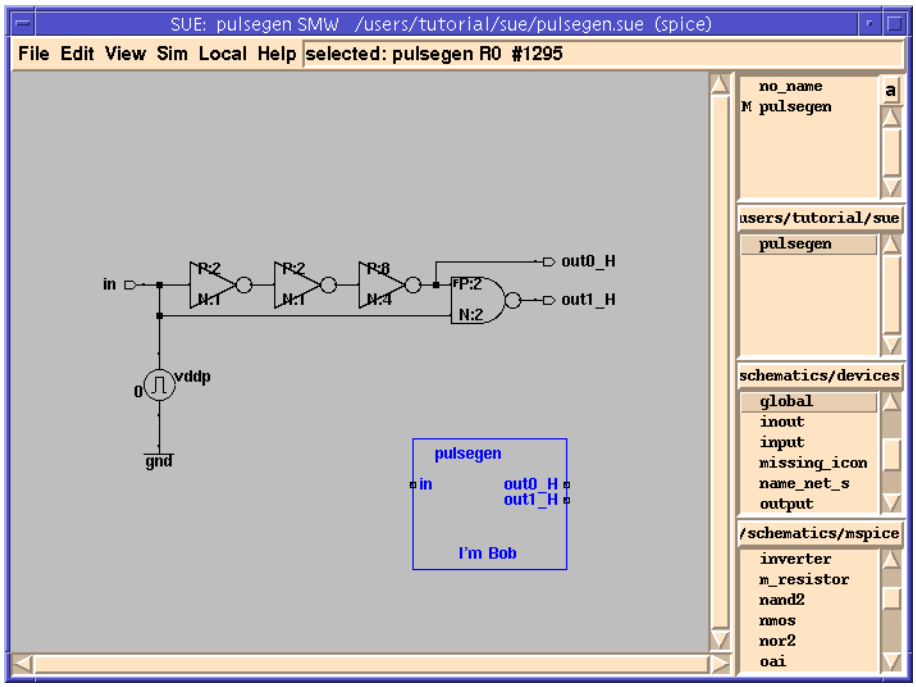
Placing Our ICON Into The Schematic

We are now going to place a copy of the "**pulsegen**" icon into the "**pulsegen**" schematic. This is handy for documentation since we can see both the schematic and the icon that goes with it at the same time. Don't worry, SUE is smart enough to know it's the icon for this schematic and to avoid recursive calls when netlisting.

Step **1.** Select "**pulsegen**" from the **Icon List** box (the second list box from the top).

Step **2.** Place the icon toward the bottom of your schematic, as shown in Figure 18.

Figure 18: Schematic with Icon



Modifying The ICON

Just in case your name isn't Bob (yes, I know there are lots of you) let's change the value of '`my_name`' in this instantiated **pulsegen** icon.

Step 1. Double-click on the **pulsegen** icon and change "*Bob*" to your name in the **Edit Properties** dialog box.

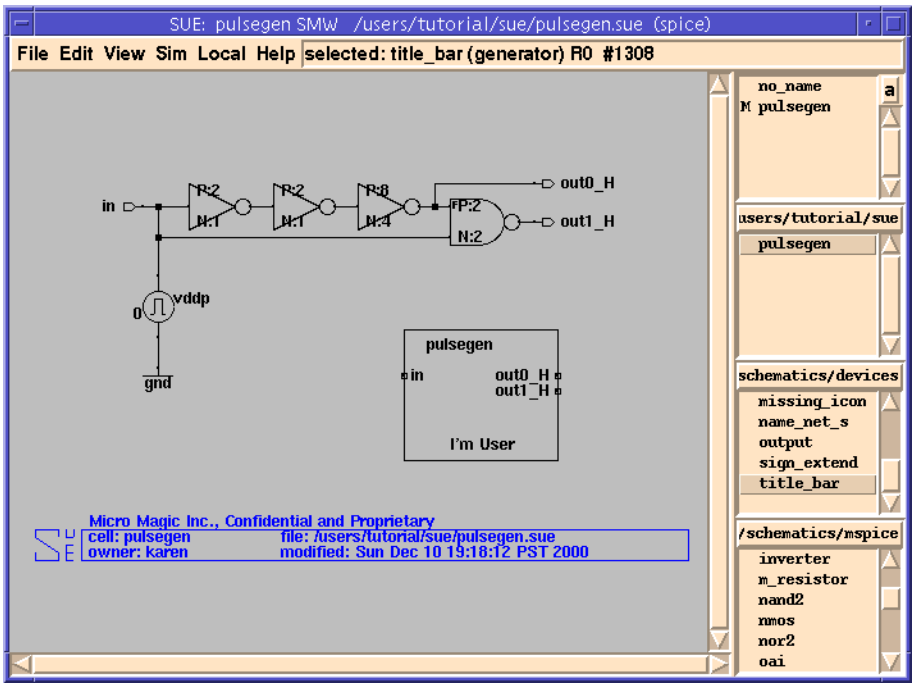
The icon should now say "`I'm <your name>`". Notice that SUE automatically added the property "`my_name`" and the default "`Bob`" to the **Edit Properties** dialog box.

Finally, let's add a **title bar** to our schematic. The title bar is another documentation feature. It automatically includes the schematic name, file name, date modified, and owner to the schematic. You can easily customize it to include your company name, too.

Step 2. Select the "**title_bar**" icon from the **devices Library** and place it at the bottom of your schematic.

Your schematic should now look like Figure 19. (The actual text in the title bar will be different than the illustration.)

Figure 19: Schematic With Title Bar



> **Note**
> The "**title_bar**" icon is a generator. If you Shift-double-click on it and change it to an "**sccs_title_bar**", then SUE will put your cell under SCCS control, also, when you save it.

Step 3. Save your schematic with Ctrl-s.

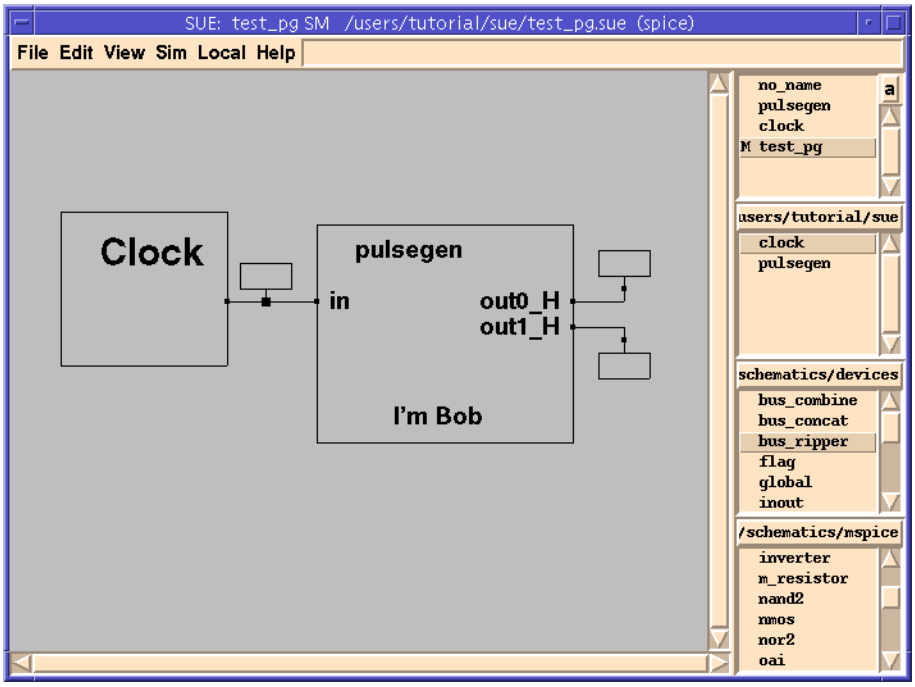
Using Pulsegen

We are now going to build a test circuit to run our pulsegen.

Step 1. Open the file "**clock.sue**" (Ctrl-l or **Open** in the **File** menu). This is the clock generator we already made for you.

Step 2. Make a new schematic with Ctrl-n or **New Schematic** in the **File** menu. Call the schematic "**test_pg**". This new schematic is going to look like Figure 20.

Figure 20: Test_pg Schematic



Now build the circuit.

Step 3. Drop a "**pulsegen**" icon

Step 4. Drop a "**clock**" icon

Step 5. Drop three "**flag**" icons from the **devices Library**. Attach them to the outputs of the "**clock**" and the "**pulsegen**" icons (refer to Figure 20 if needed). We will describe these later.

Step 6. Wire it up.

You are now ready to simulate your circuit in Verilog.

Step 7. Change simulation mode to Verilog (**Change Simulation Mode** in the **Sim** menu, click on **Verilog** and hit **Done** or Return).

Step 8. Hit Shift-n or **VERILOG Netlist** from the **Sim** menu. Notice that the **Sim** menu changed again since we changed the simulation mode to Verilog.

You just built a Verilog netlist using our "**pulsegen**" and a behavioral model of "**clock**" (which has been provided).

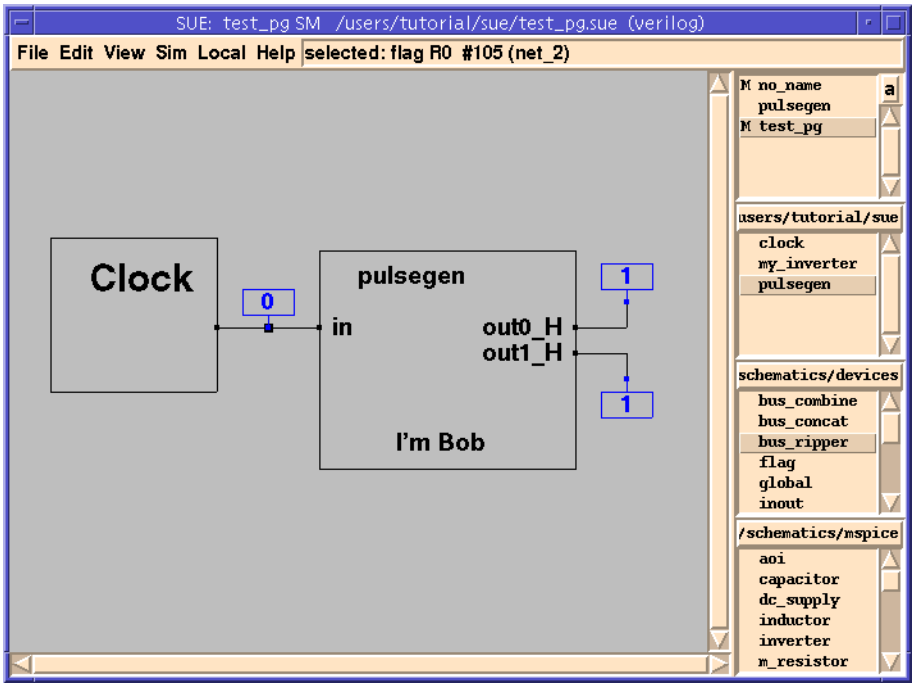
Now to run the simulator:

> **Note**
>
> If you don't have a Verilog simulator set up, skip the rest of this section. Continue the tutorial at "Creating Behavioral Verilog Models".

Step 9. Hit Ctrl-i or **Init Probe** in the **Sim** menu. This starts up an interactive Verilog simulation.

Step 10. Type s or **Step Verilog** in the **Sim** menu to step the time forward. This sends several cryptic lines to your Verilog simulator, which causes it to advance the time step 10 units. A "**0**" should appear in the flag nearest "**clock**", and a couple of "**1**"'s on the output flags, as in Figure 21.

Figure 21: Step Verilog Shows Changes in Flags



You probably figured out what the flags are by now. They report the state of the nets they are attached to and are updated after every time step.

Step **11.** Step time a few more times.

The clock is designed to transition every other Verilog time step. Notice that the "**out0_H**" flag changes every cycle and is the inverse of the clock. This makes sense since it is just the clock followed by three inverters. But what happened to our pulse on "**out1_H**"?

In HSPICE and IRSIM all transitions take a finite time, unlike Verilog where you have to specifically add delays to transitions. What's happening is that the inverters are taking no time (or 0ns), so in Verilog our "pulse" lasts for 0ns! To fix this we will change the inverter model to add a delay.

Creating Behavioral Verilog Models

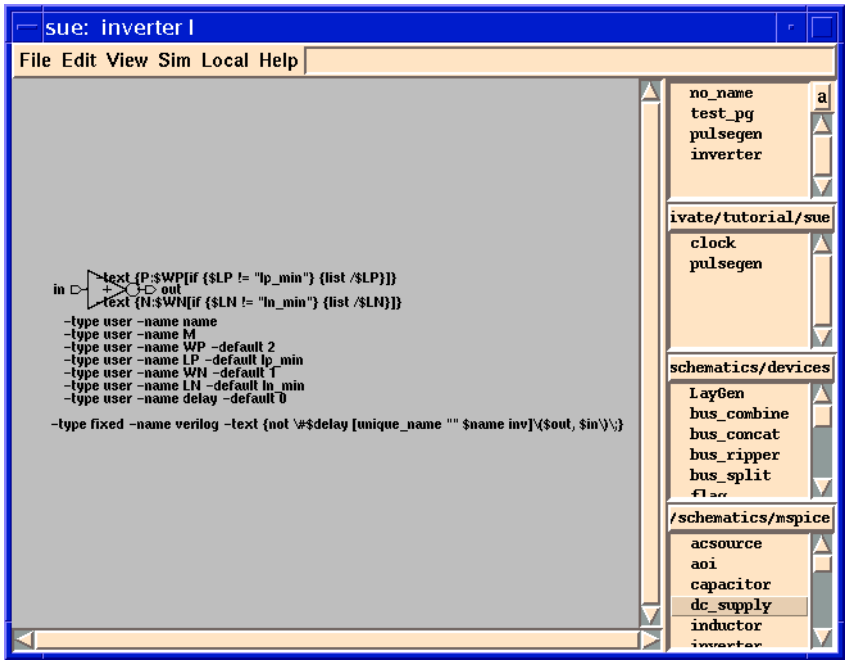
We are now going to edit the schematic so that the inverters have a non-zero delay.

Step **1.** Select and push into "**pulsegen**" (hotkey: e).

Step **2.** Select and push into an inverter (hotkey: e).

Step **3.** Now type c or **Swap Views** in the **View** menu to switch to the icon view of "**inverter**".

Figure 22: Changing Name To “my_inverter”



Step 4. Look at the Verilog property for the inverter:

-type fixed -name verilog -text {not\#$delay[unique_name]""$name inv]\($out,$in\)\;}

The text you just edited tells the netlister how to netlist this cell when in Verilog Mode. Notice the “**#$delay**” in the verilog string above. This adds the value of the delay property to the inverter. Because it is a user property, it shows up in the **Edit Properties** popup for the icon. Also, the actual net names are substituted for “**$out**” and “**$in**” during netlisting.

Step 5. Pop back up to **pulsegen** (hotkey: Ctrl-e).

Step 6. Double click with Button-1 on each of the inverters and change the delay from **0** to **4**.

Step 7. Save **pulsegen** with Ctrl-s.

Now let's get back to our pulse generator test circuit, “**test_pg**”, and simulate it again to see if we get a pulse.

Step 8. Pop back up to **test_pg** (hotkey: Ctrl-e).

Step 9. Re-netlist by hitting the Shift-n hotkey or **VERILOG Netlist** in the **Sim** menu. Then do an **Init Probe** (Ctrl-i) to restart the simulator.

The previous simulation will be stopped and a new one will be started.

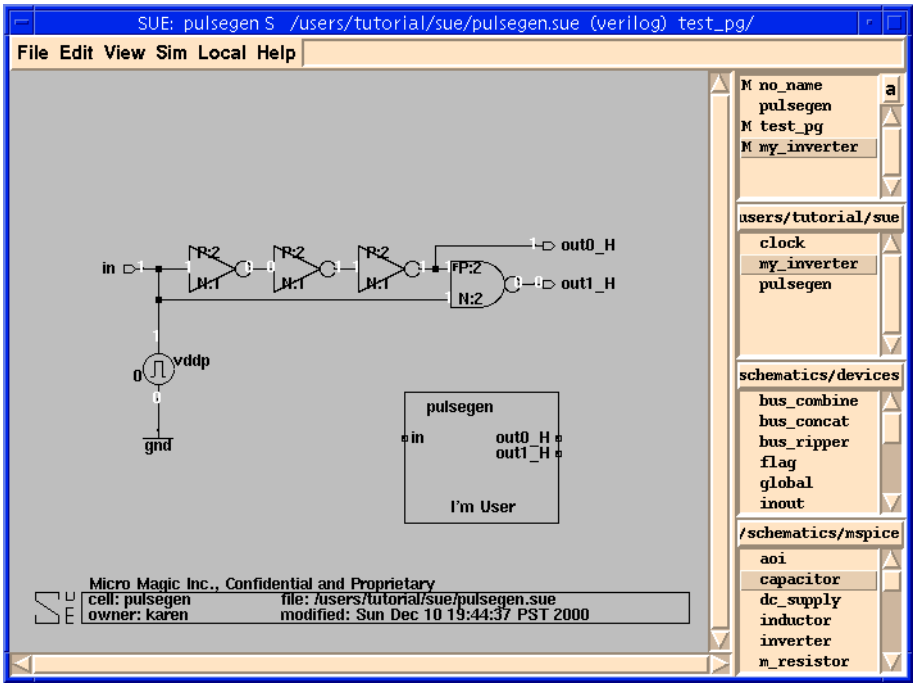
Step 10. Step time (s) a few times and see what happens.

You should see that “**out0_H**” follows “**in**”, but with a one-cycle delay. Also, now “**out1_H**” generates our pulse! (negative going).

11. Select the **pulsegen** icon and push into it (e).

Hit the Ctrl-o hotkey or **Display Term Values** in the **Sim** menu. Do you see the white 0's and 1's? Look at Figure 23.

Figure 23: Display Term Values



In addition to the flags, you can have SUE display all of the terminal values in a given schematic with **Display Term Values**. This is great for debugging!

Creating A More Complex Verilog Model

For the inverter in the above example, the Verilog model could be described in a single line. If you need to write a more complex Verilog model, you can attach a Verilog text file to a schematic. Just because SUE looks like a schematic editor doesn't mean that everything you do has to be with schematics.

Instead of modeling just the inverter, this time we'll model the entire **pulsegen** circuit from the above example.

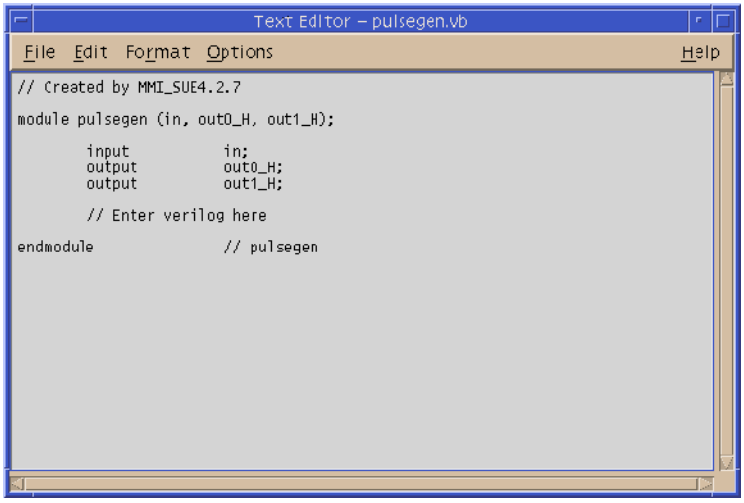
Step **1.** Return to the **pulsegen** schematic if you aren't already there. Make sure none of the icons are selected.

Step **2.** Hit the Shift-e hotkey or **Edit Verilog** in the **Sim** menu. A dialog box will come up telling you that it can't find a Verilog file for "**pulsegen**". Simply hit the **Done** button and SUE will make one for you, and then bring up the file in your favorite text editor (EMACS, VI, etc.).

> **Note**
>
> SUE chooses the editor to use based on your *EDITOR* UNIX environment variable. For example, to set to *Emacs*, type "`setenv EDITOR emacs`" in UNIX, and restart SUE.

Your text editor file will look like Figure 24.

Figure 24: Template for Verilog Model



Notice that SUE automatically creates a template for the Verilog model, adding the port information to the file. You only need to add the model for "**pulsegen**" after the line:

```
// Enter verilog here
```

Instead of creating the **pulsegen** Verilog model, please wait for the next section, which has larger and more interesting example and includes Verilog models at many levels.

Step **3.** Save the file and close the text editor window.

- If your editor is Emacs (the default), type Ctrl-x, Ctrl-s to save, followed by Ctrl-x, Ctrl-c to exit.
- In VI type :wq to save the file and quit.

SUE keeps track of your Verilog file and 'attaches' it to your schematic. You can view or edit it any time by simply selecting **Edit Verilog** from the **Sim** menu, as you did before. Since the file exists now, SUE will bring it up for you automatically. You can do this either from the schematic, or by selecting the icon in a higher level schematic.

Notice that SUE has named your behavioral Verilog file "**`pulsegen.vb`**" (note the "**`b`**"). If you want to write and attach your own behavioral Verilog files, they must have the "**`.vb`**" extension. Otherwise, SUE won't find them.

You may have also noticed that when SUE creates a Verilog netlist, it gives us the extension of "**`.vh`**". Why doesn't SUE just use the extension "**`.v`**" like everyone else? Simple, so SUE doesn't step on your existing "**`.v`**" Verilog files. The only time SUE uses the "**`.v`**" extension is to import port information from a user-written Verilog RTL file using **Load Verilog I/O's** in the **Sim** menu.

Before going on to a more interesting example, let's recap what you have done so far.

You have:

- Drawn a schematic and built its associated icon.
- Simulated the schematic in HSPICE, IRSIM, AND Verilog.
- Learned lots of SUE features.

And you have done all of this from inside of SUE, without once having to run any other tools externally.

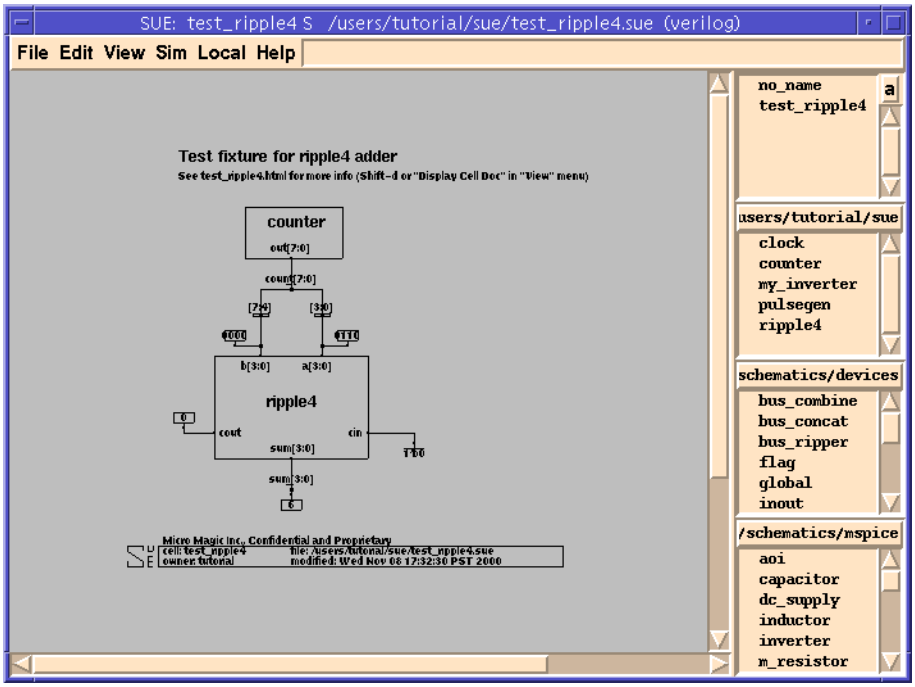
Not bad, huh?

A More Complex Example

Now let's look at something a little more complex and which has a few levels of hierarchy. The circuit is a simple 4-bit ripple-carry adder.

Step **1.** Open the file "**test_ripple4.sue**" that is in the tutorial directory. You should see something similar to Figure 25.

Figure 25: Demo “4-bit ripple-carry adder” Schematic



This example has text documentation along with it as described in the text in the schematic. Let's look at it.

> **Caution**
>
> If you are reading this tutorial in a browser and bring up the documentation, it will replace this tutorial. After reading the documentation, just hit the “Back” arrow in your browser to return to this tutorial.

Step 2. Hit Shift-d or **Display Cell Doc** in the **View** menu to view the attached documentation for this cell.

You can also attach plain text files, in place of HTML files. SUE looks for the “.html” or “.doc” extensions to files with the same names as the schematic you are editing. It brings up “.html” files in your browser, and “.doc” files in your text editor.

Step 3. Now look over the **test_ripple4** example. Push into all of the cells. It has documentation, buses, and lots of other interesting things not covered in your simple **pulsegen** example.

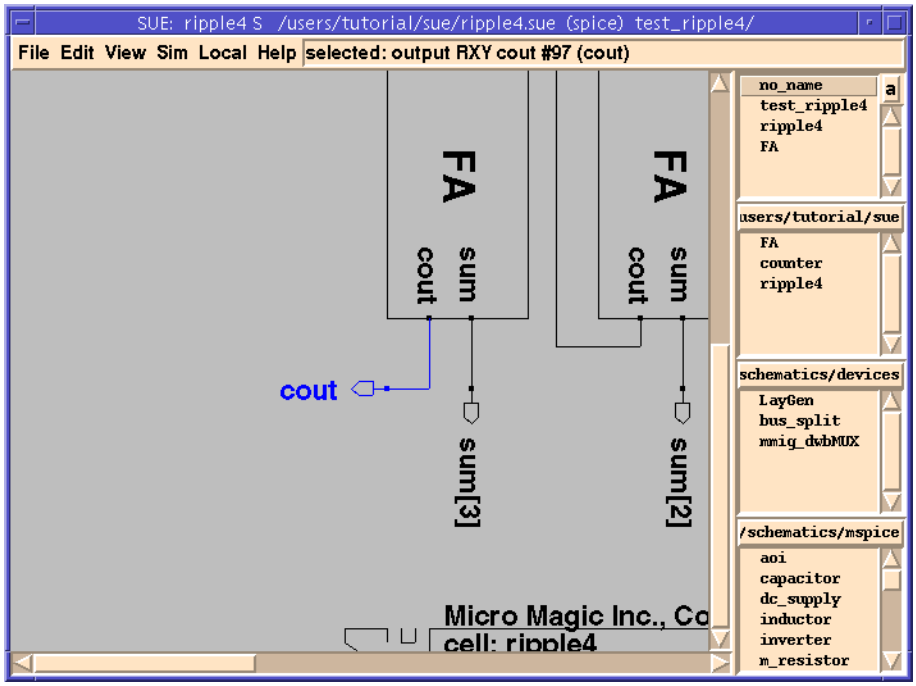
> **Note**
>
> SUE is smart enough to remember where you just were. Let's say you just pushed down five levels and you want to pop back up four levels. To accomplish this just repeat Ctrl-e four times. To push back down type e four times. This is because selected objects stay selected even when you switch cells and return.

Step 4. Select the net which connects to the "**cout**" pin on the "**ripple4**" icon.

Step 5. Type hotkey **e** to push into the icon.

Step 6. You should see something like Figure 26. SUE pushed down into the instance the net is connected to, highlights that net in the **ripple4** schematic and zooms in on it.

Figure 26: Push Into Selected Net

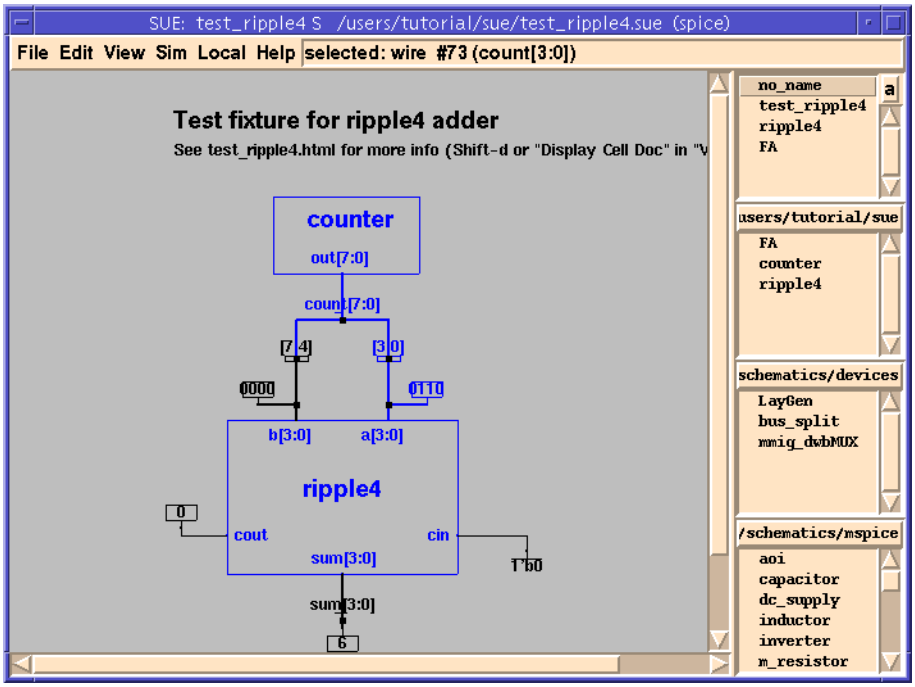


Step 7. Type **e** again, and SUE now pushes you into the "**FA**" schematic and highlights the **cout** net.

Step 8. Type Ctrl-**e** to go back to the **ripple4** schematic, and then type **v** to view the entire schematic.

Step 9. Select the **a[3:0]** net, then type Alt-**e** (or select **Pop Connected** from the **View** menu). You should see something like Figure 27. SUE popped up one level of hierarchy, and highlighted the net which connects to **a[3:0]** and the instances connected to this net.

Figure 27: Pop Up In Hierarchy Of A [3:0] Net



Displaying Design Hierarchy, and Controlling The Simulation

One of the more powerful features in SUE is that it can change what you simulate on the fly. The **Display Design Hierarchy** feature allows you to not only display the hierarchy, but to control what level of the design gets simulated.

During netlisting, SUE uses a very simple algorithm to decide what to do with subcells. If the subcell icon has a Verilog property and you are in Verilog simulation mode, then it uses that property to netlist the cell, ignoring any schematics or hierarchy below it. Otherwise it netlists its schematic.

Let's look at the **ripple4** icon to see if there is a Verilog property.

Step 1. Select the "**ripple4**" icon and push into it with the hotkey **e**. You are now looking at the **ripple4** schematic. We need to look at its icon.

Step 2. Hit c or **Swap Views** in the **View** menu. Now you should be viewing the **ripple4** icon.

You should see a line that looks like:

```
-type auto -name verilog -text ...
```

This is the Verilog property. It tells SUE exactly how to netlist this icon in Verilog. If this seems complicated, realize that it is simply a function of the name of the icon and the port names and can be generated by the **Create Verilog Property** command.

Step 3. Hit **Create Verilog Property** in the **Sim** menu and move the text it creates below the existing Verilog property (you are in a duplication command now). You should see that these are the same.

Step 4. Once you have verified that they are the same, remove the second copy.

If you wanted to simulate the **ripple4** schematic, you would then need to remove this property. Instead of removing the property, we can tell SUE to ignore it with the **Display Design Hierarchy** command.

Using Display Design Hierarchy To Netlist

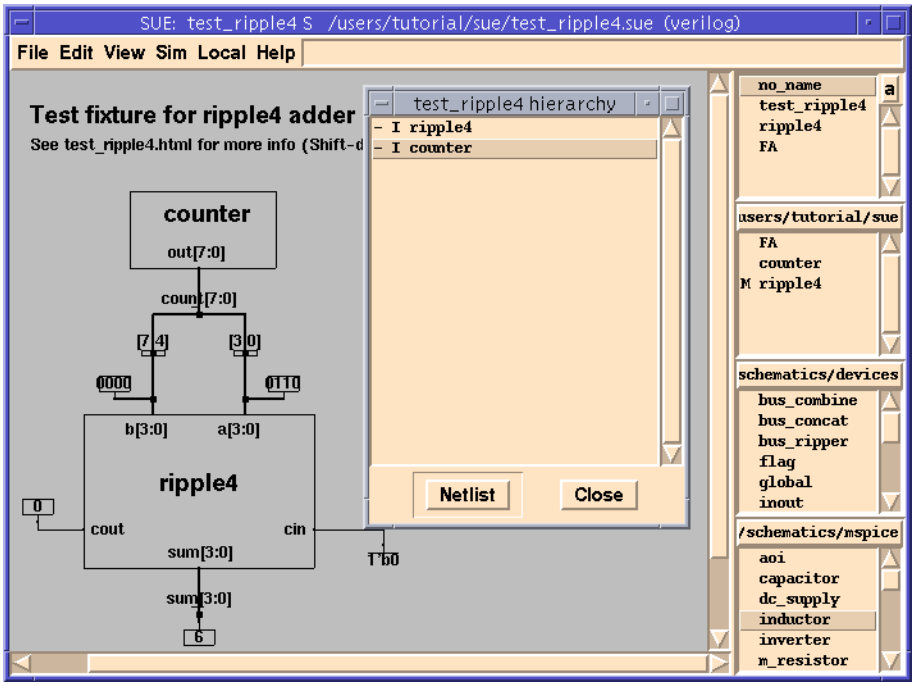
Now we are going to use **Display Design Hierarchy** to control the netlisting.

Step 1. Return to "**test_ripple4**" by hitting Ctrl-e or **Pop Out Of** in the **View** menu.

Step 2. If you aren't already in Verilog mode, change simulation mode to Verilog (**Change Simulation Mode** in the **Sim** menu, click on **Verilog** and hit **Done** or Return).

Step 3. Now select **Display Design Hierarchy** in the **Sim** menu. It will netlist your cell and pop up the **Design Hierarchy** menu on top of your schematic, as shown in Figure 28.

Figure 28: Design Hierarchy Dialog Box



The **Design Hierarchy** popup tells you that the two icons "**ripple4**" and "**counter**" were netlisted with their Verilog properties. Let's tell SUE to netlist the **ripple4** schematic and ignore its Verilog property:

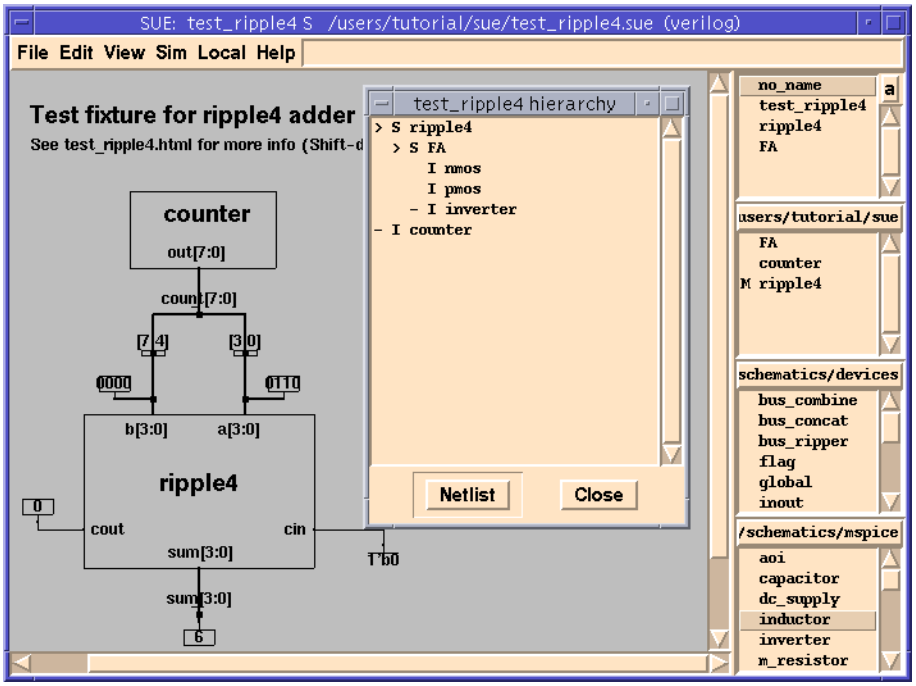
Step **4.** Click with Button-1 on "**- I ripple4**" in the **Design Hierarchy** popup.

The "**I**" in **- I ripple4** should have changed to "**S**", meaning that the Verilog property is now ignored for this icon and the *schematic* will be netlisted.

5. Hit the **Netlist** button in the **Design Hierarchy** popup. SUE re-netlists this schematic and now shows you that there is a "**FA**" icon whose Verilog property is being used.

6. Click with Button-1 on "**- I FA**" in the **Design Hierarchy** popup and then click on **Netlist**. You should see the devices contained in the **FA** cell (see Figure 29). Likewise the `test_ripple4.vh` file will include all of this hierarchy.

Figure 29: Expanded Design Hierarchy



Mixed-Mode Verilog Simulation

When we simulate our circuit in Verilog, SUE uses the following Verilog command line arguments "`+libext+.vb+.v -y . -y lib.v`". These instruct Verilog to look for any undefined modules in either the current directory (".") or the subdirectory ("`lib.v`") and look in files with extensions of "`.vb`" or "`.v`". (Note: you can add to or change these options in the `.suerc` file — see the Micro Magic Tools SUE User Guide for more information.)

Therefore, we need to make sure that we have defined Verilog modules for any icons whose Verilog properties we use (unless they are included inline in the icon view, like the **inverter/my_inverter** examples above). Let's make sure we have the "`.vb`" files:

Step 1. You should be in **test_ripple4**. If not, pop up to it with Ctrl-e.

Step 2. Select the **ripple4** icon and hit Shift-e or **Edit Verilog** in the **Sim** menu to view the Verilog. This will bring up the Verilog in an editor.

You should see that we have defined the adder behavioral model with the line:

```
assign {cout, sum[3:0]} = {1'b0, a[3:0]} + {1'b0, b[3:0]} + {4'b0, cin};
```

So if we netlist **test_ripple4** with the **ripple4** Verilog property, our simulation will be very fast since it only executes this one line instead of the underlying hierarchy. Let's do it.

Step 3. Quit out of the editor.

Step 4. In the **Design Hierarchy** popup, make sure the you see "**- I ripple4**". If **ripple4** shows an "**S**" for *schematic*, click on it to return to "**I**". Then hit **Netlist**.

Step 5. Look at **`test_ripple4.vh`** in a text editor. Do you see how short that file is? Look at the **`test_ripple4`** module.

Step 6. Hit Ctrl-i or **Init Probe** in the **Sim** menu to start up Verilog. Do you see the Verilog command line arguments "**`+libext+.vb+.v -y . -y lib.v`**" in the window you started SUE from?

Step 7. Hit s a few times to step the time forward. You should see the flags changing.

Step 8. Select the **ripple4** icon and push into it (hotkey: e). Select one of the **FA** icons and push into it (hotkey: e). Hit Ctrl-o or **Display Term Values** in the **Sim** menu. You won't get any values. Why? because this schematic is not part of the simulation. The netlist doesn't include all of this hierarchy.

Now let's simulate with all of the hierarchy:

Step 9. Click with Button-1 on **-I ripple4** in the **Design Hierarchy** popup and then hit **Netlist**. (This will also bring you to your top cell: **test_ripple4**.)

Step 10. If you see "**-I FA**" then click on that line to get the **FA** schematic and hit **Netlist**. Now we will simulate three levels of hierarchy.

Step 11. Look at **`test_ripple4.vh`** in a text editor. Do you see that there are now modules for **FA** and **ripple4**?

Step 12. Hit Ctrl-i or **Init Probe** in the **Sim** menu to start up Verilog.

Step 13. Hit s a few times to step the time forward. You should see the flags changing.

Step 14. Select the **ripple4** icon and push into it (hotkey: e). Select one of the **FA** icons and push into it (hotkey: e). Hit Ctrl-o or **Display Term Values** in the **Sim** menu. You should now get values since the **FA** schematic IS part of the netlist.

By changing the **Display Design Hierarchy** in SUE, the designer or verification person can change what is being simulated on the fly. Thus, schematics can be "flattened" to verify that they are correct down to the "transistor" level, or simulations can be done at very high-level behavioral levels to increase speed.

In addition, mixed-level simulations can also be done. That is, simulate high-level behavioral models for some blocks, and flatten others down to the transistor level.

Most importantly, what you see in the **Display Design Hierarchy** window is what you get when you simulate. You never have to wonder what you are actually simulating!

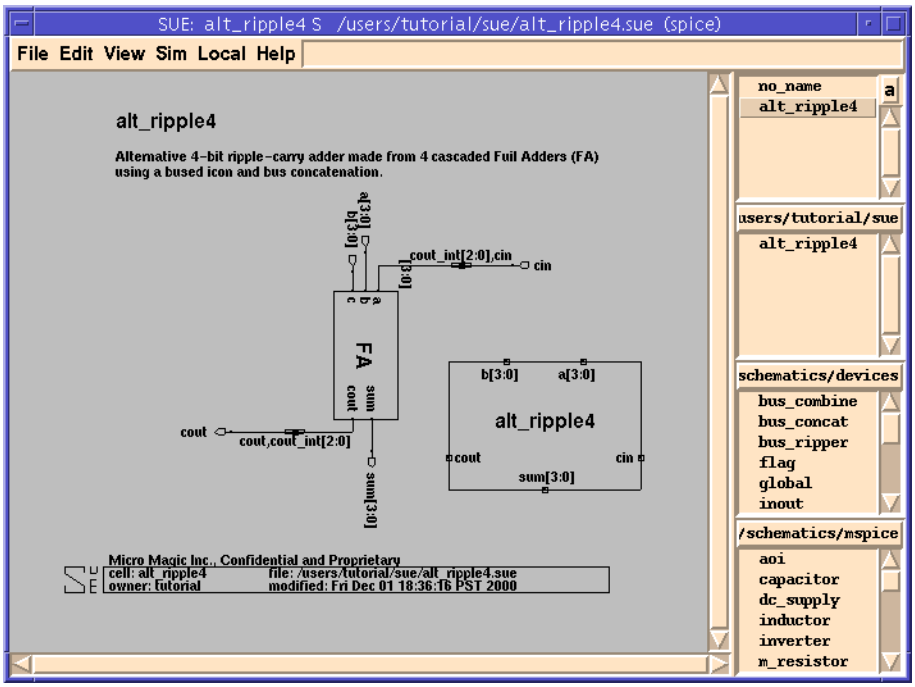
When you are done looking over the adder example go to the next section. You can also run SPICE or IRSIM on this example.

Buses

While the "ripple4" example circuit uses simple bused wires or buses, it doesn't demonstrate bused instances or bus concatenation. An alternative "ripple4" schematic was created to demonstrate these features.

Step **1.** Open up the schematic "**alt_ripple4**". It should look like Figure 30.

Figure 30: "`alt_ripple4`": Alternative Ripple-Carry Adder Schematic



Step **2.** Notice that the 4 **FA** icons have been replaced by one icon whose name is "**[3:0]**". Simply naming an icon with a Verilog bus range will cause it to be duplicated that many times.

Step **3.** Look at the "**a**" input to the **FA** icon. We want this to be connected to the "**cout**" output of the previous **FA** or "**cin**" if it is the first one. This is done by labelling the net with a **name_net_s** icon with the name "**cout_int[2:0],cin**". The comma (",") causes a Verilog bus concatenation. "**cout_int[2:0]**" are the three nets that go from "**cout**" on one **FA** to "**cin**" on the next one. For more examples of buses, see the Micro Magic Tools SUE User Guide.

Step **4.** You can replace "**ripple4**" in **test_ripple4** with "**alt_ripple4**" and simulate it to see that it works the same way as "**ripple4**".

Part 5
Cross-probing With MAX

Cross-Probing With MAX

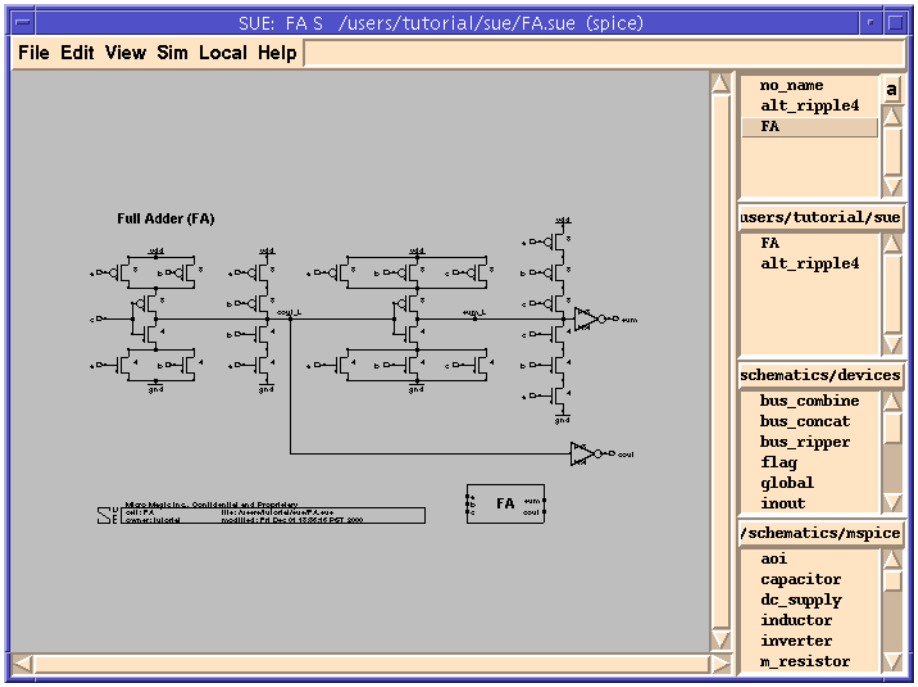
In this section we are going to bring up a schematic in SUE and a layout in MAX, the MMT full-custom layout editor. Using interprocess communication, SUE and MAX will then share data. In this case, you will be able to select any net in a schematic in SUE and tell MAX to highlight that net in the layout, and vice-versa.

Note

You will need a license for MAX to run this section. If you don't know if you have a MAX license, just type "**max**" at the UNIX prompt. If MAX comes up, you have a license.

Step **1.** If you don't have SUE running at this time, start it, and load the **FA** cell. If you are continuing from before, click on the **FA** cell in the **Schematic List** box. Your screen should look like Figure 31.

Figure 31: "FA" Schematic

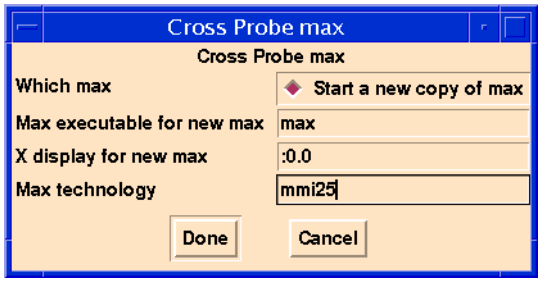


Since cross-probing to layout is a transistor-level operation, you need to be in either **spice** or **sim** simulation mode.

Step 2. If you are not sure that you are in **spice** or **sim** simulation mode — look at the **MAX Title Bar**. If you are unsure, go to **Change Simulation Mode...** in the **Sim** menu and change to **spice** or **sim**.

Step 3. Select **Max Cross Probe Init** in the **Sim** menu. the **Cross Probe** popup will appear, as shown in Figure. Be sure that "**Start a new copy of MAX**" is selected. Type in **`mmi25`** for the MAX technology. Then click on **Done**. This will set up the cross-probing as described below.

Figure 32: MAX Cross-Probe Popup



- A copy of MAX will start, and this will load the corresponding MAX layout file (**FA.max**, for this example).

- Next, MAX and SUE both create sim netlists of their current layouts/schematics.

  SUE doesn't have to be in **sim** mode for this.

- Finally, LVS using *Gemini* (a netlist compare program from the University of Washington) is run to compare the schematic and layout netlists and determine node equivalents. The nets which have correspondences are selected in both SUE and MAX.

Step 4. To cross-probe a specific wire in SUE, select one wire as shown in Figure 33 and type the hotkey k or **Max Cross Probe** in the **Sim** menu. The wire that you selected will be highlighted on the layout in the MAX window as shown in Figure 34.

Figure 33: Net Selected In SUE

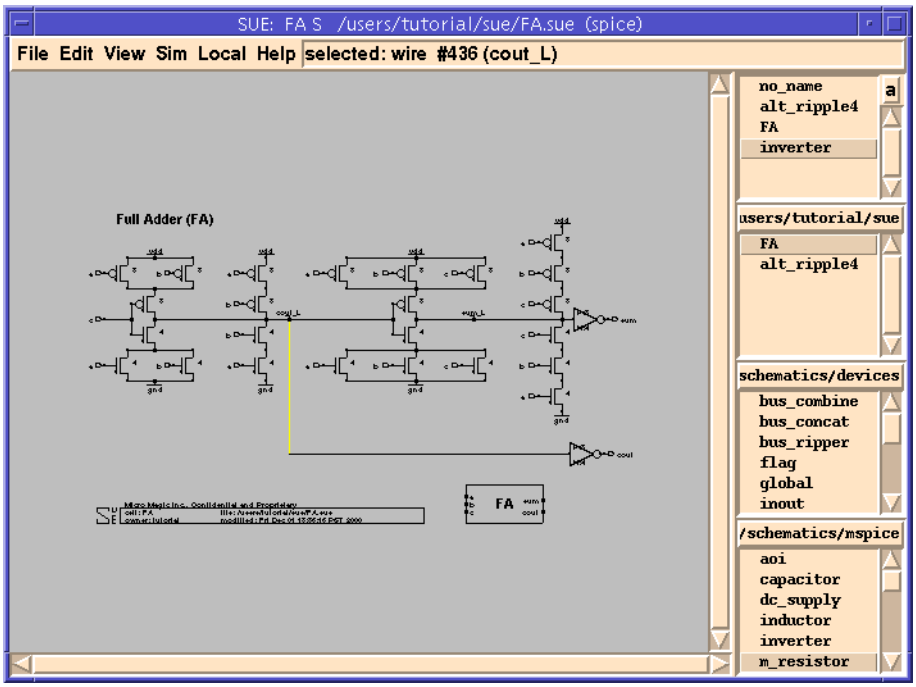
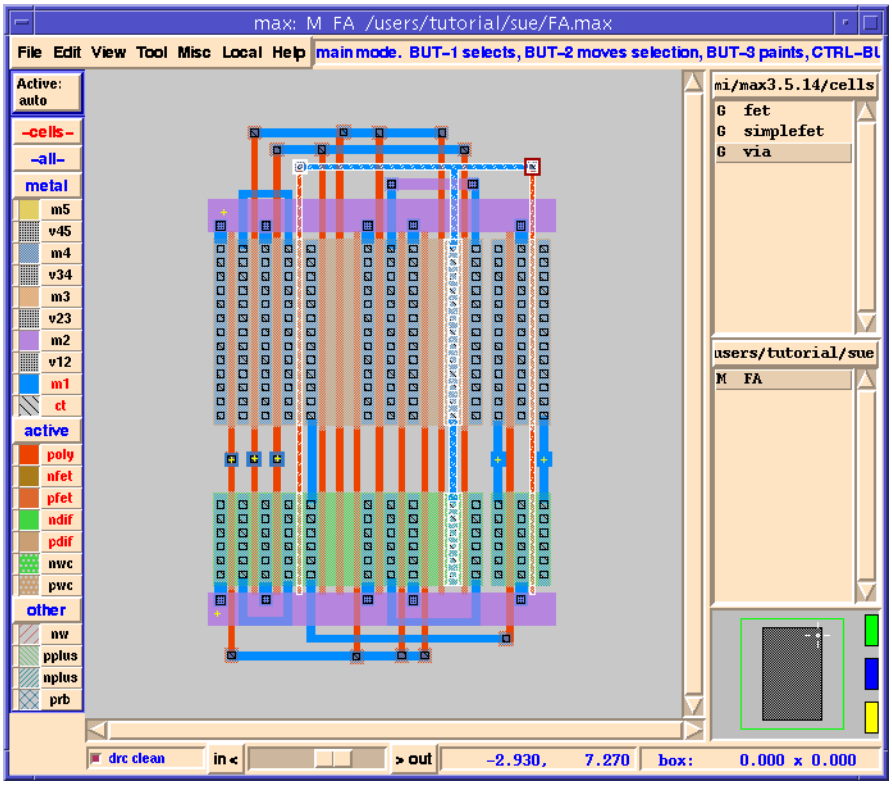


Figure 34: Highlighted Net In MAX



Step 5. Now select a different net in MAX by moving the cursor over the net and hitting s. You can also select a net in the current hierarchy by simply clicking on it with the left mouse button (Button-1).

Step 6. Once you have selected a net in MAX, hit k, or select **SUE Cross Probe Net** from the **Tool** menu. The corresponding net should be selected in SUE.

In addition to cross-probing, MAX is a very fast and powerful, full-featured, full-custom layout editor with a complete language API. Refer to the Micro Magic Tools MAX User Guide and Micro Magic Tools MAX Tutorial (accessed from either the **Help** menu in MAX, or from the Micro Magic Tools Documentation Guide) for more details.

Now take your left hand and raise it over your head and pat yourself on the back. You survived the SUE tutorial, and hopefully learned a little about SUE. However, the best way to really learn SUE - or any tool - is to try it out with something you designed yourself. So go ahead, and good luck!