Magic Tutorial #11: Using IRSIM and RSIM with Magic

Michael Chow
Mark Horowitz

Computer Systems Laboratory
Center for Integrated Systems
Stanford University
Stanford, CA 94305

This tutorial corresponds to Magic version 6.

Tutorials to read first:

Magic Tutorial #1: Getting Started
Magic Tutorial #2: Basic Painting and Selection
Magic Tutorial #4: Cell Hierarchies
Magic Tutorial #8: Circuit Extraction

Commands introduced in this tutorial:

:getnode, :rsim, :simcmd, :startrsim

Macros introduced in this tutorial:

none

1. Introduction

This tutorial explains how to use Magic's interface to the switch-level circuit simulators, RSIM and IRSIM. The interface is the same for both these simulators and, except where noted, RSIM refers to IRSIM as well. This interface eliminates the tedium of mapping node names to objects in the layout and typing node names as RSIM input. It allows the user to select nodes using the mouse and apply RSIM commands to them or to display the node values determined by RSIM in the layout itself. You should already be familiar with using both RSIM and Magic's circuit extractor. Section 2 describes how to prepare the files necessary to simulate a circuit. Section 3 describes how to run RSIM interactively under Magic. Section 4 explains how to determine the node names that

RSIM uses. Lastly, section 5 explains how to use the RSIM tool in Magic to simulate a circuit.

2. Preparations for Simulation

Magic uses the RSIM input file when it simulates the circuit. Before proceeding any further, make sure you have the correct versions of the programs **ext2sim** and **rsim** installed on your system. Important changes have been made to these programs to support simulation within Magic. To try out this tool on an example, copy all the **tut11***x* cells to your current directory with the following command:

cp ˜cad/lib/magic/tutorial/tut11* .

The **tut11a** cell is a simple 4-bit counter using the Magic scmos technology file. Start Magic on the cell **tut11a**, and extract the entire cell using the command:

:extract all

When this command completes, several **.ext** files will be created in your current directory by the extractor. The next step is to flatten the hierarchy into a single representation. Return to the Unix c-shell by quitting Magic.

The program **ext2sim** is used to flatten the hierarchy. Run this program from the c-shell by typing:

ext2sim -L -R -c 20 tut11a

This program will create the file **tut11a.sim** in your current directory.

If you are running IRSIM, the **tut11a.sim** can be used directly as input to the simulator and you should skip the next step. Instead, if you will be using RSIM, the last step is to create the binary representation of the flattened hierarchy by using the program **presim**. To do this, type:

presim tut11a.sim tut11a.rsm ˜cad/lib/scmos150.prm -nostack -nodrops

The third file is the parameter file used by presim for this circuit. The convention at Stanford is to use the suffix *.rsm* when naming the RSIM input file. The file **tut11a.rsm** can also be used as input for running RSIM alone.

3. Using RSIM

Re-run Magic again to edit the cell **tut11a**. We'll first learn how to run RSIM in interactive mode under Magic. To simulate the circuit of tut11a, using IRSIM type the command:

:rsim scmos150.prm tut11a.sim

To simulate the circuit of tut11a, using RSIM type the command:

:rsim tut11a.rsm

You should see the RSIM header displayed, followed by the standard RSIM prompt (**rsim>** or **irsim>**, depending on the simulator) in place of the usual Magic prompt; this means keyboard input is now directed to RSIM. This mode is very similar to running RSIM alone; one difference is that the user can escape RSIM and then return to Magic. Also, the mouse has no effect when RSIM is run interactively under Magic.

Only one instance of RSIM may be running at any time under Magic. The simulation running need not correspond to the Magic layout; however, as we shall see later, they must correspond for the RSIM tool to work. All commands typed to the RSIM prompt should be RSIM commands. We'll first run RSIM, then escape to Magic, and then return back to RSIM. Type the RSIM command

@ tut11a.cmd

to initialize the simulation. (Note there is a " " after the @.) Now type **c** to clock the circuit. You should see some information about some nodes displayed, followed by the time. Set two of the nodes to a logic "1" by typing **h RESET_B hold**. Step the clock again by typing **c**, and RSIM should show that these two nodes now have the value "1."

You can return to Magic without quitting RSIM and then later return to RSIM in the same state in which it was left. Escape to Magic by typing:

.

(a single period) to the RSIM prompt. Next, type a few Magic commands to show you're really back in Magic (signified by the Magic prompt).

You can return to RSIM by typing the Magic command **rsim** without any arguments. Type:

:rsim

The RSIM prompt will be displayed again, and you are now back in RSIM in the state you left it in. Experiment with RSIM by typing some commands. To quit RSIM and return to Magic, type:

q

in response to the RSIM prompt. You'll know you're back in Magic when the Magic prompt is redisplayed. If you should interrupt RSIM (typing a control-C), you'll probably kill it and then have to restart it. RSIM running standalone will also be killed if you interrupt it. If you interrupt IRSIM (typing a control-C), the simulator will abort whatever it's doing (a long simulation run, for example) and return to the command interpreter by prompting again with **irsim>**.

4. Node Names

It's easy to determine node names under Magic. First, locate the red square region in the middle right side of the circuit. Move the cursor over this region and select it by typing **s**. To find out the name for this node, type:

:getnode

Magic should print that the node name is *RESET_B*. The command **getnode** prints the names of all nodes in the current selection. Move the cursor over the square blue region in the upper right corner and add this node to the current selection by typing **S**. Type **:getnode** again, and Magic should print the names of two nodes; the blue node is named *hold*. You can also print aliases for the selected nodes. Turn on name-aliasing by typing:

:getnode alias on

Select the red node again, and type **:getnode**. Several names will be printed; the last name printed is the one RSIM uses, so you should use this name for RSIM. Note that **get-node** is not guaranteed to print all aliases for a node. Only those alises generated when the RSIM node name is computed are printed. However, most of the alaiases will usually be printed. Printing aliases is also useful to monitor the name search, since **getnode** can take several seconds on large nodes. Turn off aliasing by typing:

:getnode alias off

getnode works by extracting a single node. Consequently, it can take a long time to compute the name for large nodes, such as *Vdd* or *GND*. Select the horizontal blue strip on top of the circuit and run **:getnode** on this. You'll find that this will take about six seconds for **getnode** to figure out that this is *Vdd*. You can interrupt **getnode** by typing **ˆC** (control-C), and **getnode** will return the "best" name found so far. There is no way to tell if this is an alias or the name RSIM expects unless **getnode** is allowed to complete. To prevent these long name searches, you can tell **getnode** to quit its search when certain names are encountered. Type:

:getnode abort Vdd

Select the blue strip on top of the circuit and type **:getnode**. You'll notice that the name was found very quickly this time, and **getnode** tells you it aborted the search of *Vdd*. The name returned may be an alias instead of the the one RSIM expects. In this example, the abort option to **getnode** will abort the name search on any name found where the last component of the node name is *Vdd*. That is, **getnode** will stop if a name such as "miasma/crock/*Vdd*" or "hooha/*Vdd*" is found.

You can abort the search on more than one name; now type **:getnode abort GND**. Select the bottom horizontal blue strip in the layout, and type **:getnode**. The search will end almost immediately, since this node is *GND*. **getnode** will now abort any node name search when either *Vdd* or *GND* is found. The search can be aborted on any name; just supply the name as an argument to **getnode abort**. Remember that only the last part of the name counts when aborting the name search. To cancel all name aborts and resume normal name searches, type:

:getnode abort

getnode will no longer abort the search on any names, and it will churn away unless interrupted by the user.

5. RSIM Tool

You can also use the mouse to help you run RSIM under Magic. Instead of typing node names, you can just select nodes with the mouse, tell RSIM what to do with these nodes, and let Magic do the rest. Change tools by typing:

:tool rsim

or hit the space bar until the cursor changes to a pointing hand. The RSIM tool is active when the cursor is this hand. The left and right mouse buttons have the same have the same function as the box tool. You use these buttons along with the select command to select the nodes. The middle button is different from the box tool. Clicking the middle button will cause all nodes in the selection to have their logical values displayed in the layout and printed in the text window. We need to have RSIM running in order to use this tool. Start RSIM by typing:

:startrsim tut11a.rsm

The **.rsm** file you simulate must correspond to the root cell of the layout. If not, Magic will generate node names that RSIM will not understand and things won't work properly. If any paint is changed in the circuit, the circuit must be re-extracted and a new **.rsm** file must be created to reflect the changes in the circuit.

Magic will print the RSIM header, but you return to Magic instead of remaining in RSIM. This is an alternate way of starting up RSIM, and it is equivalent to the command **rsim tut11a.rsm** and typing a period (**.**) to the RSIM prompt, escaping to Magic. We need to initialize RSIM, so get to RSIM by typing **:rsim** and you'll see the RSIM prompt again. As before, type **@ tut11a.cmd** to the RSIM prompt to initialize everything. Type a period (**.**) to return to Magic. We are now ready to use the RSIM tool.

As mentioned earlier, **tut11a** is a 4-bit counter. We'll reset the counter and then step it using the RSIM tool. Locate the square blue area on the top right corner of the circuit. Place the cursor over this region and select it. Now click the middle button, and the RSIM value for this node will be printed in both the text window and in the layout. Magic/RSIM will report that the node is named *hold* and that its current value is *X*. You may not be able to see the node value in the layout if you are zoomed out too far. Zoom in closer about this node if necessary. Try selecting other nodes, singly or in groups and click the middle button to display their values. This is an easy way to probe nodes when debugging a circuit.

Select *hold* again (the blue square). This node must be a "1" before resetting the circuit. Make sure this is the only node in the current selection. Type:

:simcmd h

to set it to a "1." Step the clock by typing:

:simcmd c

Click the middle button and you will see that the node has been set to a "1." The Magic command **simcmd** will take the selected nodes and use them as RSIM input. These uses of **simcmd** are like typing the RSIM commands *h hold* followed by *c*. The arguments given to **simcmd** are normal RSIM commands, and **simcmd** will apply the specified RSIM command to each node in the current selection. Try RSIM commands on this node (such as *?* or *d*) by using the command as an argument to **simcmd**.

You can enter RSIM interactively at any time by simply typing **:rsim**. To continue using the RSIM tool, escape to Magic by typing a period (**.**) to the RSIM prompt.

The node *RESET_B* must be set to a "0." This node is the red square area at the middle right of the circuit. Place the cursor over this node and select it. Type the Magic commands **:simcmd l** followed by **:simcmd c** to set the selected node to a "0." Click the middle mouse button to check that this node is now "0." Step the clock once more to ensure the counter is reset. Do this using the **:simcmd c** command.

The outputs of this counter are the four vertical purple strips at the bottom of the circuit. Zoom in if necessary, select each of these nodes, and click the middle button to check that all are "0." Each of these four nodes is labeled *bit_x*. If they are all not "0", check the circuit to make sure *hold=1* and *RESET_B=0*. Assuming these nodes are at their correct value, you can now simulate the counter. Set *RESET_B* to a "1" by selecting it (the red square) and then typing **:simcmd h**. Step the clock by typing **:simcmd c**. Using the same procedure, set the node *hold* (the blue square) to a "0."

We'll watch the output bits of this counter as it runs. Place the box around all four outputs (purple strips at the bottom) and zoom in so their labels are visible. Select one of the outputs by placing the cursor over it and typing **s**. Add the other three outputs to the selection by placing the cursor over each and typing **S**. These four nodes should be the only ones in the selection. Click the middle mouse button to display the node values. Step the clock by typing **:simcmd c**. Click the middle button again to check the nodes. Repeat stepping the clock and displaying the outputs several times, and you'll see the outputs sequence as a counter. If you also follow the text on the screen, you'll also see that the outputs are also being watched.

You may have noticed that the results are printed very quickly if the middle button is clicked a second time without changing the selection. This is because the node names do not have to be recomputed if the selection remains unchanged. Thus, you can increase the performance of this tool by minimizing selection changes. This can be accomplished by adding other nodes to the current selection that you are intending to check.

To erase all the RSIM value labels from the layout, clear the selection by typing:

:select clear

and then click the middle mouse button. The RSIM labels do not affect the cell modified flag, nor will they be written in the **.mag** file. When you're finished using RSIM, resume RSIM by typing **:rsim** and then quit it by typing a **q** to the RSIM prompt. Quitting Magic before quitting RSIM will also quit RSIM.

We've used a few macros to lessen the typing necessary for the RSIM tool. The ones commonly used are:

:macro h "simcmd h"
:macro l "simcmd l"
:macro k "simcmd c"