ARMY RESEARCH LABORATORY

Full Custom Integrated Circuit (IC) Design Flow at U.S. Army Research Laboratory

by James Wilson

ARL-TN-0422 **February 2011**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TN-0422 **February 2011**

Full Custom Integrated Circuit (IC) Design Flow at U.S. Army Research Laboratory

James Wilson
Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
February 2011		

4. TITLE AND SUBTITLE
Full Custom Integrated Circuit (IC) Design Flow at U.S. Army Research Laboratory

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
James Wilson

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: RDRL-SER-E
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TN-0422

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR"S ACRONYM(S)

11. SPONSOR/MONITOR"S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The steps required to set up and run the Cadence Full Custom Integrated Circuit Design Flow are given in detail. Included is a walkthrough showing the design of an inverter, from schematic capture through layout, including design rule check (DRC), layout versus schematic (LVS) and parasitic extraction.

15. SUBJECT TERMS
Design flow, cadence, tutorial

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	28	James Wilson
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED			19b. TELEPHONE NUMBER (*Include area code*) (301) 394-0328

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Introduction

The U.S. Army Research Laboratory (ARL) has identified the need to reengage in the field of analog and radio frequency (RF) custom integrated circuit (IC) design. The major hurdle to this effort is the procurement, installation, and set-up of the software needed to design the ICs, as well as the process design kit (PDK) for each foundry that a chip is to be designed in. These efforts have been successfully completed, and this report details the flow that has been implemented.

Flow Overview

The steps required designing a custom analog or RF IC are shown in figure 1.

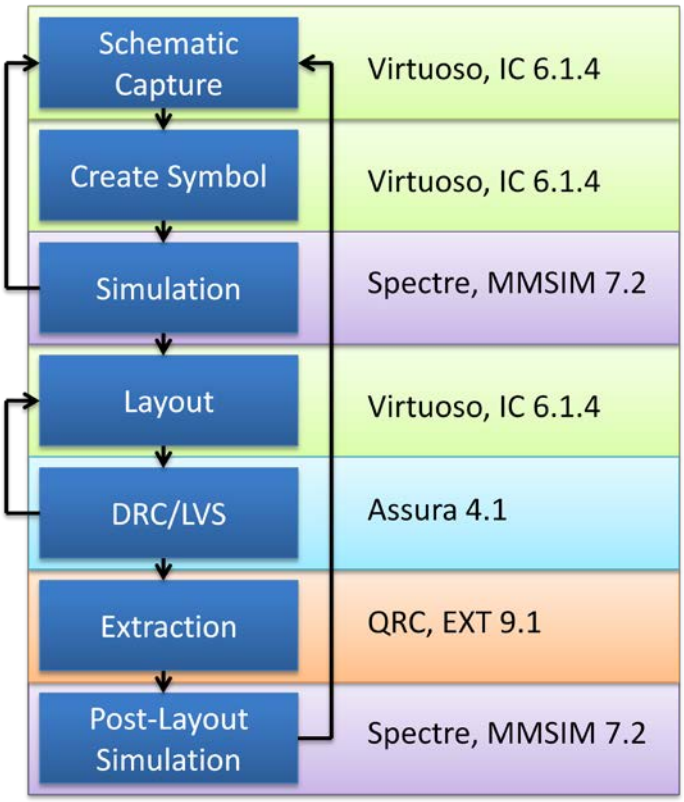

Figure 1. Full custom design flow.

The design stage is the column on the left, while the column on the right is the tool available at ARL through the Cadence Design Systems, Inc. licensed software. The first step is schematic capture, performed in the Virtuoso design software. After the schematic is entered, a symbol is

created to match the schematic. Then this symbol is instantiated into a testbench schematic. This testbench is simulated using the Spectre simulator, available from the MMSIM software tree. The schematic is adjusted and fine-tuned until the simulations show proper function. At this stage, the schematic is used as the basis to create a layout. The layout is done in Virtuoso, as well, though it is performed in a different part of the software. After the layout is completed, design rule check (DRC) and layout versus schematic (LVS) are performed. These steps ensure that no design rules have been violated in the layout, and that the layout matches the schematics in terms of electrical connections and design parameters. After DRC and LVS, an extracted netlist, including parasitic elements, is created. This new netlist is used in the original testbench to verify that the completed layout still matches the required simulation parameters. If it does not, the schematic is altered and the whole process from capture through extraction is iterated again until the extracted netlist simulates correctly.

The specific tool versions currently installed at ARL are summarized in table 1.

Table 1. Tool names and versions.

Tool Name	Installation Package	Version
Virtuoso	IC614506	6.1.4.500.1
Spectre	MMSIM720284	7.2.0.284.isr9
Assura	ASSURA41USRHF10A_613	av4.1:dfII6.1.3.500.15
QRC	EXT910BASE	9.1.0-p006

Initial Setup

The Cadence tools are only available for the Linux operating system. They are currently installed on the machine, eagle.arl.army.mil, in the directory /home/jwilson/cadence_install. In order to run the tools, a special directory structure needs to be set up for each different PDK that is to be used. These are usually organized into a single directory, typically called cadence. As an example, this report will go through a simple design of an inverter in the IBM 8HP BiCMOS process.

To start, create a directory in the home directory called cadence. Copy the file /home/jwilson/cadence/setup_614.sh into the new cadence directory. Inside this directory, create another directory named ibm8hp. Copy the following files into this ibm8hp directory:

- /home/jwilson/cadence_install/pdks/ibm/IBM_PDK/bicmos8hp/relHP/cdslib/bicmos8hp/display.drf
- /home/jwilson/cadence_install/pdks/ibm/IBM_PDK/bicmos8hp/relHP/cdslib/examples/.cdsinit
- /home/jwilson/cadence_install/pdks/ibm/IBM_PDK/bicmos8hp/relHP/cdslib/examples/cds.lib

The file display.drf controls the colors, line widths, stipple patterns, and other graphical data that are used to display the graphics. The file .cdsinit is loaded at program start-up and sets up the appropriate initialization routines needed by the design kit. The file cds.lib contains the pointers to the default libraries used in most designs. These files will be read only by default, so to modify them, you will need to add write permissions for yourself.

While in the ibm8hp directory, source the setup_614.sh file using the following command:

source ../setup_614.sh

This will set up the proper environment variables to allow the programs to run. Now launch the Virtuoso tool with command “virtuoso &”. You should see the command interpreter window (CIW) shown in figure 2.

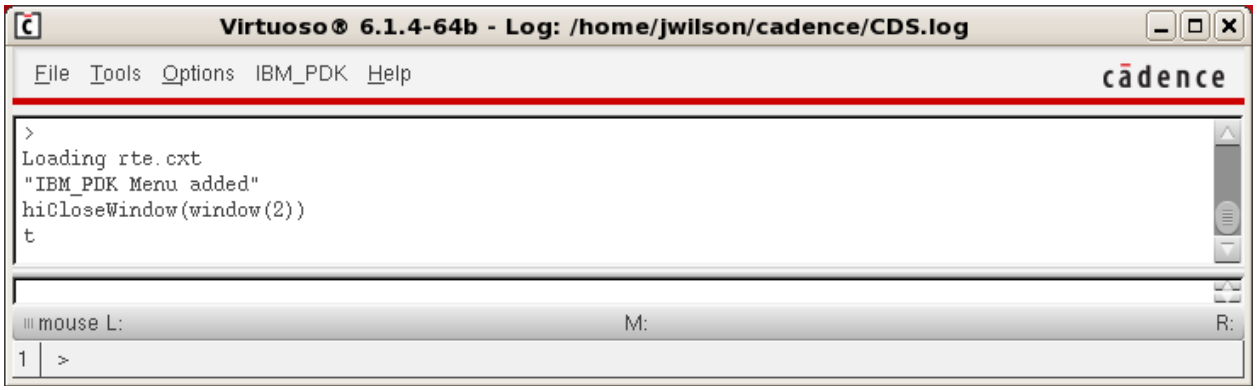

Figure 2. CIW window.

You can enter commands by hand in the bottom of the CIW, and all actions you perform can be seen in the log here. To get started, click Tools->Library Manager. This will open the library manager window, which is shown in figure 3. This window is where you can view all of the individual design elements in your IC. It also allows you to browse through the available libraries. To make organization easier, click on the “Show categories” checkbox in the top left corner of the window. Then click on the library bicmos8hp. This is the library containing all of the elements that IBM provides for designing in the 8HP process. Under the category “fet” are all the MOSFETs, under the category “ind” are the inductors, etc.

Before a new design can be created, a library needs to be created into which the design will be placed. This is done through custom commands that IBM provides. In the CIW, click on IBM_PDK->Library->Create. This opens the new library window. In this window, enter a name for the new library, such as inverter_tutorial. Under Technology File, select “Attach to an existing technology library” then click ok. When the new window pops up, select bicmos8hp for the technology library, and click ok. A final window will pop up, in which you can change the number of metals to be used. Choose M7 and click ok one more time.

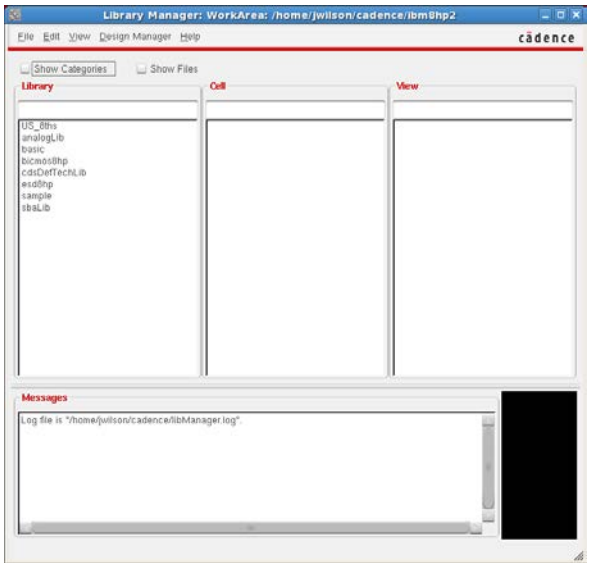

Figure 3. Library manager window.

Back in the library manager window, you should now see a new library named inverter_tutorial in the list. This is the library where you will create the schematics and layout to finish this tutorial.

Schematic Capture

To create the inverter, first make sure that inverter_tutorial is highlighted in the library column. If it is not, go ahead and click on it. Then click on the file "menu->New->Cell View…" In the New File window that pops up, enter "inverter" for the cell, and verify that schematic is the view. Click on "ok". If you get a window asking about using the XL license, click "ok". You should now have a blank schematic window open. To add the NMOS fet to the design, click on "create->instance" or press "i". This will bring up the "Add Instance" window. Click the browse button, and a new library browser window will open. In this window, click on "Show Categories," then click on the Library "bicmos8hp" Category "fet," Cell "nfet," View "symbol". You will notice that when you click on nfet, the "Add Instance" window populates with all of the fields that you can edit for the fet. Just ignore this for now and move the mouse back into the schematic editor window. You should see a yellow stick figure of the fet at your mouse. Move the device to where you want it and click once. Hit "Escape" to exit the "Add Instance" mode. By default, Cadence will keep the same command running that you selected until you hit escape to cancel it.

Now click on the fet to select it, and hit "shift-q". This will bring up the property editor, where you can change the size of the device and any other options that IBM has made available. We will leave the device with the default settings for this design, so click "cancel" to dismiss the

window. Back in the schematic editor, press “i” to add a new instance, click “browse,” and navigate to the pfet. Place this in the schematic above the nfet, and hit “Escape”. These are all the components needed for the inverter; however, we still need to add pins to allow for the inputs and outputs.

To create a pin, click on “p”. This brings up the “Add Pin” window. You can enter a single pin name, or all your pin names at once, in this window. So type “in out vdd gnd” and change the direction to “inputOutput”. When you move the mouse back into the schematic window you will notice a polygon under the mouse cursor. This will be the first pin in the list you entered. Since this the pin “in” click on the left of the schematic to place it. Back in the “Add Pin” window, click the “Sideways” button to mirror the “out” pin so that it is pointing the right direction, then click on the right side of the schematic to place it. Click the “rotate” button then click on the top of the schematic to place the “vdd”, then click “upside down” and click on the bottom of the schematic to place the gnd pin. Hit “Escape” to exit pin mode. Now that all the elements are placed in the design, the only thing left to do is wire them together.

Press the “w” key to enter wire mode. By default, the wire window does not appear. If you want to see the options available, press the F3 key to bring up the window. By default, F3 toggles the mode window on and off. While in wire mode, wire up the inverter so that it looks something like figure 4.

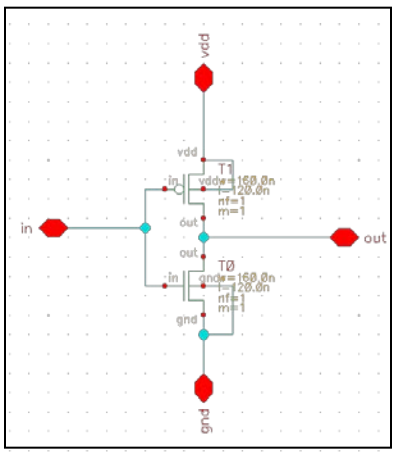

Figure 4. Inverter schematic.

Create Symbol

With the schematic complete, the next step is to create a symbol of the schematic so that it can be placed into other schematics. To do this, from the schematic editor window, click on “Create”->“Cellview”->“From Cellview”. Leave the defaults in the pop-up window and click “ok”. In the “Symbol Generation Options” pop-up window, you can move the pin names around so that

they are in the most logical positions. So move “in” to the “left pins” section, “out” to “right pins”, and “gnd” to “bottom pins”. Then click “ok”.

This will create a new cellview called “symbol” for the design. The symbol editor will open with the new symbol ready to edit. Here you can delete the drawings and add new artwork to make the symbol look like the component it is supposed to represent. However, in this tutorial, we will just leave it as the default symbol. So click on “File”->“Check and Save” to save it, then close the symbol editor window. Back in the schematic editor window, also do a “check and save”, then close that window.

Simulation

To simulate the inverter, we will create a new schematic for the testbench. In the library manager window, click on the inveter_tutorial library, then click “File”->“New”->“Cellview”. Name the cell “tb_inverter” and click “ok”.

In the new schematic window that opens, click “i”, then “browse”, and browse to your inverter_tutorial library. Select the inverter from the cell column, and place it in the schematic. To add the input signal, click “i”, “browse” and click on the “analogLib” library. In the “category” column, expand the sources and click on “independent”. Then select the vpulse source and add it to the schematic. Next, click on the vdc source and add that to the schematic. Then add a “gnd” from the “Globals” subcategory of the sources. Finally, add a capacitor for the load from the “Passives” category. Wire the vpulse to the input, vdc to vdd, and the cap to the output. Tie all the grounds to the gnd. Click check and save and correct any errors. Your schematic should look similar to figure 5.

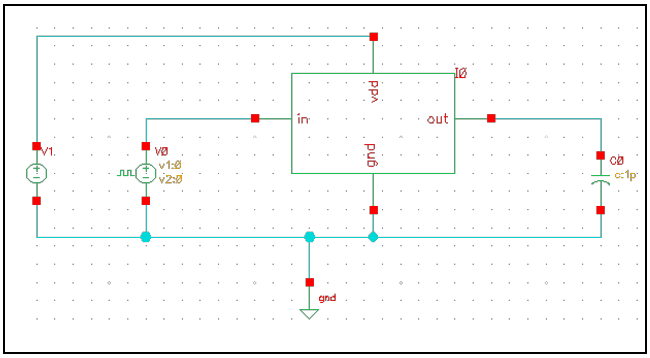

Figure 5. Testbench schematic of the inverter.

Click on the vdc source, and hit “shift-q”. For the DC voltage, enter “vdd”, and then click ok. Hit “ctrl-d” to deselect everything, then click on the vpulse source and hit “shift-q”. Here, enter “vdd” for “Voltage 2”, “1n” for “rise time”, “1n” for “fall time”, “1/fclk” for “period”, and

"1/(2*fclk)" for "pulse width". Then click "ok". Now edit the capacitor properties to change it to "50f". Do a check and save.

To launch the analog simulation environment, click on "Launch"->"ADE L". If you get a pop-up window asking about using a different license, click "yes". You should now see a window similar to figure 6.

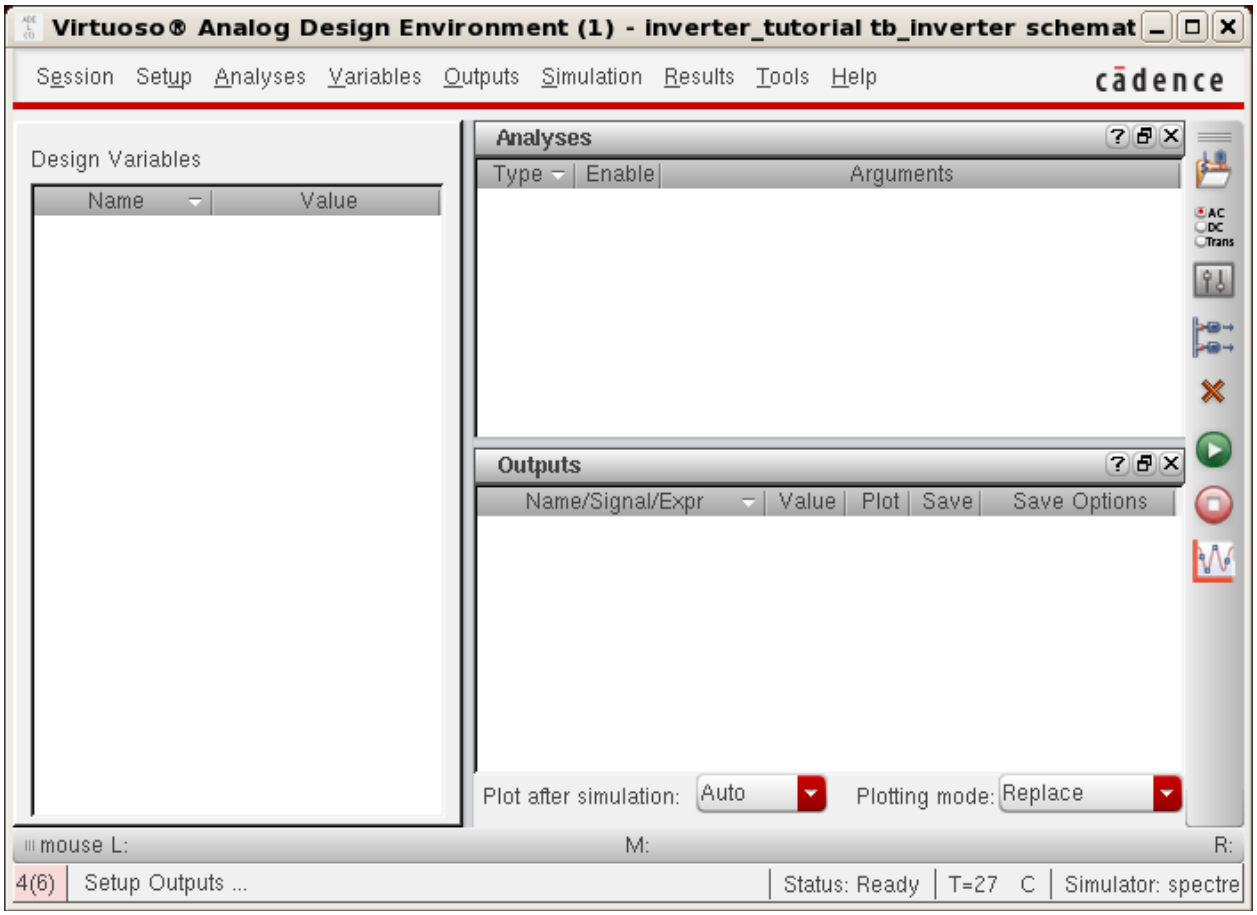

Figure 6. ADE window.

We will do DC and transient simulation, so click on "Analysis"->"Choose" and click on "tran" in the top of the new window. For the stop time, enter "1u", and click on "moderate" for the accuracy. Then click on apply. Click the "DC" selection at the top of the window to edit the DC simulation. Chose "Save DC Operating Point", then click ok. Now click "Variables"->"Edit". Click "Copy From" to have the tool read the schematic and create all the variables automatically. Set fclk to 10M and vdd to 1.3. Make sure you hit "Change" after entering *each* value, or else any changes you make are lost. Once you have set the variables, click ok.

There is one more step before we can run the simulation. The library has to be added that contains all the IBM components. To do this, click on "Setup"->"Model Libraries". In the pop-up window, click on the "…" button in the first entry of the list, then browse to the file

/home/jwilson/cadence_install/pdks/ibm/IBM_PDK/bicmos8hp/relHP/Spectre/models/allModels.scs and click open. Click ok on the library setup window.

Now click on the green arrow to run the simulation. Once the simulation is complete, you can plot results by going to "Results"->"Direct Plot"->"Main Form …" to open the direct plot window. Since we only ran a DC and transient simulation, and DC doesn't have anything to plot, only the transient is available in the analysis selection. With the direct plot window open, click on the wire leading to the cap and the wire leading to the input of the inverter. Each time you click on the signal, the plot is added to the window.

Layout

Close the schematic window with the testbench in it, as well as the simulation results and Analog Design Environment window. In the Library Manager window, click on "inverter_tutorial", "inverter", and then double click the schematic view. This will launch the schematic view of the inverter. Click on "Launch"->"Layout XL" to launch the layout window. In the pop-up window, choose "Create New" for the Layout and "automatic" for the Configuration, and click ok. A "New File" window will open. Leave these as the default options and click ok again. You will now have three windows open—the LSW window, shown in figure 7, the schematic window you originally opened, and a blank layout window.

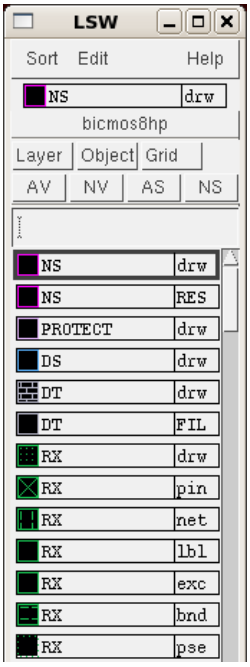

Figure 7. LSW window

The first step in creating a new schematic using the Layout XL flow is to have the tool generate all the layout pcells. This is done by clicking on “Connectivity->Generate->All From Source”. The window shown in figure 8 will open.

Figure 8. The generate window.

Unclick the “PR Boundary” in the Generate tab. Under the I/O Pins tab, drag and select all four of the pins in the “Specify Pins to be Generated” pane. The layer on which the pins will be generated needs to be changed from PC, which is poly, to M1, the lowest metal. With all four pins selected, change the layer dropdown to M1,pn. Then click the update button. Notice that the layer for all four pins now shows (“M1” “pin”) instead of (“PC” “pin”). Click ok to generate the layout.

You should now see a screen similar to figure 9. The initial view has the display stop level set to 0, so that all components in the hierarchy below this level will be hidden, and only the boundary of those blocks will be seen. To change this, click on “Options”->“Display” to open the display options window. Here you can change the display levels so that the stop level is much deeper. For this tutorial, you only need to change the stop to 1, but I prefer 5. Click ok to set the changes.

You will notice that the transistors don’t look quite right. This is because the default size for the pcell is too small to allow for the contacts to be created. So click on the pfet, then click “shift-q” to bring up the properties. Change the “Width Single Finger” to 300n, and click ok. The pcell

will be updated with the new width, and now you can see metal and contacts at the source and drain.

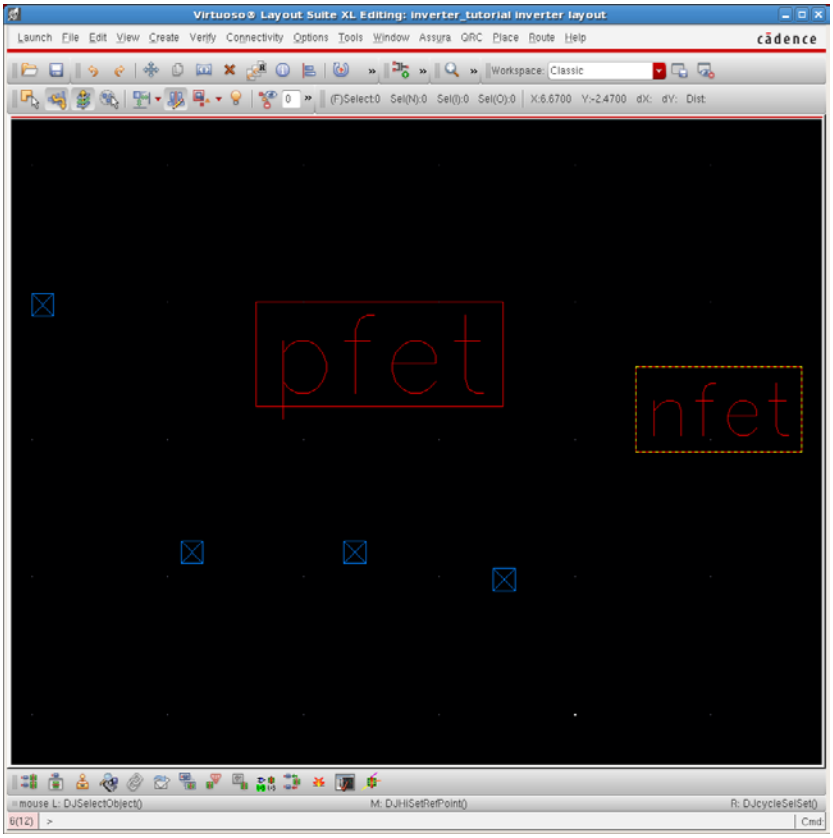

Figure 9. Initial layout.

The initial snap size for the layout is set to 0.1, which is too large. Change this by clicking "Options"->"Display", and altering the "X Snap Spacing" and "Y Snap Spacing" to 0.01. Now select the nfet. Move it by clicking "m" and then clicking on the selected nfet. This will pick it up. Move the mouse to the area below pfet. You will notice that the tool will only let you move it orthogonally. You can edit this by pressing F3 to bring up the command options window. Here, change the snap mode from orthogonal to any angle. There is one other feature that you might not be used to, and this is called gravity. Notice that when you move the mouse around, it will snap to the corners and other objects as you get near them. You can control this feature by pressing "g" to turn it on and off. Using the move command, line up the pfet and nfet vertically.

Next we will move the pins to their appropriate locations. You can see that the pins do not have any text to tell you what it what. But this is fine, because the tool is synchronized between layout and schematic. So as you click on a pin, it is highlighted in the schematic, allowing you to see which pin you selected. Or you can click the pin in the schematic and it will select it in the layout, as well. Using the move command, move the pins so that the input is on the left, the

output on the right, the vdd on the top, and the gnd on the bottom. Your layout should look similar to figure 10.

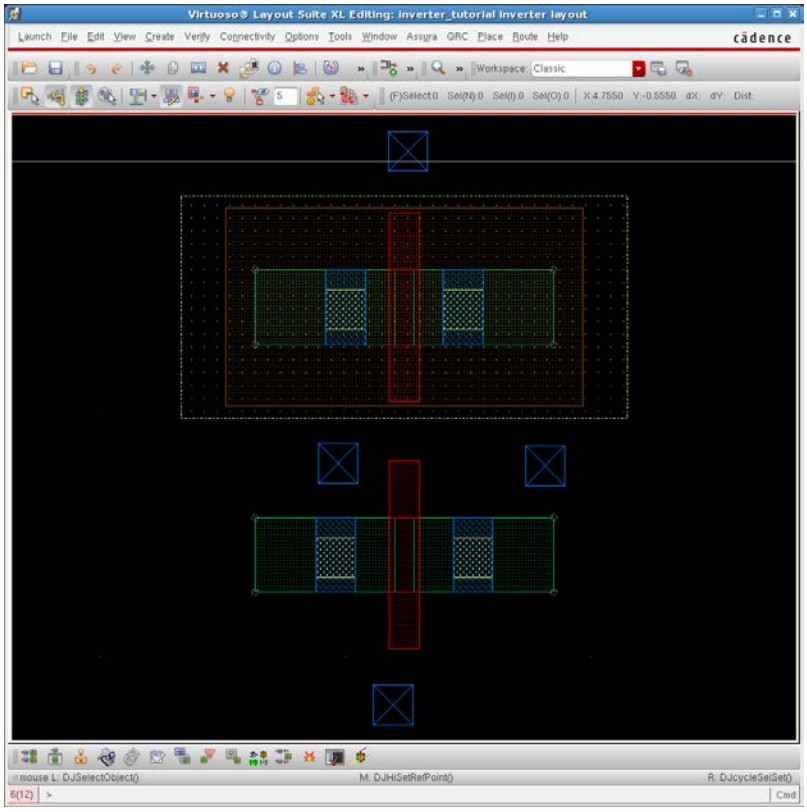

Figure 10. Updated layout with components moved into location.

Now we will wire up the nets. One of the features that the layout XL gives you is the ability to see which nets need to be connected at which locations. To use this feature, click on "Connectivity"->"Nets"->"Show/Hide Selected Incomplete Nets…". This will put you into the selection mode. Now click on each of the pins to see where they need to be wired to. You can click on them again to turn off this feature for that net.

Let's start by wiring up the gnd pin to the source of the nfet. There are two ways to create a point-to-point connection in the layout. The first is the "dumb" way, which is the old way of doing layout. This doesn't take into account the connections, allowing you to make mistakes. The layout XL flow gives us the "smart" way, which knows where wires should go. To use the "smart" option, click "create"->"Wire". The keybinding for a wire is "ctrl-shift-w", so click "ctrl-shift-w" and the click on the right side of the gnd pin. This will start the wire command. Press F3 to bring up the wire options window. Here you can change the width and snap mode. Wire the pin to the source (left side) of the nfet. You can see this is where the pin connects to using the connectivity option discussed before. Click once to make a turn and either double-click or press "enter" to terminate the wire.

Repeat this to connect the vdd to the source of the pfet, and the output to the drains of both the n and pfet.

To connect the input, we are going to do things a little differently. First, let us change the pin from metal 1 to poly so that we don't have to create a contact in the middle of our inverter. To do this, select the pin, then press "shift-q" to bring up its properties. Using the layer dropdown, change it from m1,pn to pc,pn, then click ok. Notice now that the pin is red instead of blue. Now use the wire tool to create a connection in poly between the gates of the two fets. Set the width of the connection to 0.12 so that it is the same width as the poly gates. Now move the input pin on top of this poly wire. You will notice that the pin is wider than the poly. DRC will complain about the pin not being covered, so we need to adjust the size of the pin to be contained in the poly. To do this, we will use the "stretch" command. The keybinding is "s", so press "s". Then turn off gravity so you can select the edge of the pin. Click the edge, and then slide it over so that it is contained in the poly routing. The new layout should look like figure 11.

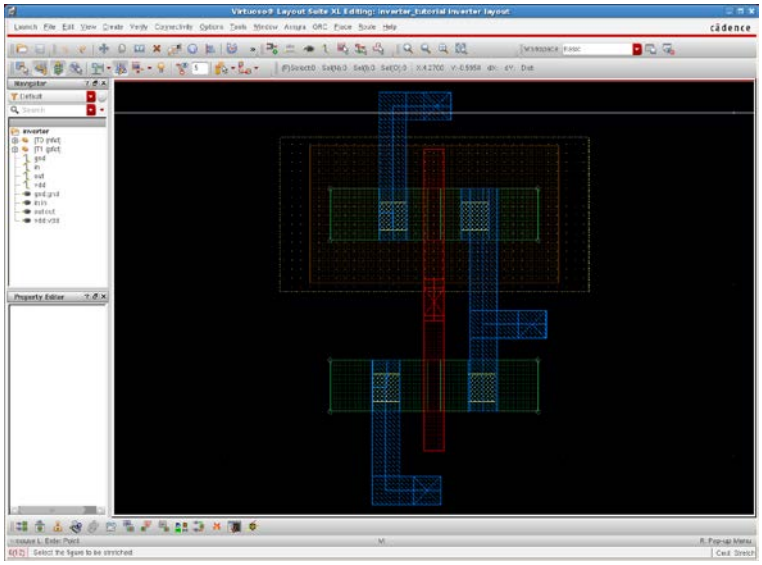

Figure 11. Updated layout with routing.

The layout isn't quite complete; we need to provide contacts to the substrate and well for the nfet and pfet, respectively. To do this, we must create a contact. For some reason, however, the contacts that connect down to the substrate and nwell are not available in the default contact creation tool. There are several options available to place substrate contacts. The first is to switch to the Layout L flow by going "Launch"->"Layout L", then "IBM_PDK"->"Misc"->"Add substrate/nwell contact based on RX". Then draw a rectangle to define the size of the contact. Another way is to stay in the Layout XL flow, and insert them as you would in a schematic. So press "i", browse to the bicmos8hp library, contacts category, and select subc for the substrate contact or nwCont for the nwell contact. Place one of each, making sure to abut the nwell contact to the nwell of the pfet. Connect them to vdd/gnd, and your finished layout should look like figure 13. However, the easiest way for pfets is to use the pcell to generate the contact

for you. Click the pfet to select it, and press “shift-q” to open the properties. In the parameters tab, scroll down to “Add NW contact”, select it, and click ok.

Now that the layout has contacts, we need to include a matching circuit in the schematic. This is unique to the IBM 8HP flow. In the schematic, you have to add the element subc or subc_inh from the bicmos8hp library, category contacts. Wire it up so that the bottom of the element connects to the substrate of the nfet, and the top connects to the ground (source). The schematic and layout are shown in figures 12 and 13.

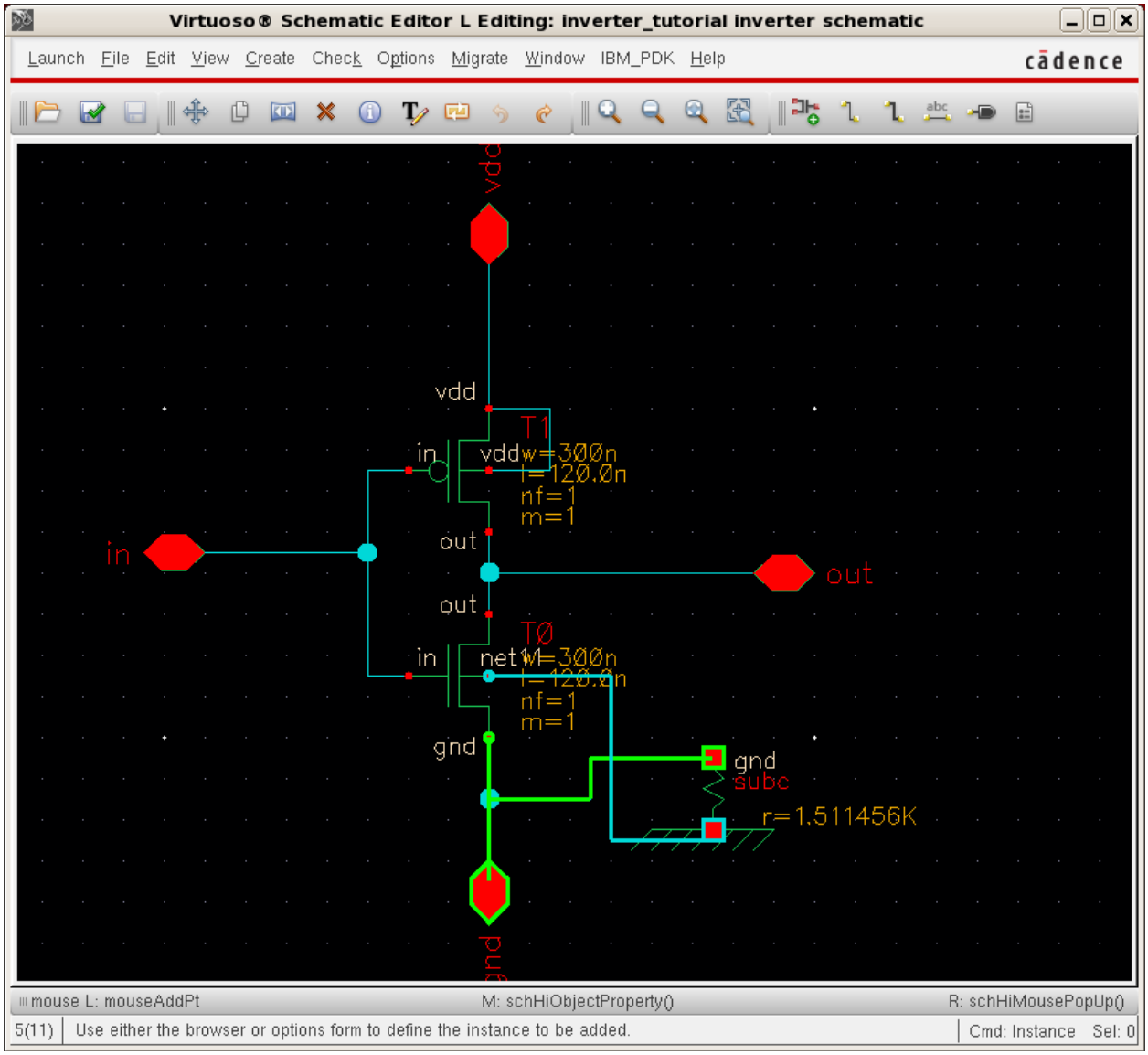

Figure 12. Finished schematic.

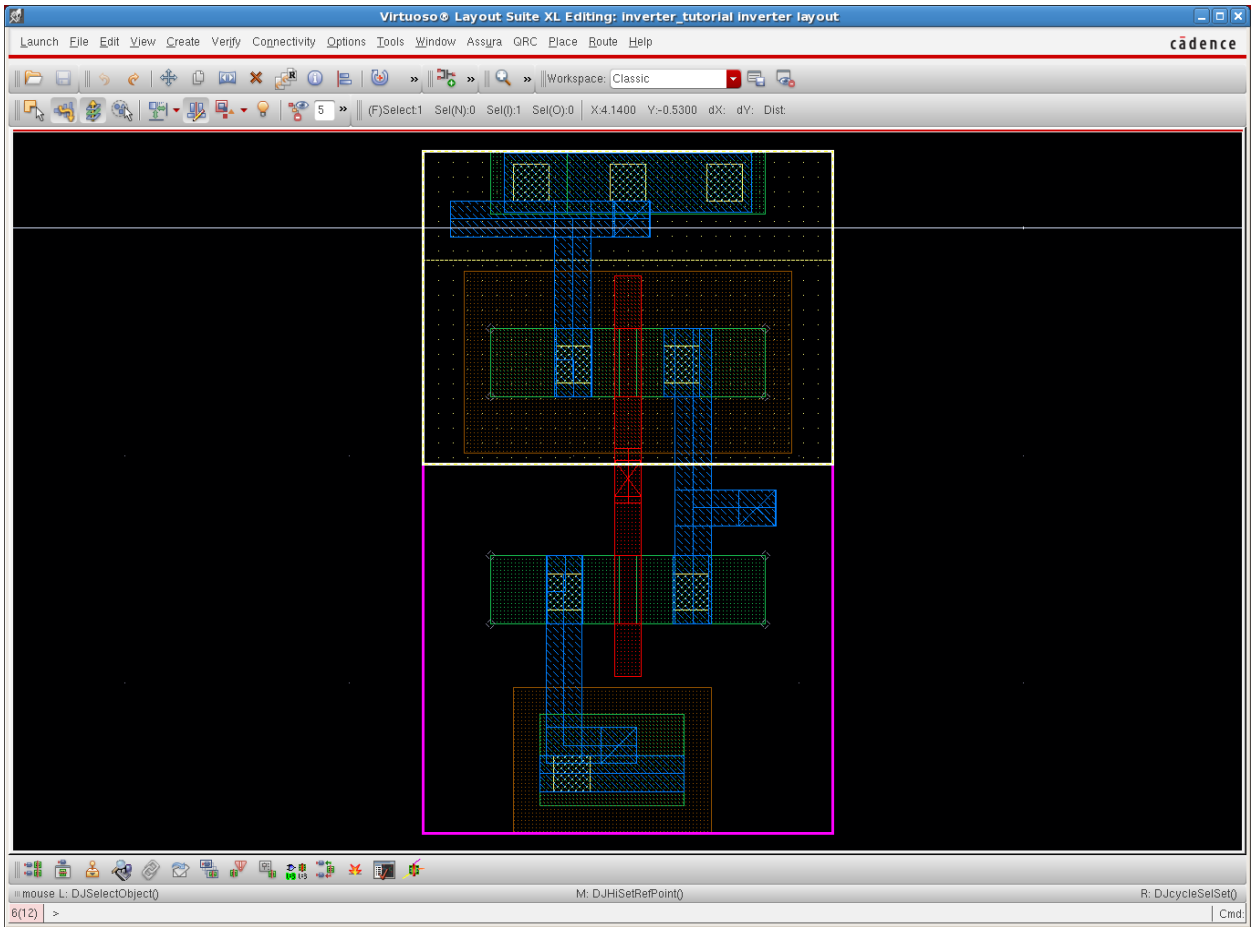

Figure 13. Finished layout.

Design Rule Check (DRC)

Once the layout is finished, the next step is to run DRC. To setup DRC from the layout window, click “Assura”->“Technology”, then browse to select the file “/home/jwilson/cadence_install/pdks/ibm/IBM_PDK/bicmos8hp/relHP/assura_tech.lib”. Click ok to load the file. Then go to “Assura”->“Run DRC” to launch the DRC window. In this window, choose “8HP_7LM” from the technology dropdown menu and “7LM” for the Rule Set. Then click ok to launch the DRC run. You will notice a small window pop-up that lets you know that DRC is running, and once it finishes, you will be prompted, asking if you want to view the results from the run. Click Yes. A new window opens, the “Error Layer Window”. This window lets you browse any errors in your design. First click on the error in the left side of the window, then use the left and right arrows on the right side to scroll through each instance of that error. Correct any errors in the design, and rerun DRC. Repeat this until your design is DRC error-free.

Layout vs. Schematic (LVS)

LVS is run very similar to DRC. Since we already setup the Assura technology file, we can run LVS by choosing "Assura"->"Run LVS". In this window, select 8HP_7LM for the technology, and "7LM" for the Rule Set. Click ok to start LVS, and wait for it to finish. There should not be any errors in the LVS, but if there are, you can use tools that Cadence provides to debug your circuit.

Extraction

Extraction is performed with the QRC tool from Cadence. In the layout window, click "QRC"->"Run Assura-QRC". Choose a directory in which to store the results; you can leave this as default. The next window is where you will set the options for the extraction. In the Setup tab, Technology should be "8HP_7LM" and the RuleSet "7LM". Change the "Output" dropdown to "Extracted View". This view will produce a Cadence schematic that can be opened like a regulator schematic. You can leave the rest of the options in this tab as default.

In the Extraction tab, you can choose the extraction type. Most circuits in the analog regime will work best with a RC extraction, so that is what we will choose. You can leave the rest of the options as default.

The rest of the tabs control other extraction options, but in this case you can leave them as default, as well.

Click ok, and the extraction will run. This can take several minutes, even on a fast computer. When it does complete, you should see a pop-up window informing you that the run completed successfully. If it does not, you will have to go back through the log file to figure out what part of the run failed.

To see the extracted schematic, go to the Library Manager window and browse to the inverter tutorial library, and click on the inverter cell. Notice that there are several new views for the cell. The one we are interested in is the av_extracted view. Double-click this view to open it. The extracted schematic will look like figure 14. You can zoom in to the routing and see the location and size of the parasitic resistors and capacitors that the extraction added.

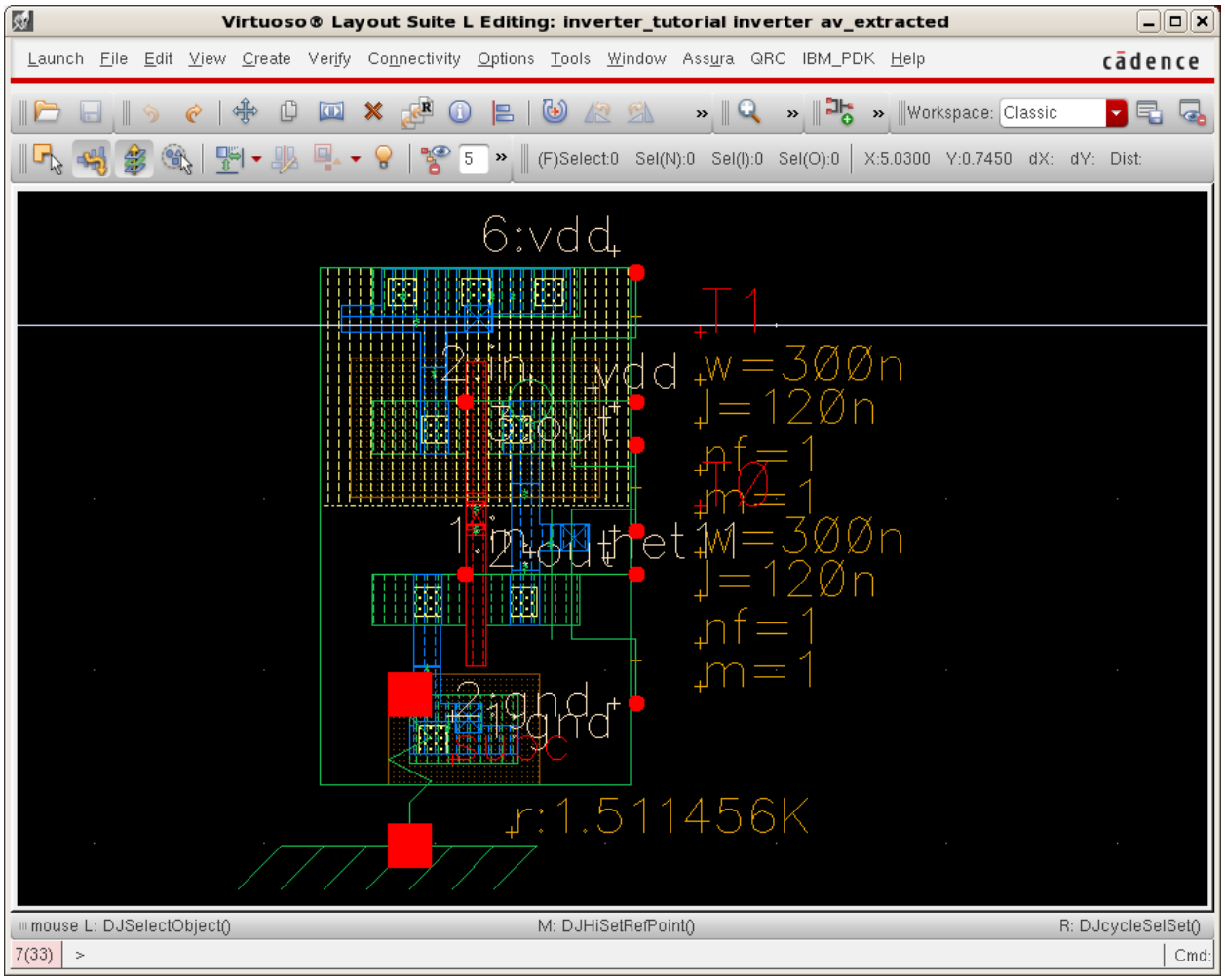

Figure 14. Extracted schematic.

Post Layout Simulation

There are some extra files that need to be created to allow for this schematic to be used in the simulation in place of the default schematic. Click on the inverter_tutorial library, tb_inverter cell. Then click “File”->“New”->“Cell View”. In the “New File” window, make sure the cell is tb_inverter and choose “config” from the Type dropdown box. This should change the view to config; click ok. You will now be presented with a “New Configuration” window on top of the hierarchy editor. At the bottom of this window, choose “Use Template” and select “spectre”. This will populate the window with some default settings. Change the “View” to schematic, and the “Library List:” to inverter_tutorial. Your window should look like figure 15. Click ok to go to the Hierarchy Editor window.

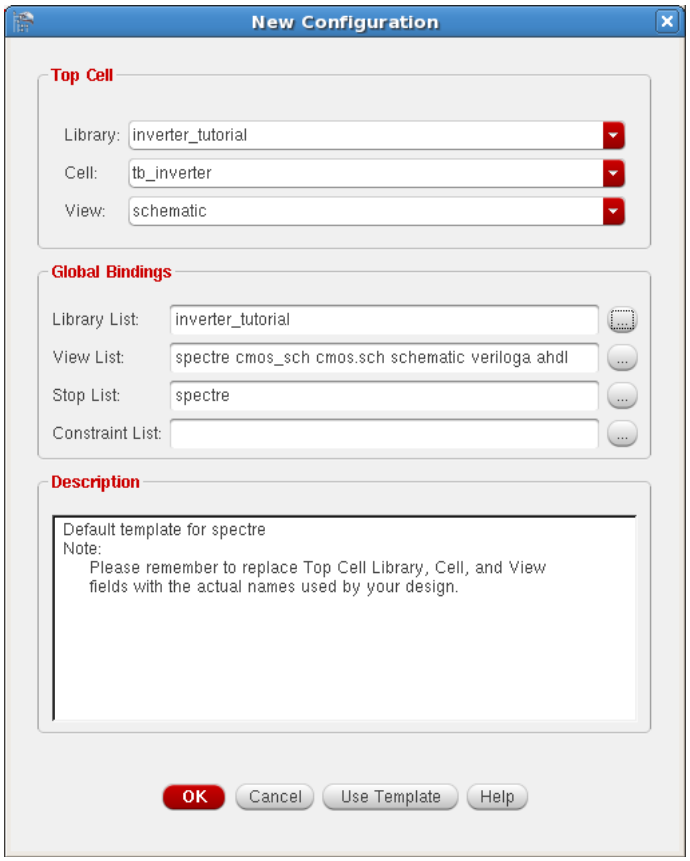

Figure 15. Configuration setup.

The first thing to do here is save the config, so click the “save” button. This window allows you to set what views are used for which blocks in the design. The tool starts at the Top Cell, defined in the upper-left corner, and traverses the design hierarchy using the view list, until it finds a view listed in the stop view list. This is the same algorithm that is run when a netlist is created by the simulator. All the cells and their corresponding views are listed in the table view. This is shown in figure 16. You can open the cells from this window by right clicking on the cell, and choosing open. Go ahead and do this for the tb_inverter cell. This will open up the testbench schematic.

Notice at the top of the schematic window that “Config: inverter_tutorial tb_inverter config is now displayed”. This is how you know that the config view is controlling the netlisting of this testbench. So open the simulator, load the previous state, and run the simulation. This is the schematic only simulation, to give us something to compare the extracted results to. Plot the transient output of the inverter.

Now go back to the config window and right-click on the cell inverter. Choose “Set Cell View”->“av_extracted” to choose the extracted view just created. Now that we have changed the view being used, we have to recompute the hierarchy. The hierarchy is recomputed by selecting “View”->“Update”. Click ok on the new pop-up window. Now go back to the simulator

window and click run. Replot the output, and notice that the plot is different. As shown in figure 17, it will only be slightly different, as the parasitics associated with a single inverter are very small, but it will be different.

With the flow now complete, you could go back and edit the inverter to update the sizes if the parasitic elements changed the results too much, and rerun the whole process again. If you are happy with the results, the block can be considered complete.

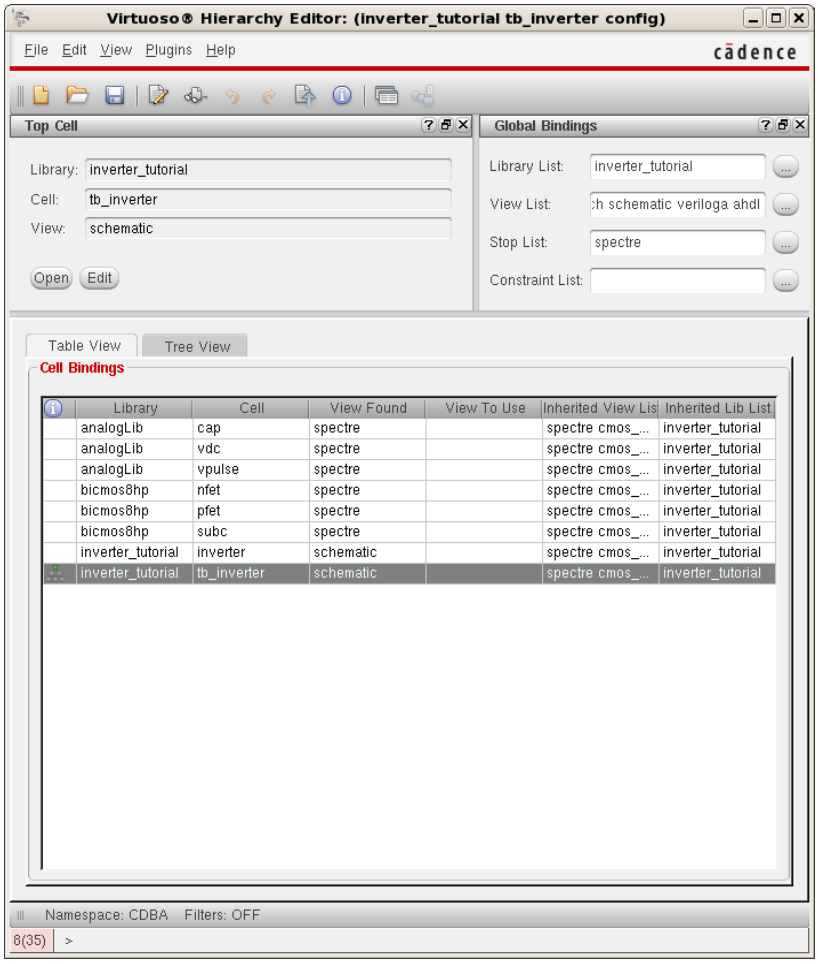

Figure 16. Config view.

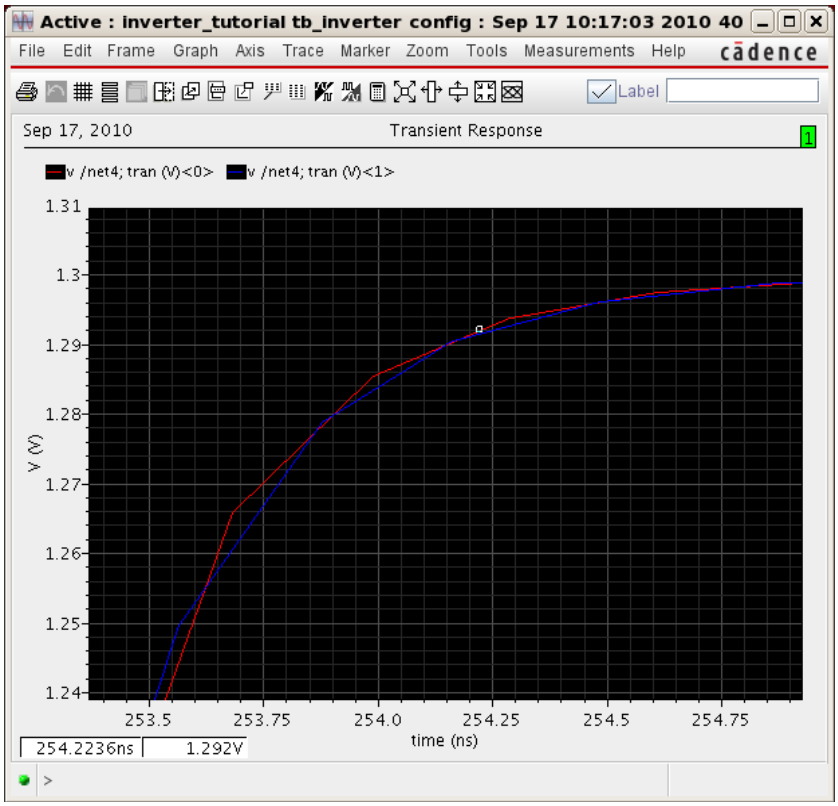

Figure 17. Extracted simulation in blue showing difference.

List of Symbols, Abbreviations, and Acronyms

ADE	analog design environment
ARL	U.S. Army Research Laboratory
CIW	command interpreter window
DC	Direct Current, also refers to zero frequency steady state condition
DRC	design rule check
IC	integrated circuit
LVS	layer versus schematic
LWS	layer selection window
NMOS	N-type metal oxide semiconductor
PC	polycrystalyn silicon
PDK	process design kit
RF	radio frequency

NO. OF COPIES	ORGANIZATION
1 ELECT	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1 CD	OFC OF THE SECY OF DEFNS ATTN ODDRE (R&AT) THE PENTAGON WASHINGTON DC 20301-3080
1	US ARMY RSRCH DEV AND ENGRG Cmnd ARMAMENT RSRCH DEV & ENGRG CTR ARMAMENT ENGRG & TECHNLGY CTR ATTN AMSRD AAR AEF T J MATTS BLDG 305 ABERDEEN PROVING GROUND MD 21005-5001
1	PM TIMS, PROFILER (MMS-P) AN/TMQ-52 ATTN B GRIFFIES BUILDING 563 FT MONMOUTH NJ 07703
1	US ARMY INFO SYS ENGRG CMND ATTN AMSEL IE TD A RIVERA FT HUACHUCA AZ 85613-5300
1	COMMANDER US ARMY RDECOM ATTN AMSRD AMR W C MCCORKLE 5400 FOWLER RD REDSTONE ARSENAL AL 35898-5000
1	US GOVERNMENT PRINT OFF DEPOSITORY RECEIVING SECTION ATTN MAIL STOP IDAD J TATE 732 NORTH CAPITOL ST NW WASHINGTON DC 20402

NO. OF COPIES	ORGANIZATION
1	US ARMY RSRCH LAB ATTN RDRL CIM G T LANDFRIED BLDG 4600 ABERDEEN PROVING GROUND MD 21005-5066
13	US ARMY RSRCH LAB ATTN IMNE ALC HRR MAIL & RECORDS MGMT ATTN RDRL CIM L TECHL LIB ATTN RDRL CIM P TECHL PUB ATTN RDRL SED E S BEDAIR ATTN RDRL SER E B HUEBSCHMAN ATTN RDRL SER E C WAIYAKI ATTN RDRL SER E E VIVEIROS ATTN RDRL SER E J QUI ATTN RDRL SER E J WILSON ATTN RDRL SER E K MCKNIGHT ATTN RDRL SER E R DEL ROSARIO ATTN RDRL SER E T FARMER ATTN RDRL SES E J A PENN ADELPHI MD 20783-1197
TOTAL:	21 (19 HCS, 1 CD, 1 ELECT)

INTENTIONALLY LEFT BLANK.