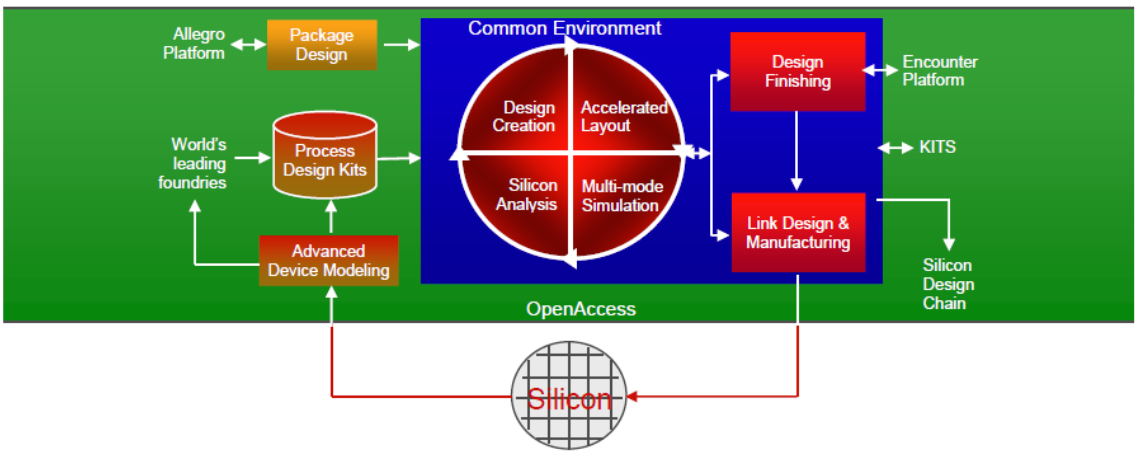
IC 6.1.6 Rapid Analog Prototyping Workshop

Analog Design Environment XL

Virtuoso Schematic Editor XL

Virtuoso Layout Suite XL/ GXL

(Supporting Releases: MMSIM and PVE)



IC 6.1.6 Rapid Analog Prototyping Workshop

Version 6.1.6
March 2015

© 2006-20015 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

Cadence Design Systems, Inc., 2655 Seely Avenue, San Jose, CA 95134, USA

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

All trademarks are the property of their respective holders.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Table of Contents

1 Introduction 3
2 Design Entry, Test Set-up and Pre-layout simulation 4
3 Interactive/ Assisted Constraint Creation in Schematic 9
4 Automatic Constraint Generation in Schematic 19
5 Automatic Constraint Driven Placement & Routing 26
6 Verification, Extraction and Parasitic Re-simulation 38
7 Quick iteration following changes in Schematic or Layout 47

1 Introduction

This workshop steps through a Rapid Analog Prototyping Flow in Virtuoso in IC 6.1.6. The objective of this flow is to generate the layout of an analog circuit in an automated manner, in order to obtain early feedback on parasitics and device effects on circuit simulation. The basic design goals and requirements are captured through a set of constraints in the schematic which are implemented in the layout through automatic placement and routing. The resultant LVS clean layout is then extracted, and the circuit re-simulated using the extracted data. The circuit designer can thus identify issues early on and make necessary changes to quickly iterate through the flow which helps avoid costly changes late in the cycle and enables faster design convergence. We demonstrate this flow using a sample and hold circuit based on a generic 45nm PDK.

- This workshop assumes that users are familiar with the usage of the following products:
  - Analog Design Environment (L and XL)
  - Virtuoso Schematic Editor (L and XL)
  - Virtuoso Layout Suite (L, XL and GXL)
  - Some level of familiarity with PVE (PVS and QRC)

- While this workshop follows the development of a design from concept through implementation, it is not a comprehensive treatment of any one tool in the flow. Additional product specific workshops, training courses, and methodology kits are available to provide more details.

- The beginning of each exercise will contain a statement of what is "In this section". This information will let you know what you will be investigating in that module. The statement looks like this:

In this section...

This module introduces the basics of the workshop organization.

- Actions that you will need to perform on the workshop machine are highlighted in this manner:

Action 1: This is something you need to do on the workshop machine.

- This workshop was validated on 09/22/13 with the IC 6.1.6isr13, MMSIM14.1isr3 and PVE12.1.1. Other versions are either unsupported or untested. To run this workshop, you will need to have these releases in your path and license for the same.

- Online help for each of the tools can be accessed either by selecting "Help" in the individual tools, usually the rightmost banner menu item, or as a Help button on individual forms, or by running "cdnshelp" from a terminal window.

2 Design Entry, Test Set-up and Pre-layout simulation

In this section...

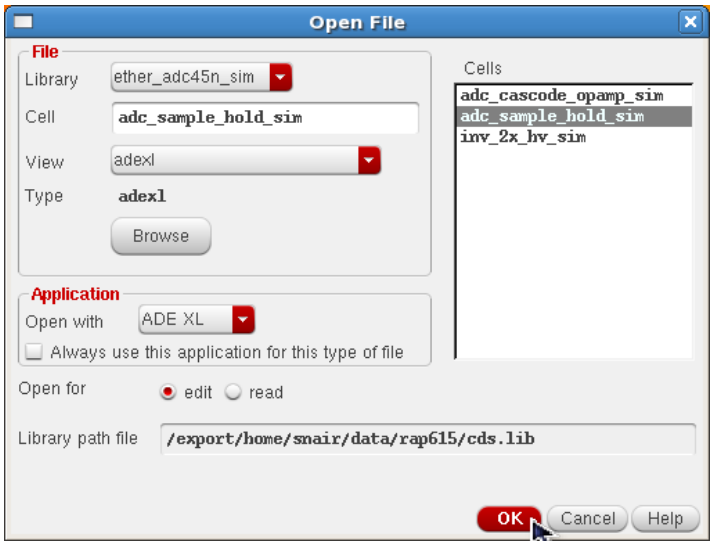
We will review the test set-up and run the tests in ADE-XL environment.

Action 1: cd WORK

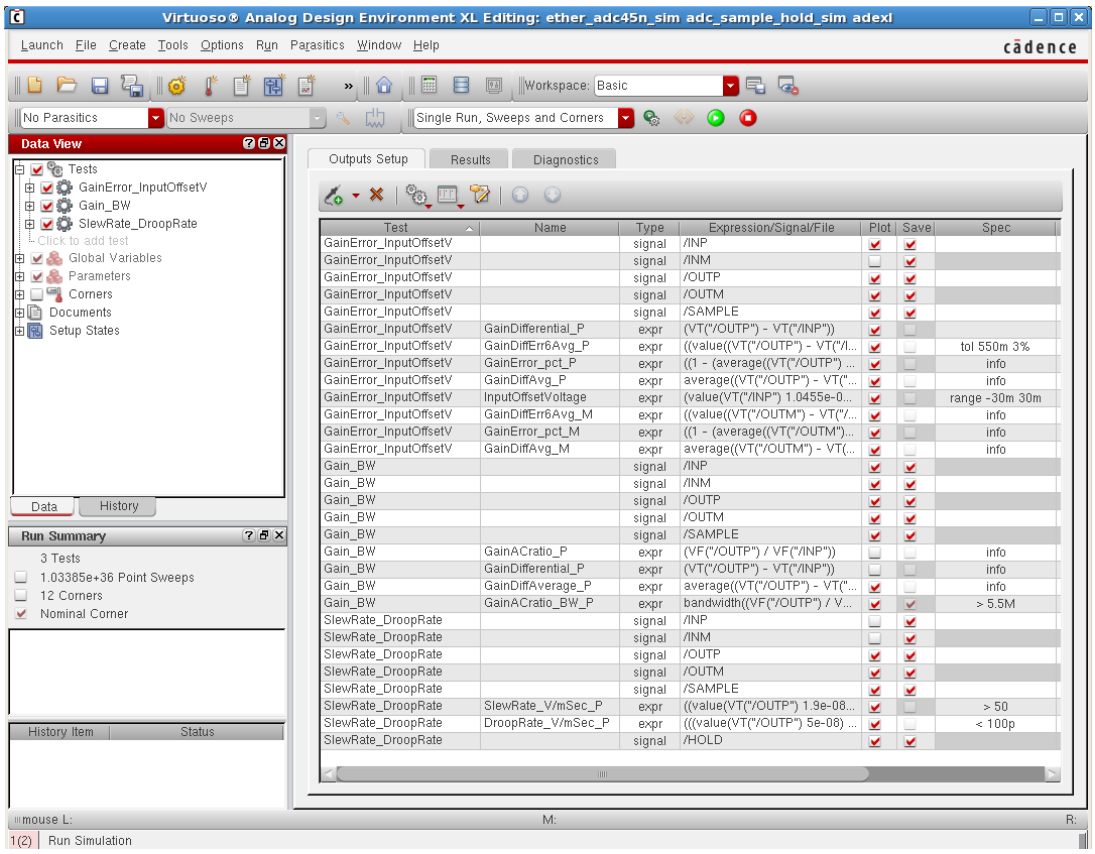
Action 2: virtuoso &

Action 3: From the CIW menu pull-down, click on 'File > Open' to open the following:

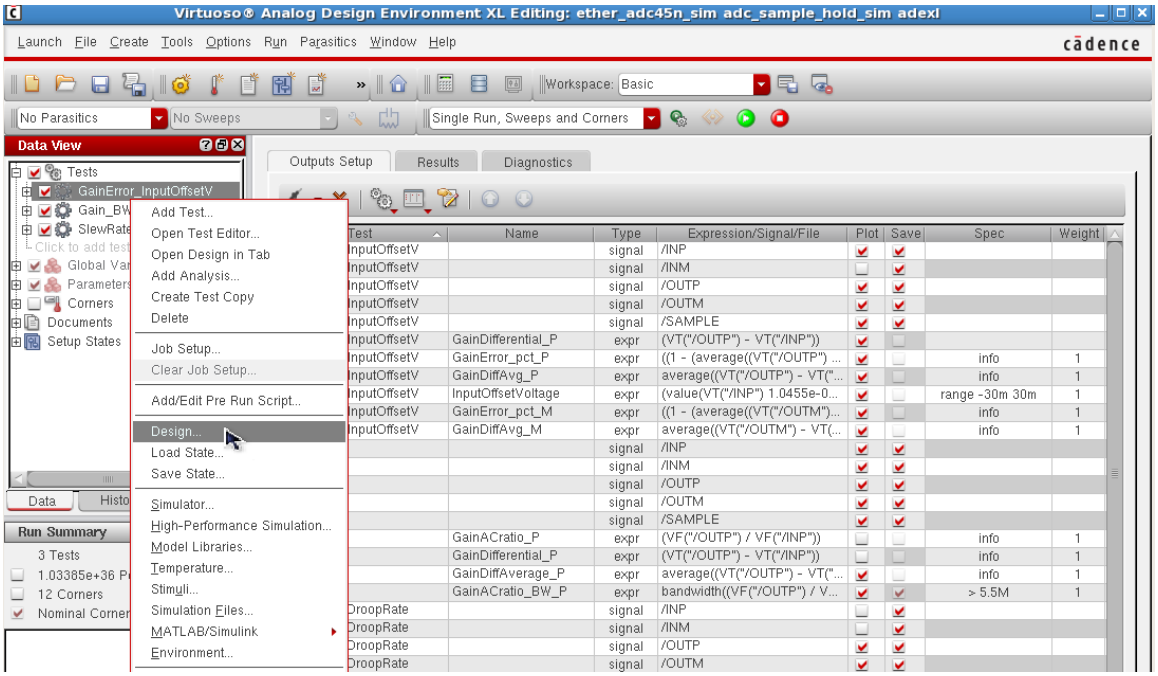
Library: ether_adc45n_sim
Cell: adc_sample_hold
View: adexl



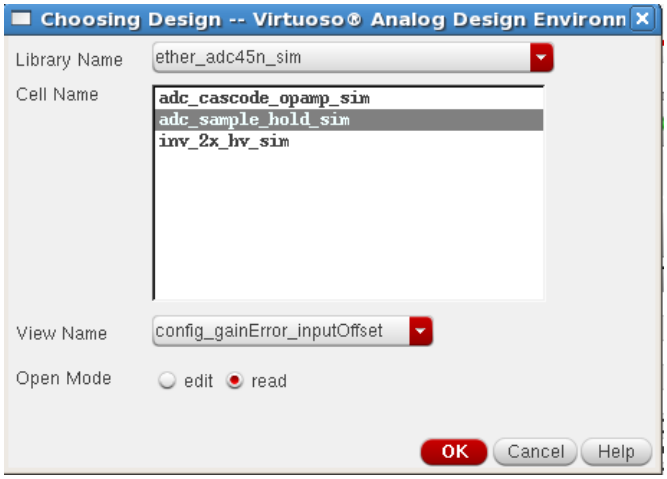
Action 4: In the ADE-XL window, under 'Data View' (left pane), click on 'Tests' to expand and view the three tests created for this design. The 'Output Setup' tab on the right shows the setup and the specs associated with the tests.



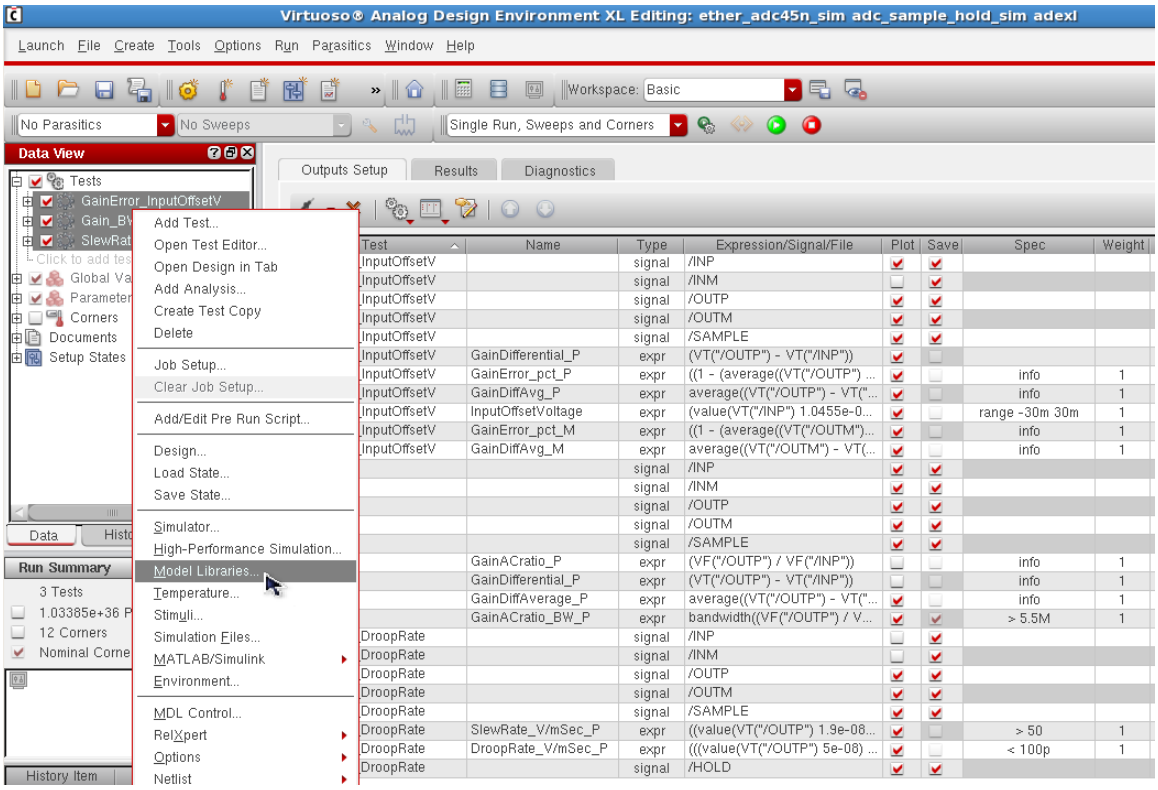
Action 5: Select the first test (GainError_InputOffsetV) and using the right mouse click, bring up the context menu associated with the test. Click on 'Design'.

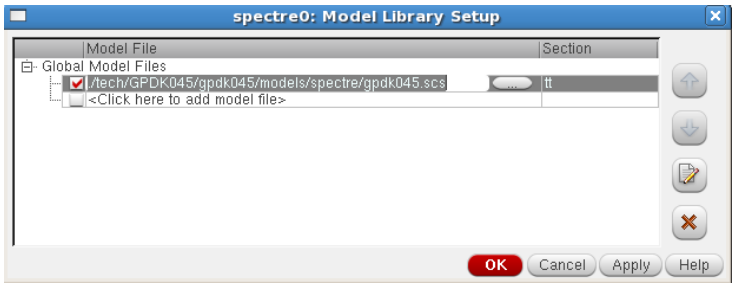


Action 6: In the form that comes up, you can see the config view that is associated with this test. You can open the config view in the Hierarchy Editor to see the viewlist and the bindings.



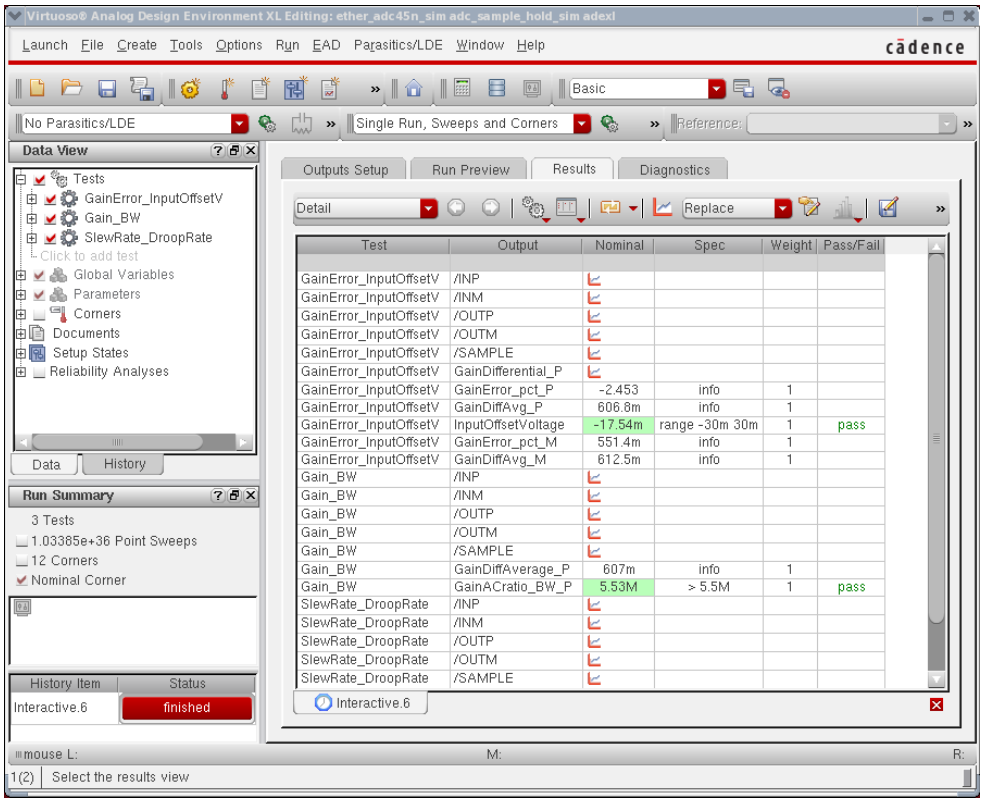
Action 7: Select all three tests and right click to view the RMB options to modify any of the setup. Click on 'Model Libraries …' to make sure that the path to the model file and the section (corner) are set up correctly. This step is important as the model file must have the correct path to run the simulation correctly. Both update and click OK or Cancel the form if the path is set correctly.



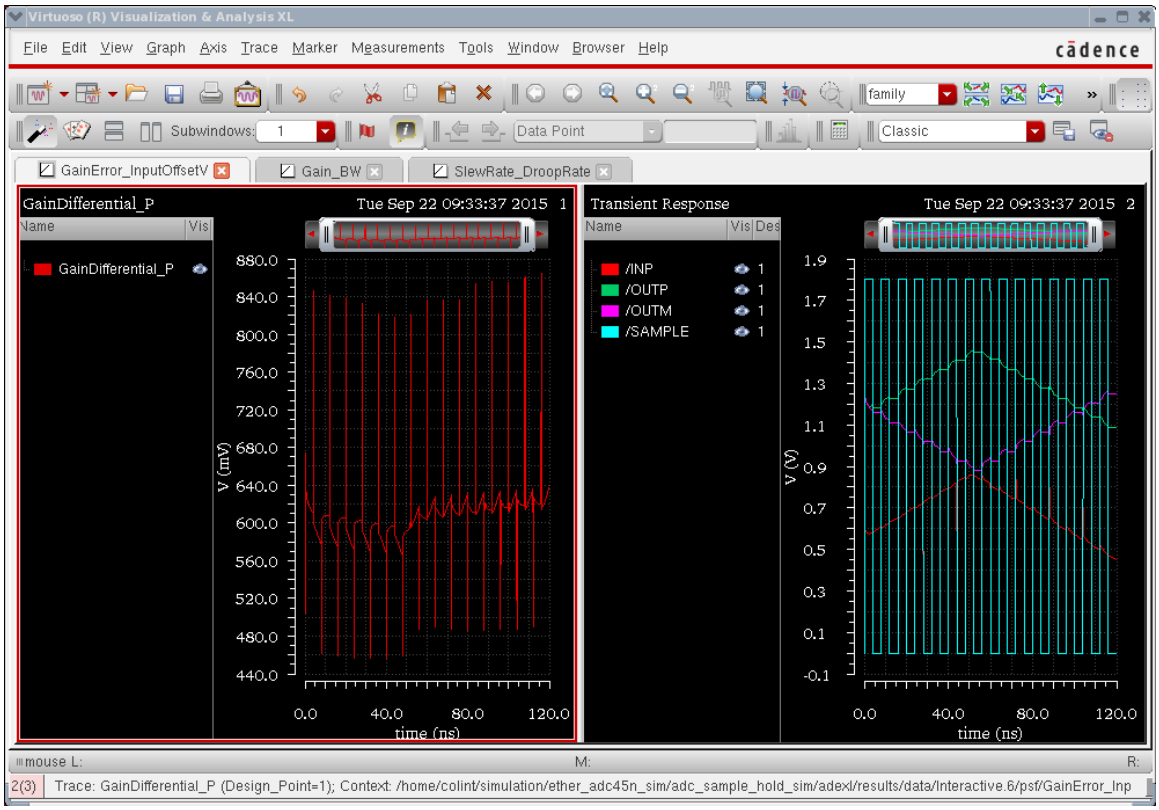


Action 8: Make sure the parasitic mode is set to 'No Parasitics/ LDE' (leftmost pull-down in the ADE-XL window) since we do not yet have the extracted data for simulation that will require the layout to be generated. Click on the 'Run Simulation' button to run all three tests as specified, and plot the results.

Action 9: Once the run completes, you can see the results under the 'Results' tab, and indication that all specs were met.



Action 10: You can also see the results plotted in the ViVA window for all three tests, in three different tabs.



Action 11: Quit the ADE-XL session without saving the changes.

3 Interactive/ Assisted Constraint Creation in Schematic

In this section...

We will see how constraints to drive the Layout can be created in the Schematic using the Constraint Manager, or captured with tool assistance using the Circuit Prospector.

Action 1: cd WORK

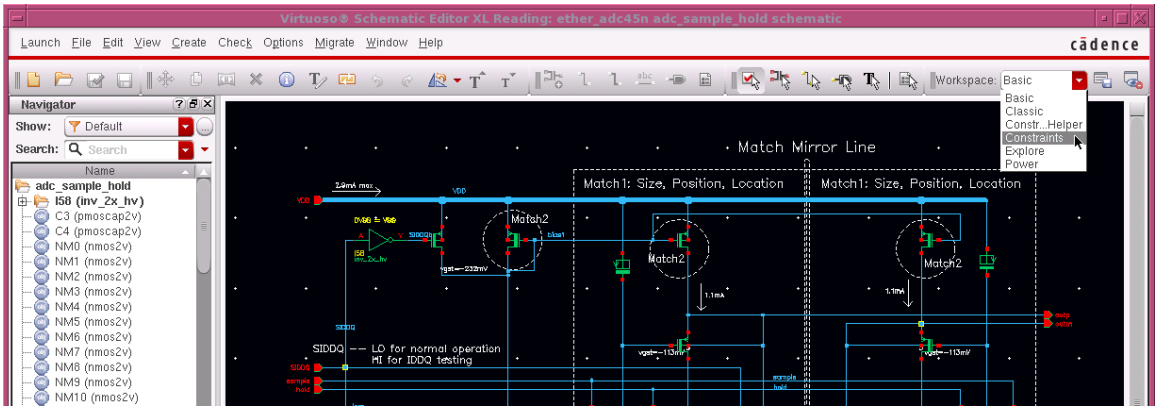
Action 2: virtuoso &

Action 3: 'File > Open' in the CIW:

Library: ether_adc45n
Cell: adc_sample_hold
View: schematic

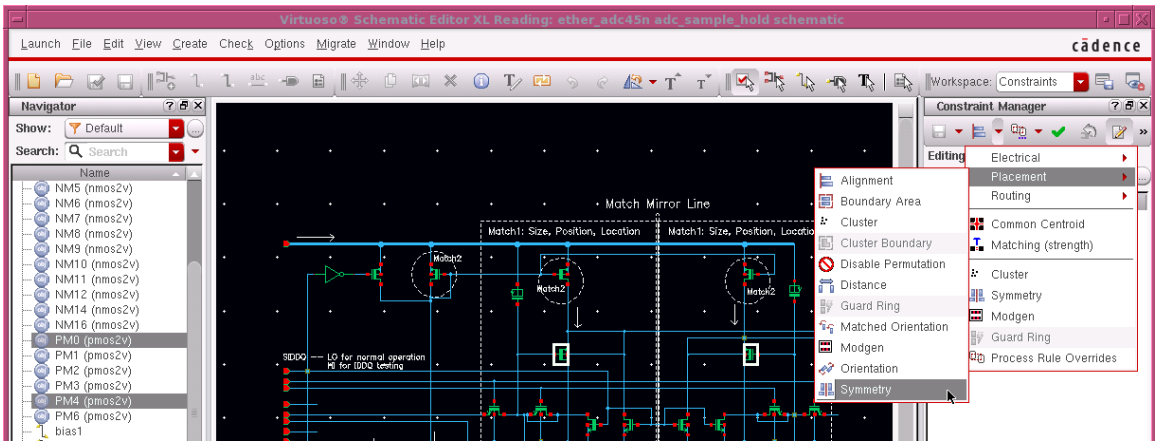
You can see that the Schematic has several annotations that are meant to guide the Layout. We will see how the Constraint System lets us capture these design intents

Action 4: Switch to Schematics XL ('Launch > Schematics XL') in order to access the Assistants that will help capture and enter constraints in the design. Choose the predefined Workspace 'Constraints' which will bring up the Constraint Manager Assistant. The Constraint Manager displays the default 'constraint' view which currently has no constraints defined.

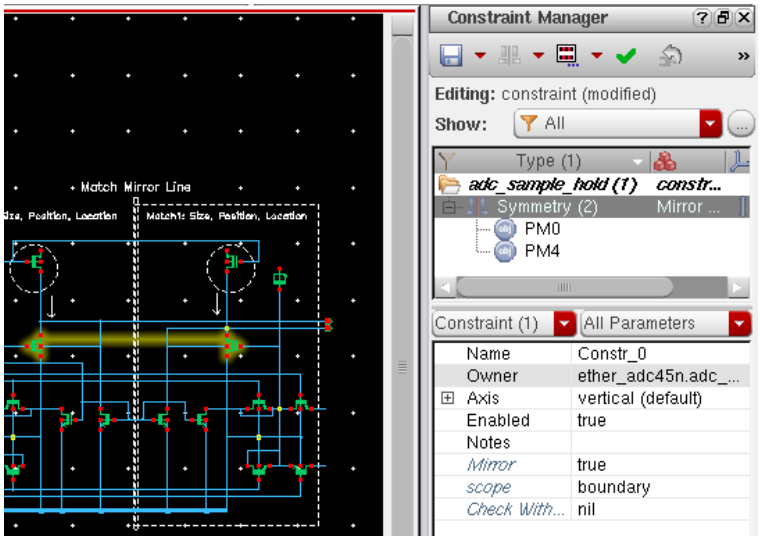


Action 5: Following the Schematic annotation, select instances PM0 & PM4 (in the Canvas or in the Navigator) to create a symmetry constraint on them since they are supposed to be matched and mirrored.

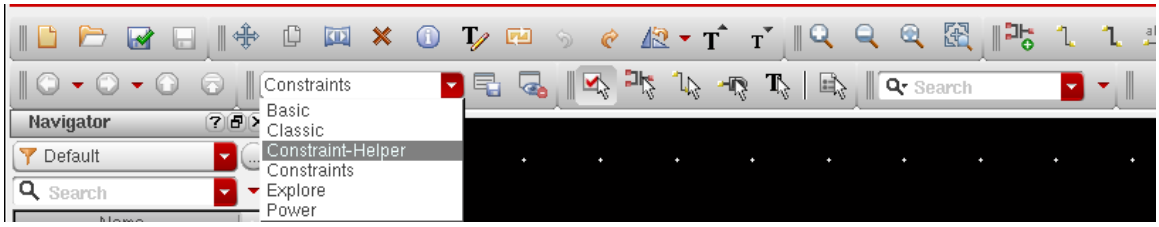
Action 6: Click on the ‘Constraint Creation’ pull-down in the Constraint Manager Toolbar (second icon from the left) and choose ‘Symmetry’ constraint from the ‘Placement’ category. You will notice that the constraints list is context sensitive i.e. only constraints that are relevant to the selected objects are highlighted while the rest are grayed out.



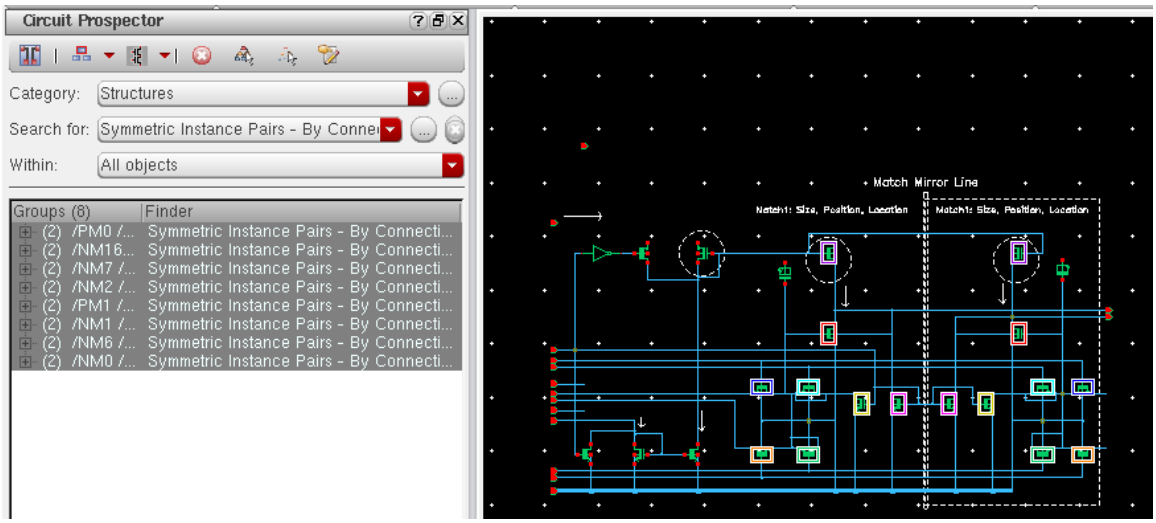
Action 7: Notice that a symmetry constraint shows up in the Constraint Manager and a halo appears in the canvas around the constrained instances. Clicking on the ‘+’ sign in the Constraint Manager expands the constraint to display its members. The parameters of the constraint such as axis information and type of symmetry appear in the Constraint Parameter Editor pane below.



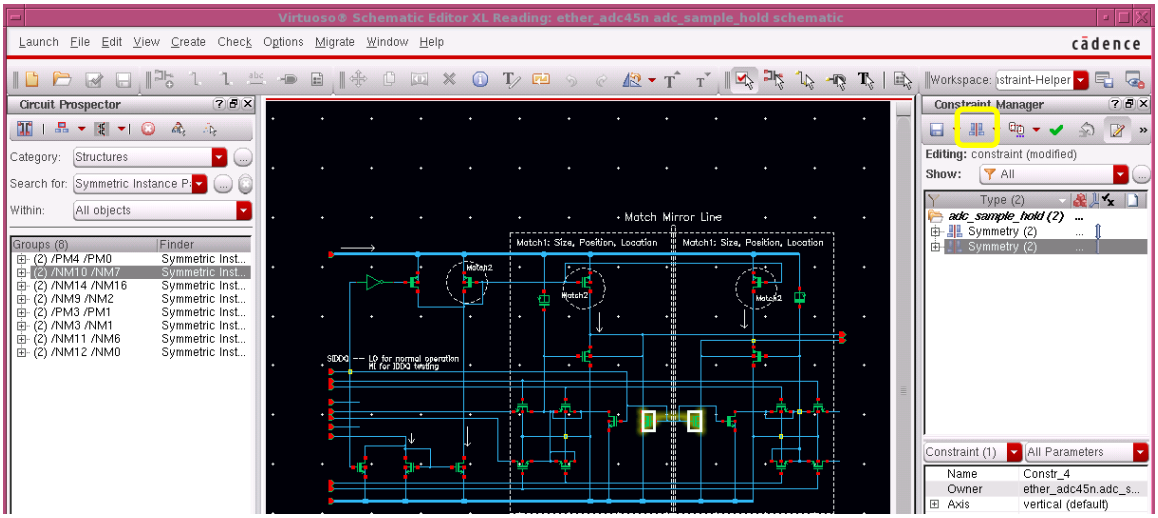
Action 8: Switch to workspace ‘Constraint-Helper’. This will bring up the ‘Circuit Prospector’ Assistant that can help identify groups of devices, nets and structures in the Schematic that are important for constraint creation, and optionally create a set of predefined constraints on them.



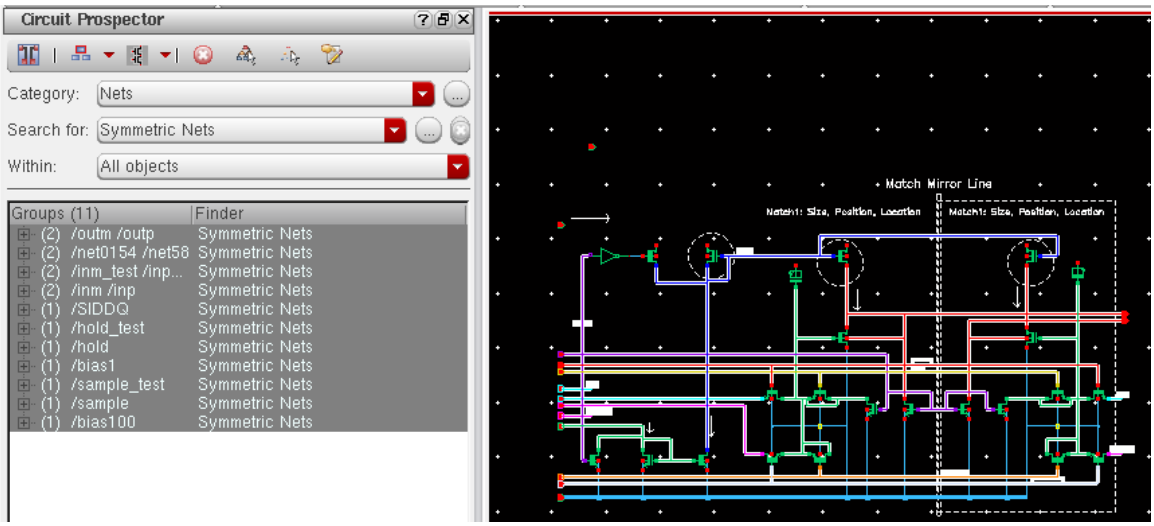
Action 9: In the Circuit Prospector, choose the ‘Structures’ category and under ‘Search for’, pick ‘Symmetric Instance Pairs - By Connectivity’. This returns instance pairs including those that are supposed to be matched along the mirror line as annotated in the Schematic.



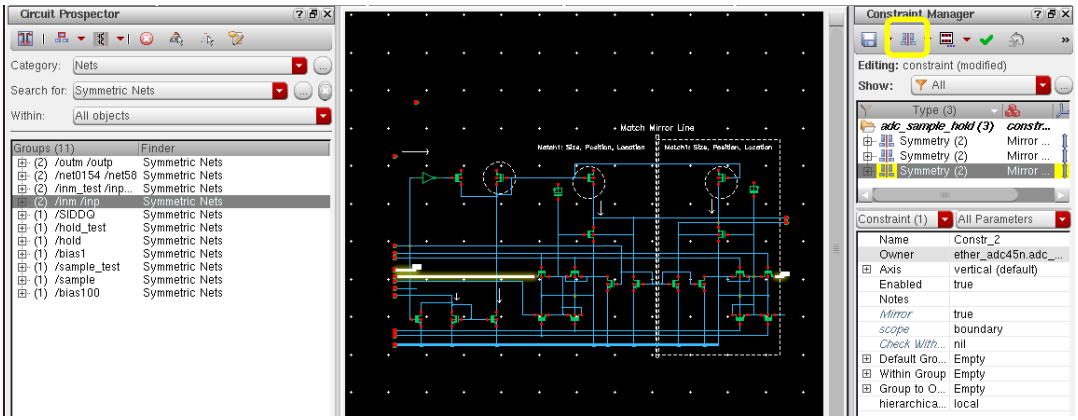
Action 10: You can apply ‘Symmetry’ constraint on all of them or on selected entries in the list by clicking on the Constraint Creation pull-down in the Constraint Manager, as shown below. Select the instance pair comprising of ‘NM6’ and ‘NM11’ (you can stretch the ‘Group’ column header to see the instance names clearly) and apply a ‘Symmetry’ constraint to it (the icon in the Constraint Creation pull-down updates to the last applied Constraint Type to facilitate repetitive entry, and hence you can simply click on that).



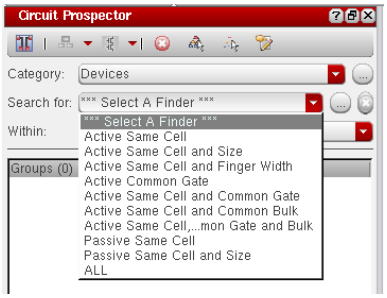
Action 11: Similarly, you can find symmetric nets in the design. In the Circuit Prospector, under Category, pick ‘Nets’ and search for ‘Symmetric Nets’ to identify nets that need to be routed symmetrically for matching. You can see that a number of pairs as well as single nets that qualify to be ‘self-symmetric’ are reported.



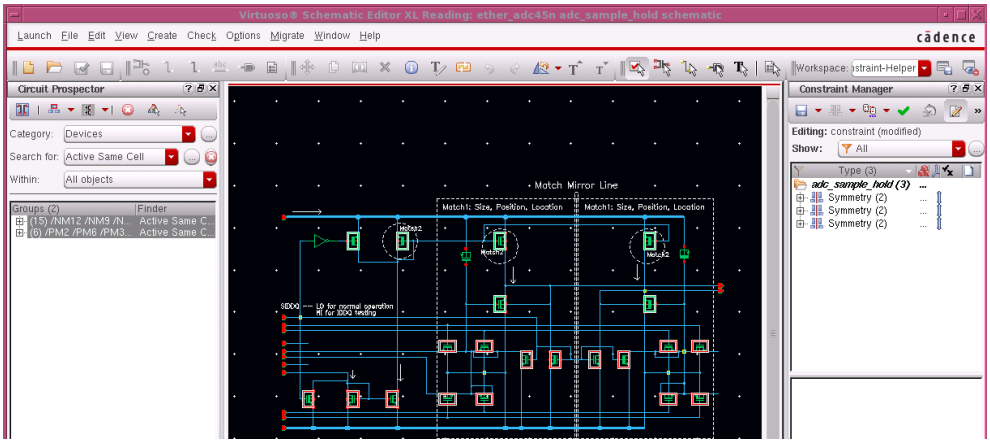
Action 12: Once again, you can apply ‘Symmetry’ constraint on all matches, or on selected pairs, by clicking on the Constraint Creation pull-down in the Constraint Manager, as shown below. Select the net pair comprising of ‘inp’ and ‘inm’ and apply a ‘Symmetry’ constraint to it.



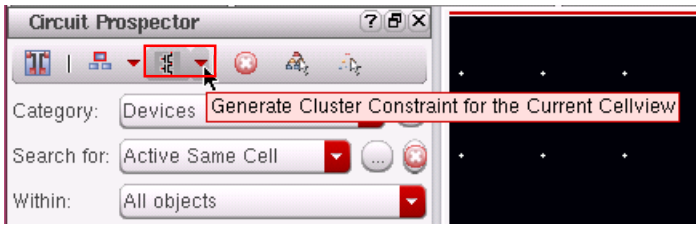
Action 13: Click on the ‘Category’ pull-down in the Circuit Prospector and switch from ‘Nets’ to ‘Devices’. Under the ‘Search for’ pull-down, you can now see the available finders that pertain to active or passive devices in the design.



Action 14: Select the ‘Active Same Cell’ finder to search the design for groups of active devices that are instances of the same cell. Notice how the Circuit Prospector returns two groups, one comprising of pmos instances and the other nmos instances, which have the same master.

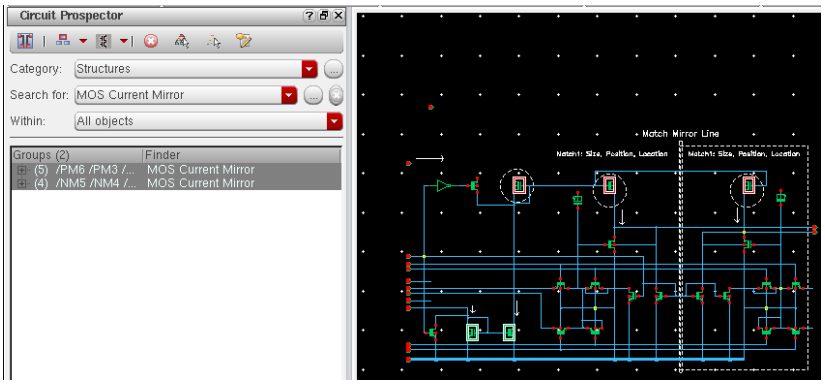


Action 15: Hover your mouse over the 3rd icon in the Circuit Prospector toolbar. This button helps users auto-generate relevant constraints on the matches returned by a finder. The tooltip shows that 'Cluster' and 'Matched Orientation' constraints will be generated for these groups of devices. Click on the icon to generate these constraints for the found structures.

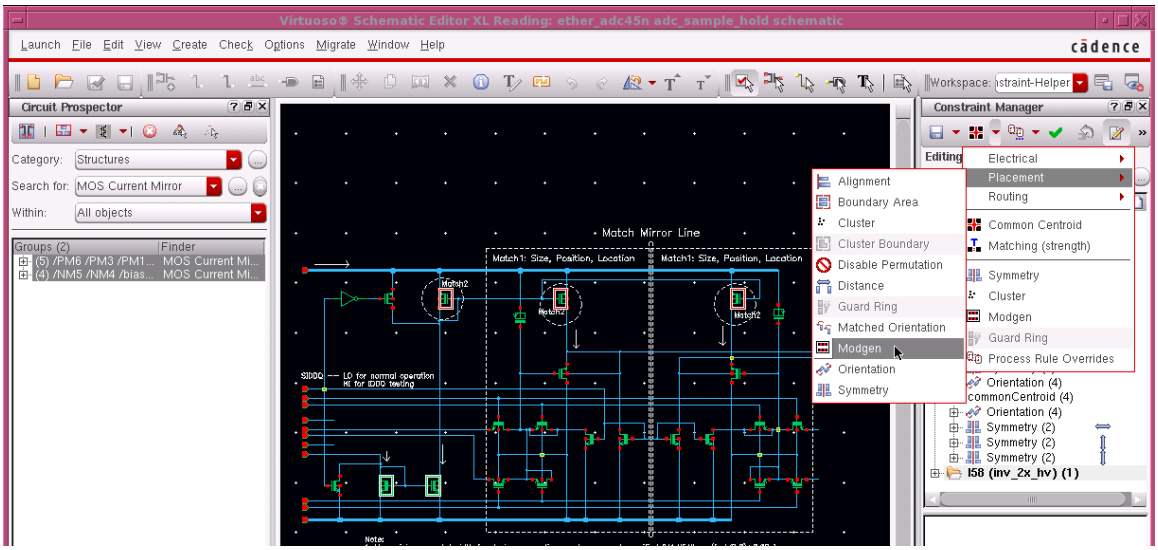


Note: The 'default' set of constraints associated with a given structure or device finder are meant to serve as examples. Users can customize them to specify the constraints that would be auto-generated for a given finder.

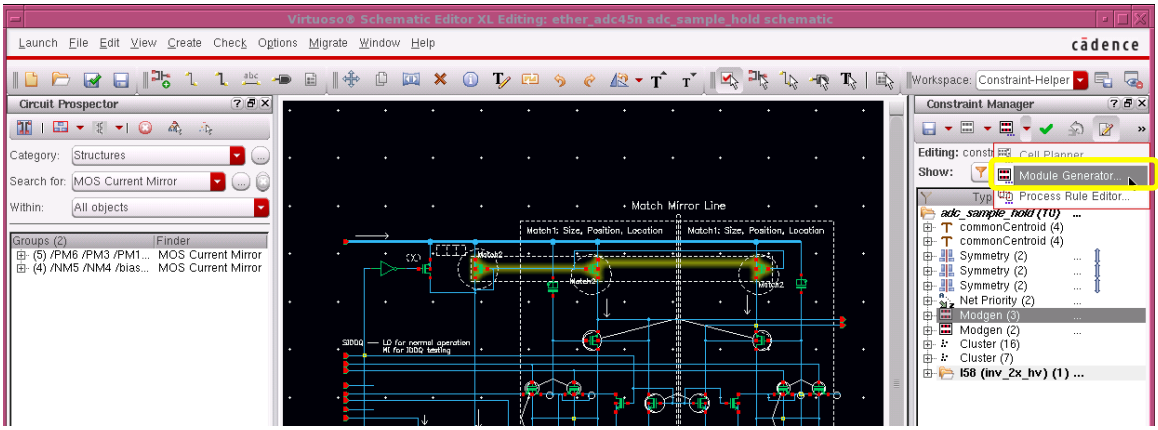
Action 16: In the Circuit Prospector, select 'MOS Current Mirror' from the 'Search for' pull-down under 'Structures' category. This returns the two current mirrors in the design that need to be precisely matched, as indicated in the Schematic annotation.



Action 17: We can generate the current mirrors as matched arrays in the Layout by applying a ‘Modgen’ constraint to each one of them. Click on the ‘Constraint Creation’ pull-down in the Constraint Manager Toolbar and choose ‘Modgen’ constraint from the ‘Placement’ category, as shown below, to create a ‘Modgen’ constraint on the two groups.

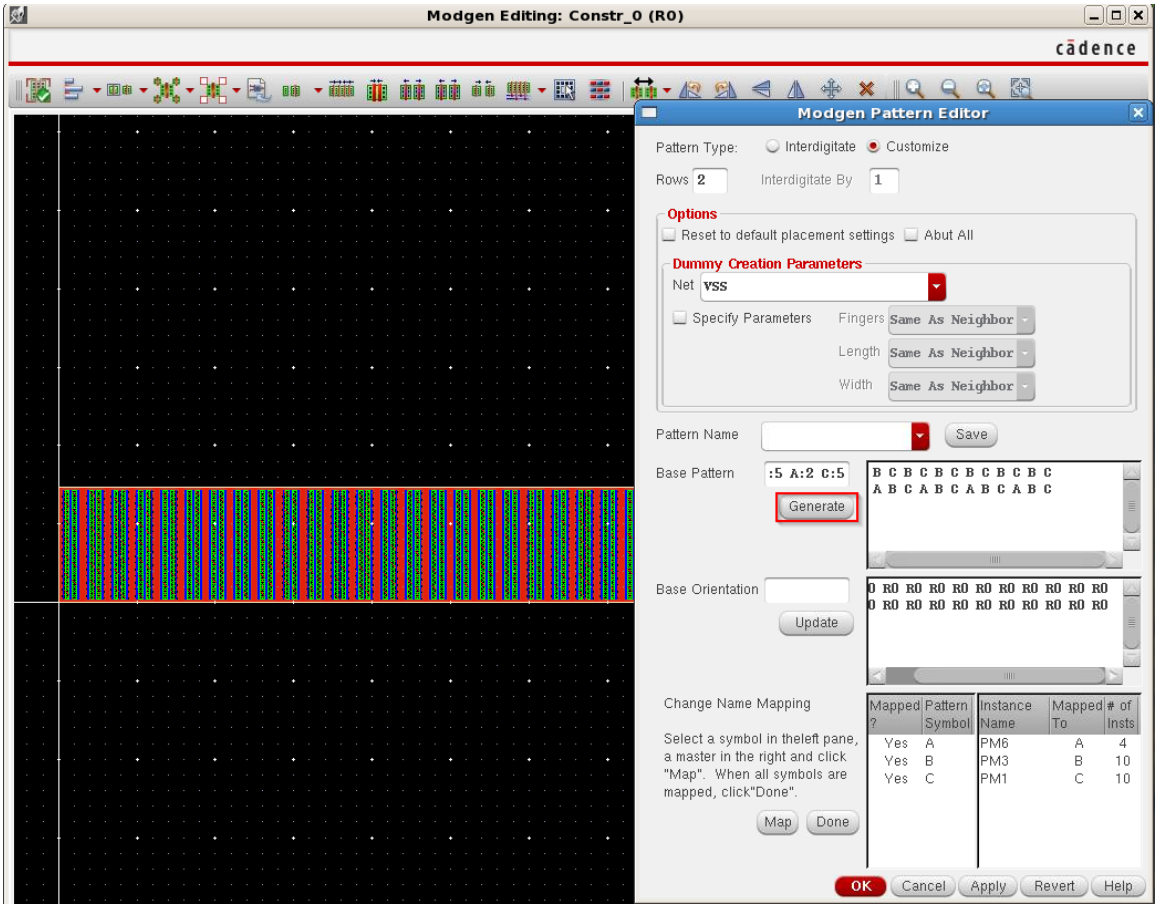


Action 18: In the Constraint Manager, select the Modgen constraint on the PMOS Current Mirror and invoke the Modgen Editor by clicking on ‘Module Generator’ on the Constraint Editors pull-down (3rd icon from the left in the Constraint Manager Toolbar).



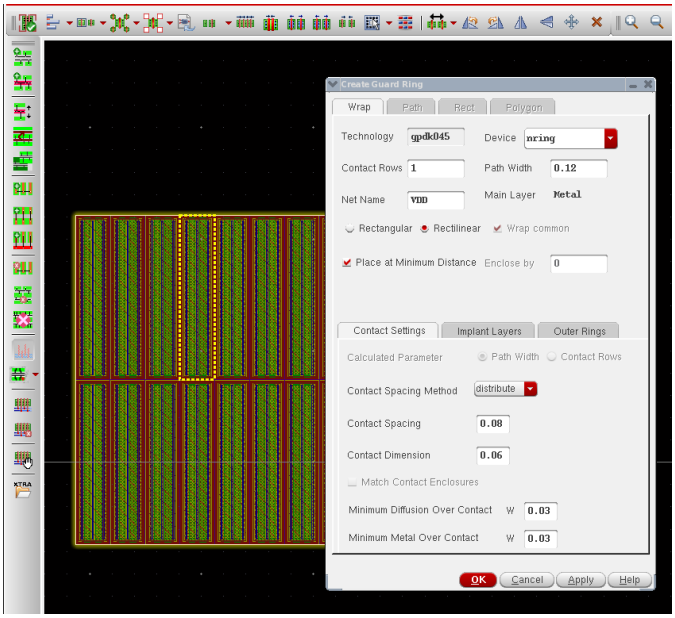
Action 19: In the Modgen Editor window that comes up, you will see the layout view of the m-factored instances PM6, PM3 and PM1. You can modify how these instances are placed in an interdigitated manner.

- Click on the (Pattern) icon in the Toolbar which brings up the pattern form. Note that symbols have been used to denote the instances, to make the pattern entry easier.
- Click on the 'Customize' radio button under 'Pattern Type' and change 'Rows' to 2.
- Click on the 'Base Pattern' field and enter the pattern B:5 A:2 C:5 so that 5 instances of B (PM3) are followed by 2 instances of A (PM6) and 5 instances of C (PM1), repeating for the remaining instances thereafter.
- Click on the 'Generate' button for the base pattern to populate the pattern field. (You can enter a name for the pattern under 'Pattern Name' and click on the 'Save' button next to it so you can restore the pattern later on, if needed.)

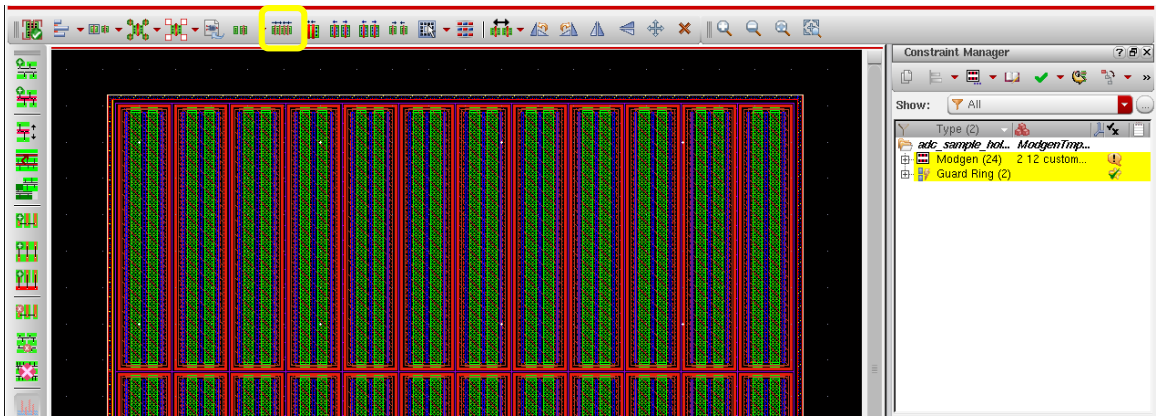


Action 20: Click OK in the pattern form which will update the Modgen as per the specified pattern. Additionally, you can see the instances are placed DRC correct and the well is merged (well merging is controlled by an environment variable that is set to true in this Workshop database).

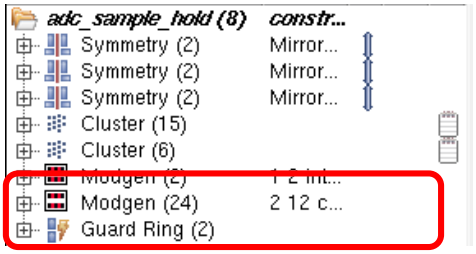
Action 21: Now click on the (Guardring) icon in the Modgen Toolbar. Choose the Device to be ‘nring’ and the Net Name to be ‘VDD’. Ensure that ‘Place at Minimum Distance is set to true to ensure that minDRC values are used. Click ‘OK’ in the form.



Action 22: The Modgen Editor lets you abut all or selected instances. Click on the (Abut All) icon in the Modgen toolbar to merge the devices.



Action 23: Click on the (Exit) icon to quit the Modgen Editor. Notice that the Modgen constraint in the Constraint Manager has updated to reflect the changes made inside the Modgen Editor. It now shows the actual number of physical instances due to expansion of m-factor, and there is also a separate Guardring constraint on the Modgen.

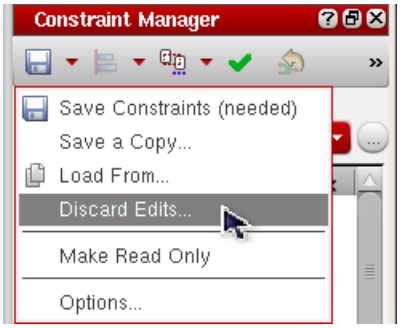


4 Automatic Constraint Generation in Schematic

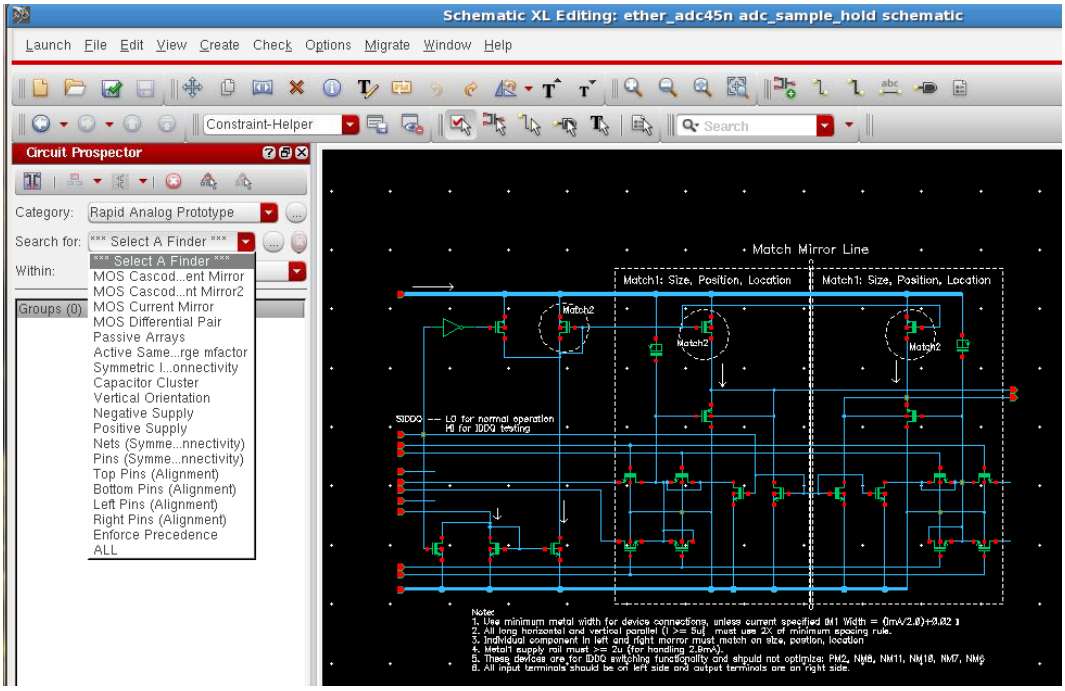
In this section...

*We will see how a set of constraints***including Modgens* ,** required to drive the Layout for a given design, can be created automatically using built-in finders under the 'Rapid Analog Prototyping' category in the Circuit Prospector.*

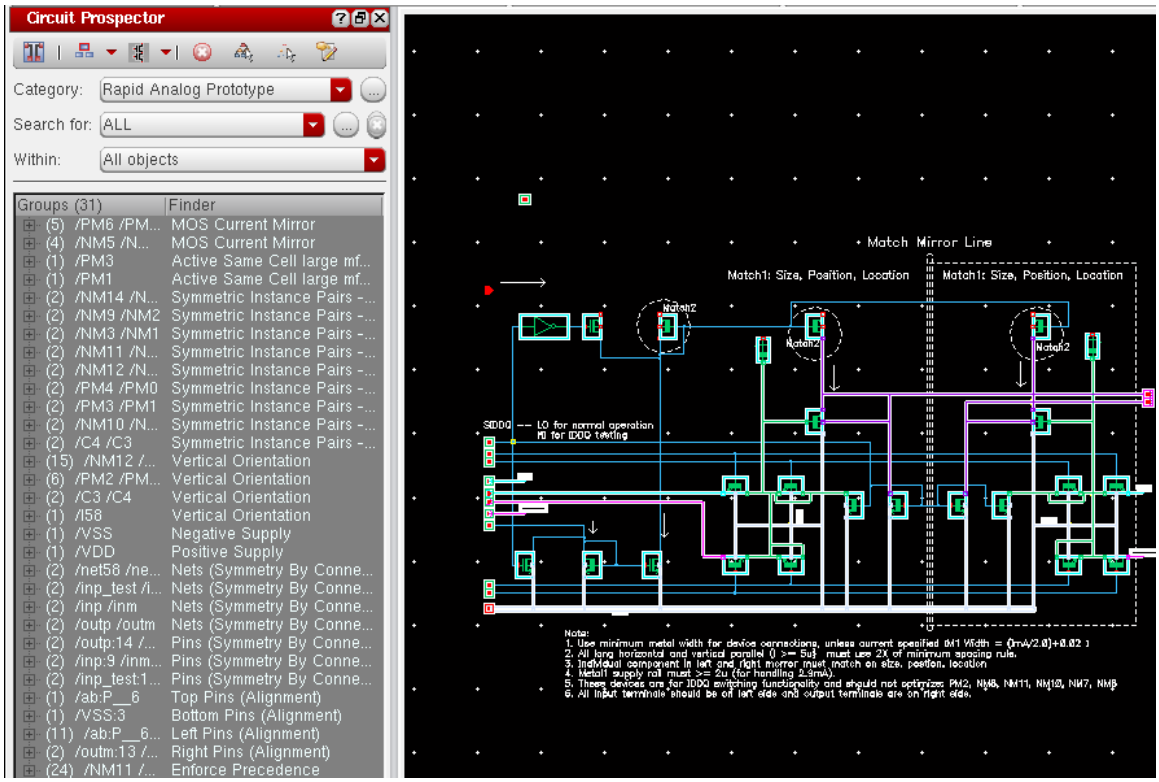
Action 1: Click on 'Discard Edits' under the File pull-down in the Constraint Manager Toolbar, and click 'Yes' in the Discard Edits form. This clears the constraints created in the last chapter.



Action 2: In the Circuit Prospector, choose the 'Rapid Analog Prototyping' category. Click on the 'Search for' pull-down to see all the available finders under this category.



Action 3: From the finders pull-down, choose ‘All’ to run all the finders on this schematic. The Circuit Prospector returns all groups of devices/ nets identified by these finders e.g. Current Mirrors, Active Devices with Large Multipliers, Supply Nets etc.

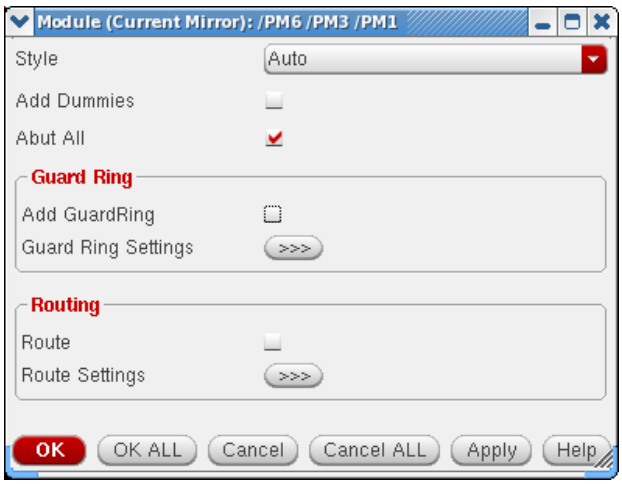


Action 4: Each finder in the ‘Rapid Analog Prototyping’ category in the Circuit Prospector has a corresponding generator to create the relevant constraint for that structure or group of devices/ nets. Click on the ‘Default Action’ button in the Circuit Prospector menu (highlighted below) to run these generators to create necessary constraints automatically.

Note: User can modify the default set of finders in the ‘Rapid Analog Prototype’ category to add/ remove finders, customize the generators, or create new Modgen generators. This is outside the scope of this workshop but is detailed in presentations/ training slides that are available in COS.

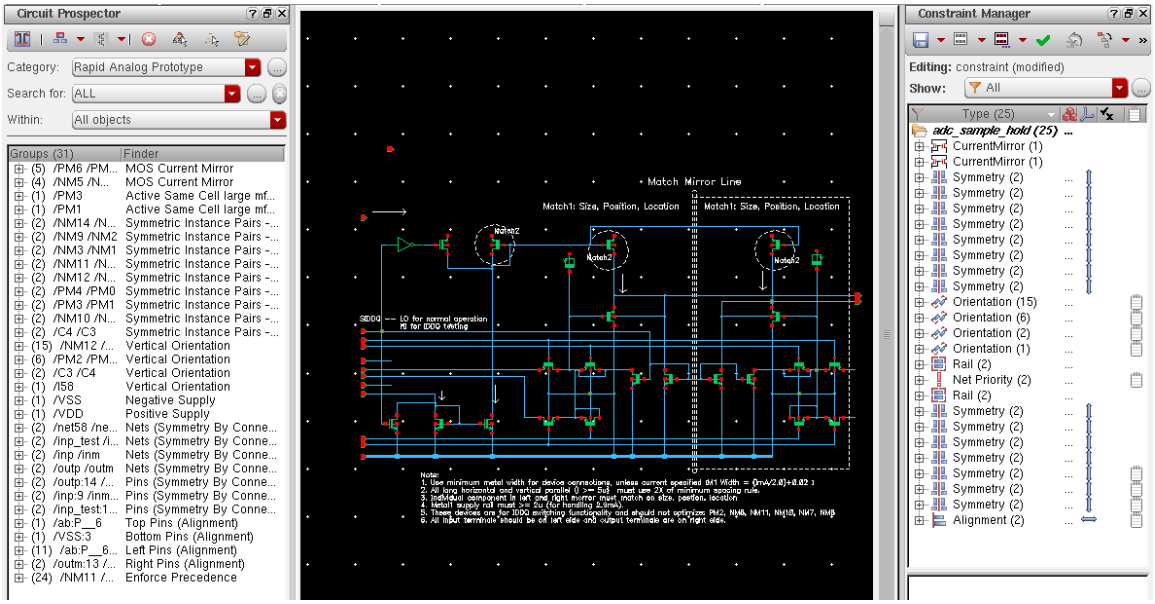
Action 5: Notice that a form comes up for each structure that has a Modgen constraint associated with it (as well as for some other constraints). This is to let users change the default settings for the Modgen being created, e.g. whether or not the devices in the Modgen should be abutted, whether or not a Guard Ring or Dummy Devices should be

added, and if the Modgen should be routed. Some of the settings (such as Guard Ring and Routing) can be expanded further for more control.



Action 6: Click 'OK All' in this form. This suppresses all subsequent forms, and the constraints are created with the default settings as seen in the Constraint Manager.

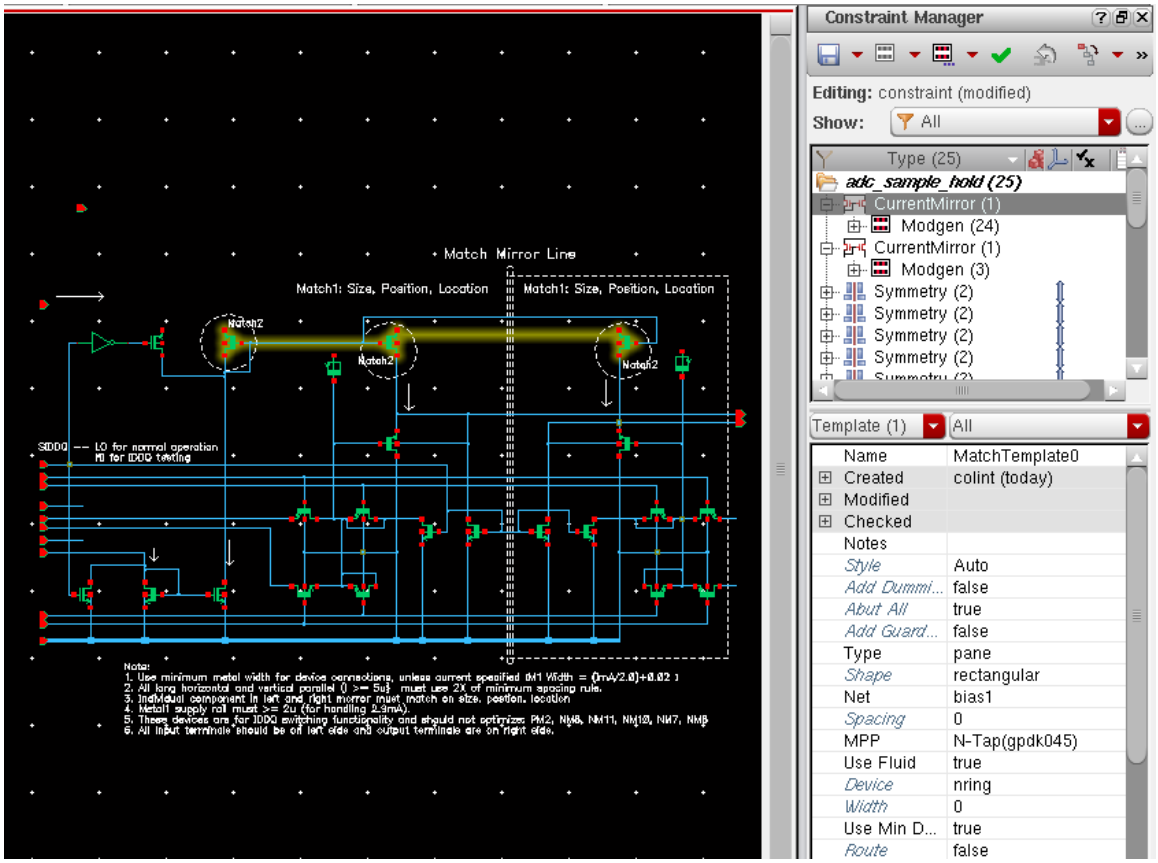
Note: The settings in these pop-up forms for generating Modgens can be seeded up-front using SKILL APIs, and they can be changed after the constraints have been created (we will see how in a subsequent step).



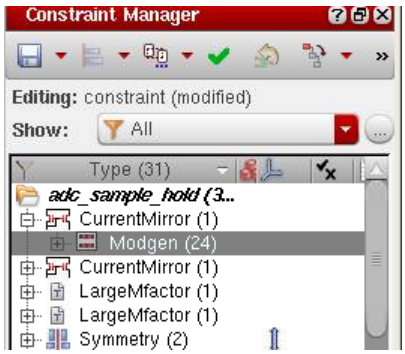
Action 7: Note that Modgen constraints (displayed as Templates) were created for the Current Mirrors, along with symmetry constraints for symmetric devices and nets, orientation constraints for all devices, and rail, alignment and symmetry constraints for

pins. Also notice that the constraints were applied with precedence e.g. symmetry on Modgens rather than on individual devices in the Modgen, by the ‘Enforce Precedence’ finder/ generator.

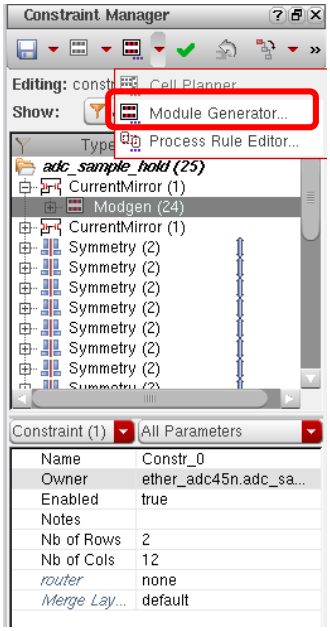
Action 8: Select the ‘Current Mirror’ Modgen Template in the Constraint Manager for the top current mirror. Notice the parameters section in the Constraint Manager displays parameters corresponding to the fields in the pop-up form when the Modgen was created. The parameters can be modified at any time to change the Modgen configuration.



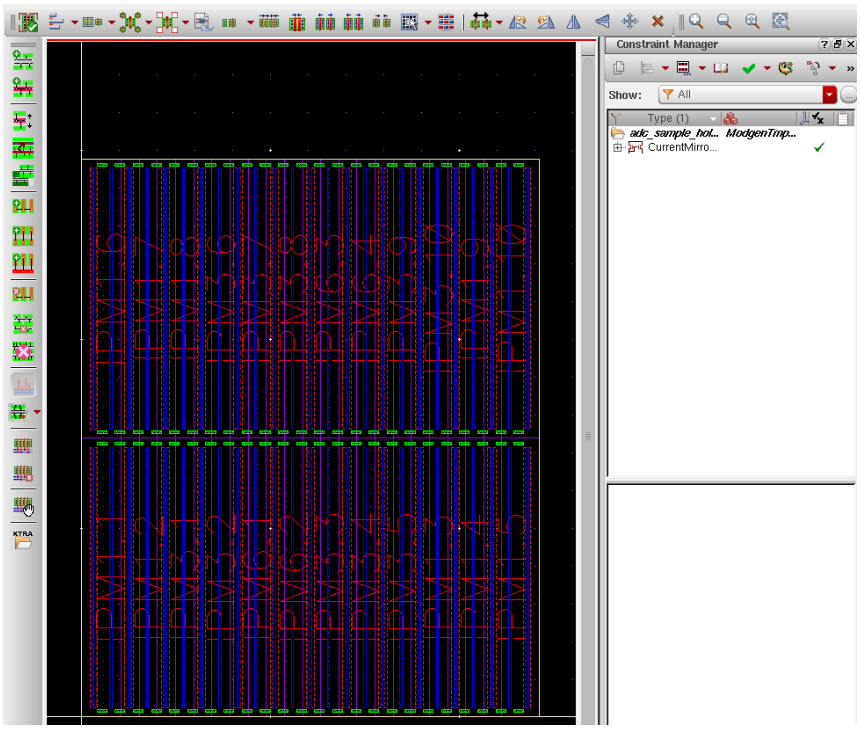
Action 9: Click on the plus button next to the Template to display its members. You can see the Modgen constraint underneath.



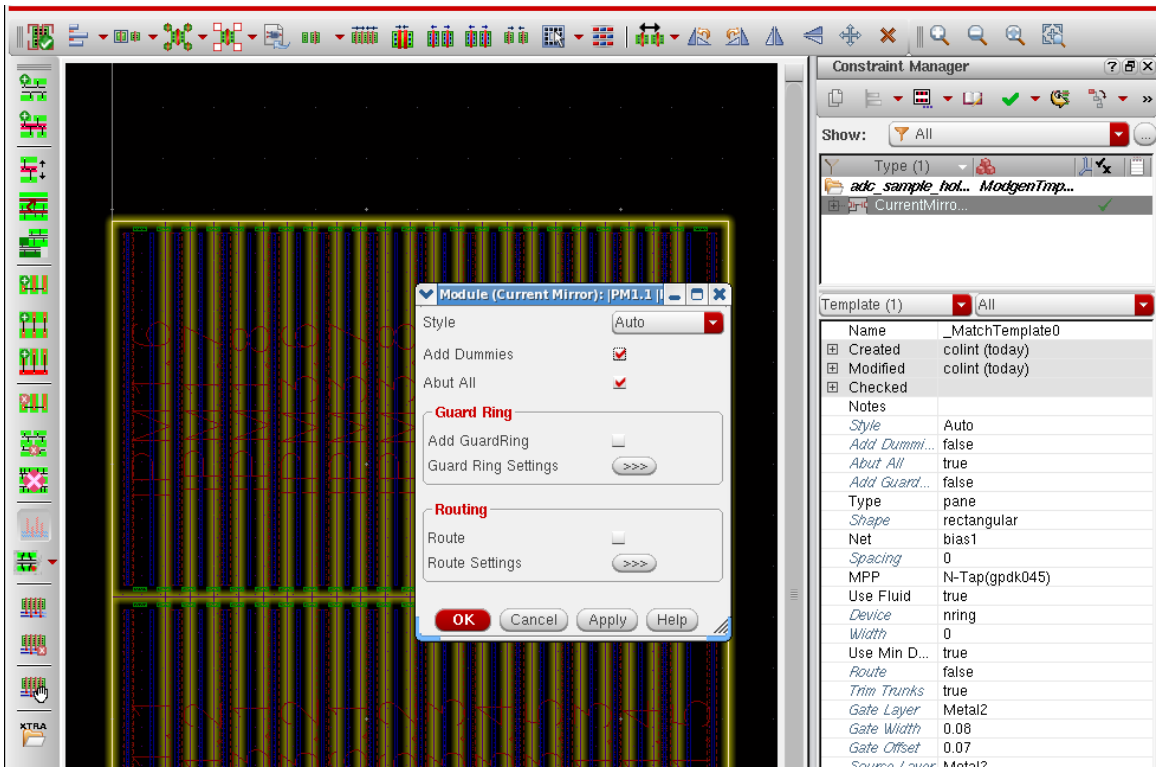
Action 10: Select the Modgen constraint and invoke the Modgen Editor from the Constraint Manager Editors pull-down.



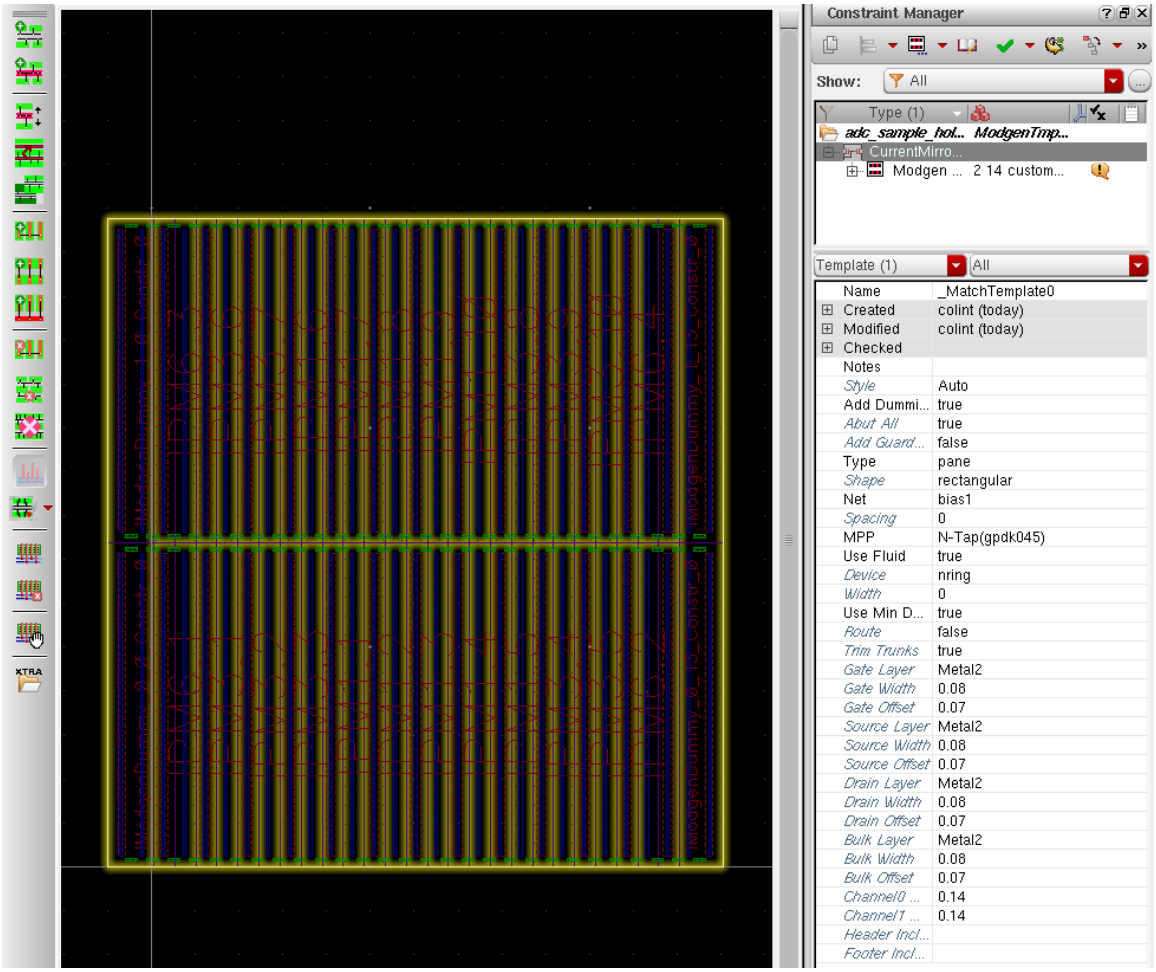
Action 11: In the Modgen Editor window, you can see how the Modgen has been generated as per the default settings, with the devices placed to meet the matching requirement for the current mirror. You can turn the display off (set display level to 0) to see the member instance names using the Shift-F bindkey.



Action 12: In the Constraint Manager inside the Modgen Editor window, select the Current Mirror template that describes the Modgen constraint. In the bottom pane of the Constraint Manager, click the ‘Add Dummies’ field. This opens the Module Template form. In this form set ‘Add Dummies’ option to ‘true’ (change from the default value of false) and click OK.



Action 13: You will see that the template is modified and the Modgen is regenerated with dummies on either side. The Modgen template parameters can be modified at any time to regenerate the Modgen, and this can be done in the Constraint Manager inside or outside the Modgen Editor (in the Schematic Constraint Manager window).



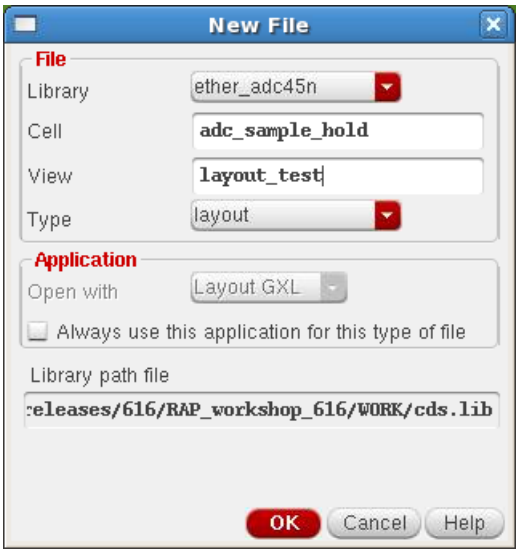
Action 14: Quit the Modgen Editor (click on the 'Exit' button) and discard edits in the Constraint Manager (the constraints auto-generated using the RAP finders have been saved in constraint view 'constraint1' which we will use to drive the layout in the next section).

5 Automatic Constraint Driven Placement & Routing

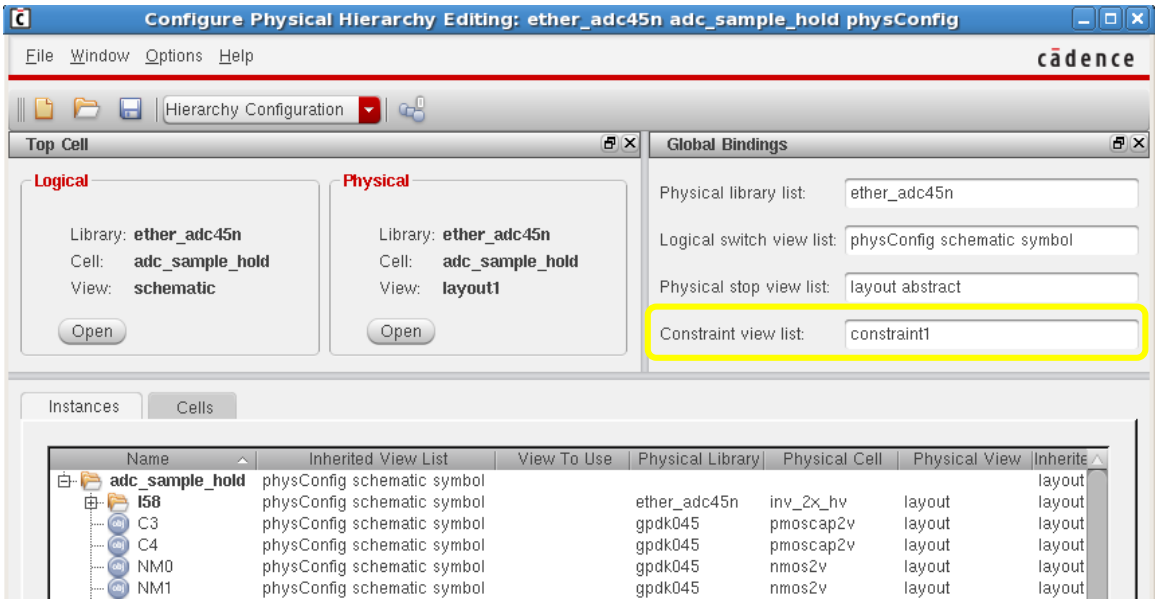
In this section…

We will see how constraints are passed automatically to the Layout from a specified constraint view or brought over from another Layout view. We will see the constraints are implemented in Layout by automatic tools as well as during interactive editing.

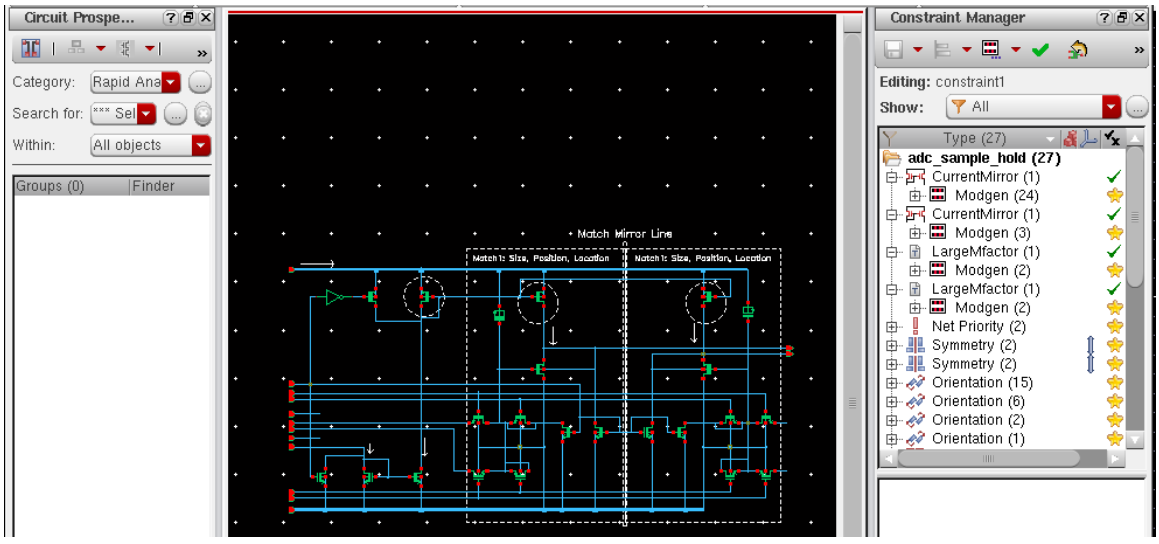
Action 1: From the Schematic window, click on 'Launch > Layout-GXL'. In the form that comes up, choose to create a new layout view and in the subsequent form, enter the layout view name (say 'layout_test') as shown below and click 'OK'.



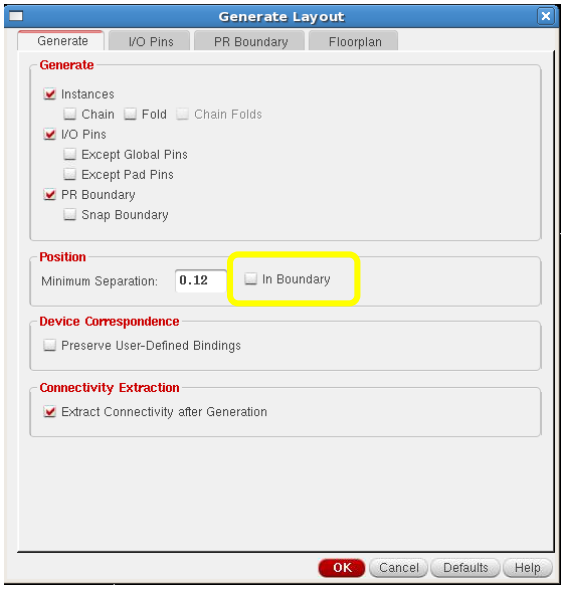
Action 2: Invoke the CPH utility ('Launch > Configure Physical Hierarchy' from the Layout GXL pull-down menu) to make sure that the right constraint view (in this case, 'constraint1') is being used to drive the Layout. After changing the view name, if needed, click on the 'Save' button in the CPH and close it.



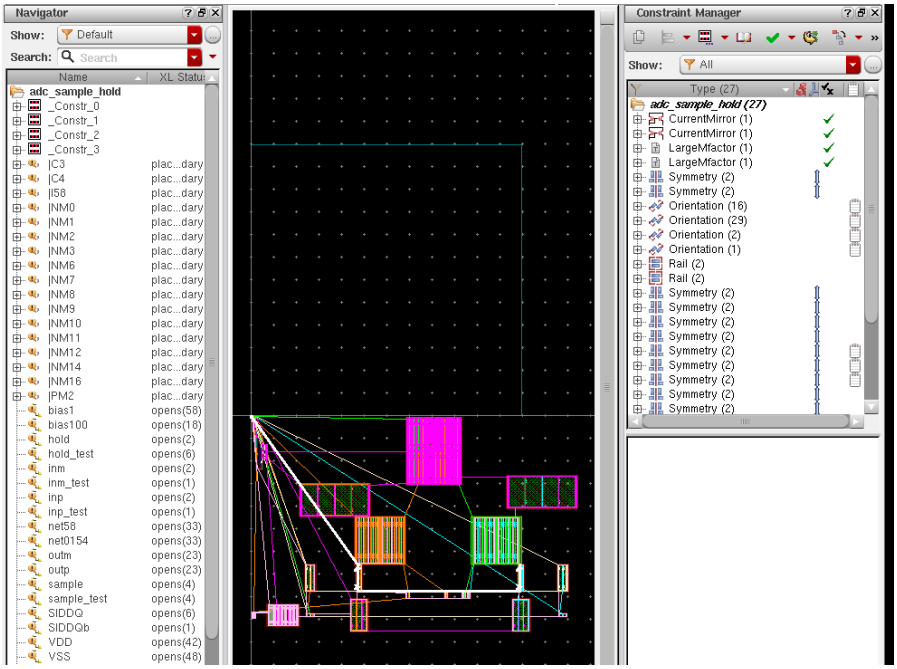
Action 3: Look at the Constraint Manager in the Schematic. It is now populated with constraints as specified in the view 'constraint1'. The status column indicates that these constraints are not yet in the Layout, and the Constraint Manager in the Layout does not display any constraint at this time. (To see the status of the Modgen constraints created under templates, you will need to expand the templates).



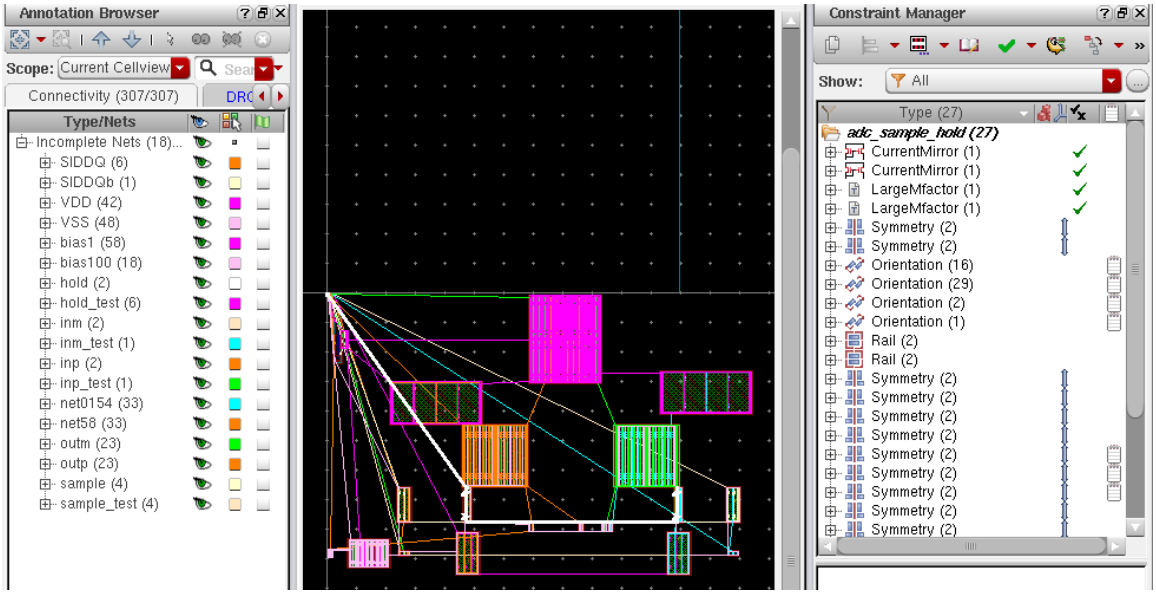
Action 4: From the Layout-XL menu (or using icon in the Layout XL Toolbar), invoke ‘Connectivity > Generate > All from Source’. Make sure you uncheck the ‘In Boundary’ option (so instances and pins will get placed outside the prBoundary) and click ‘OK’ in the form to generate the components and nets in the Layout.



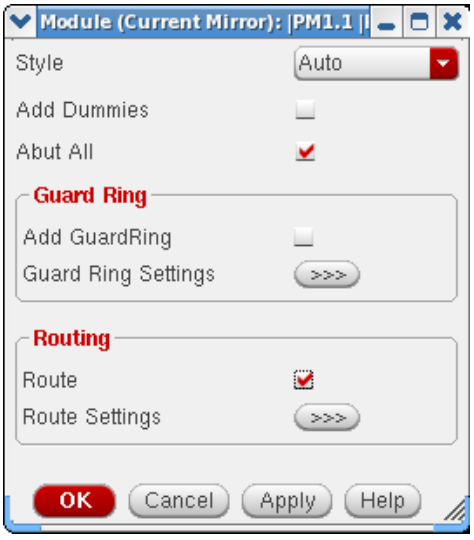
Action 5: See that instances and pins (and associated nets as indicated by the flight lines) are generated in the Layout canvas, and the Constraint Manager in the Layout also gets populated with constraints from the Schematic. No constraint (other than Modgens) is enforced at this time, and they are not checked either. For that, we will need to run the placement and routing tools (interactive or automatic).



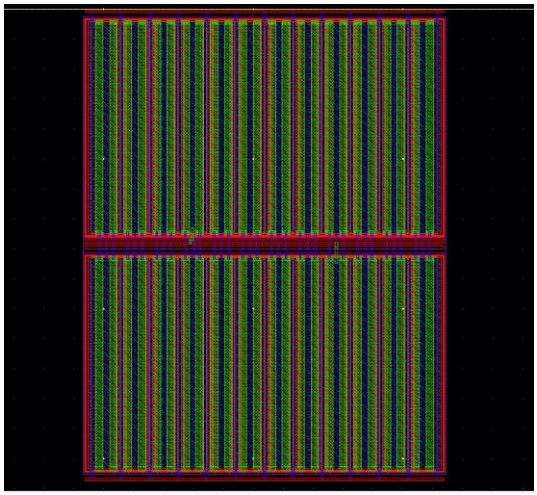
Action 6: Click on the Annotation Browser tab on the left hand side of the canvas next the Navigator to see all the incomplete nets (opens) in the Layout. Click to toggle the ‘eye’ icon next to the ‘Incomplete Nets’ to turn off the display of flight lines so we can better view the instances and Modgens.



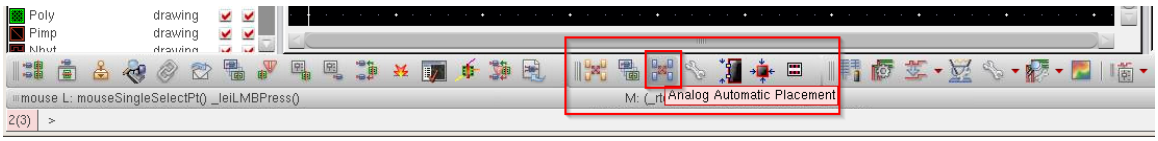
Action 7: Before running the placer and the router, we will go ahead and route the Modgen for the top Current Mirror as the routing requirement for it is more complex. Click on the ‘Current Mirror’ template in the Constraint Manager which highlights the Modgen in the canvas. In the bottom pane of the Constraint Manager, click on one of the fields on the right to open the Module template form. In the form check the ‘Route’ parameter to set it to true. Click OK on the form.



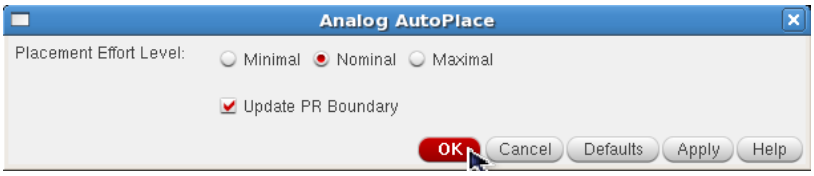
Action 8: Zoom into the Modgen to see that it has been routed.



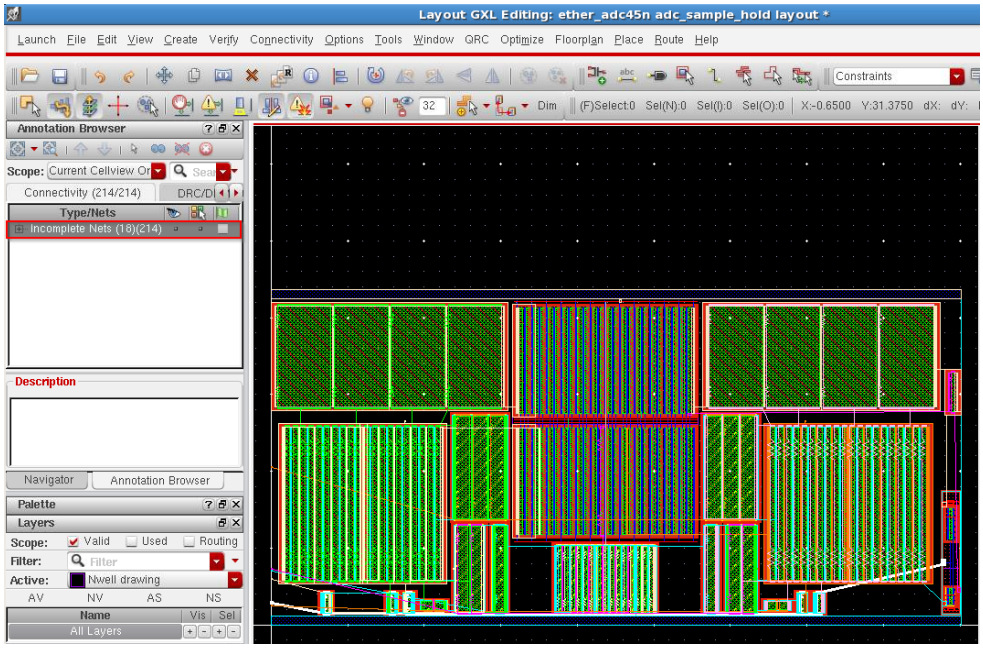
Action 9: Click on the (Analog Auto Placer) icon in the Analog Auto Placer Toolbar in the bottom of the Layout window, or from the menu (‘Place > Analog > Automatic Placement …’).



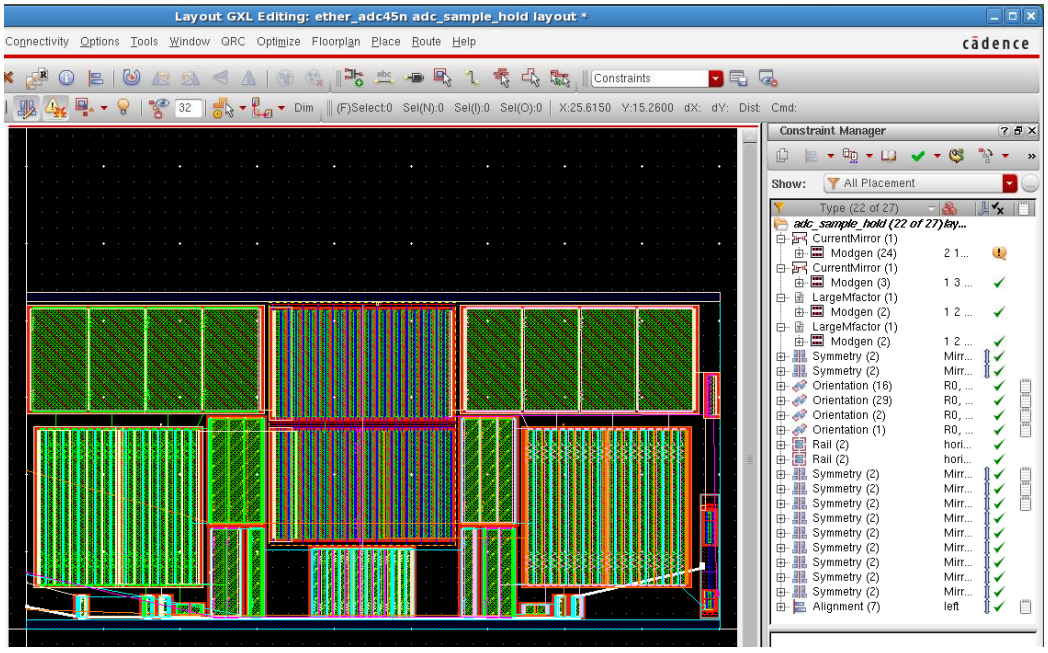
Action 10: Run the Analog Placer by clicking ‘OK’ in the form, with the default ‘Nominal’ effort and the option to update the boundary size.



Action 11: Once the placement is complete, notice the rails are generated automatically, and instances and pins are placed following connectivity and constraints. Click on the 'Incomplete Nets' in the Annotation Browser to see that the placement is optimal based on connectivity.

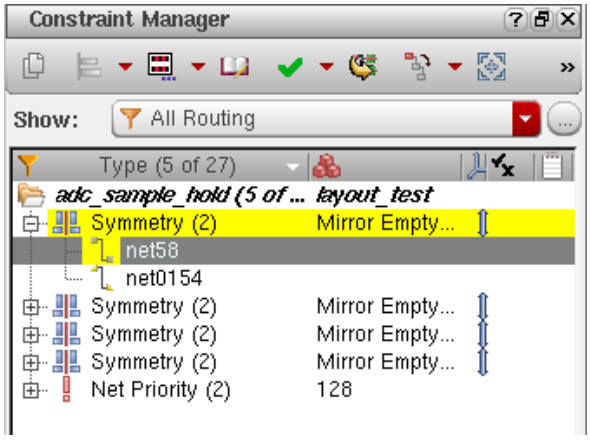


Action 12: Apply the 'All Placement' filter in the Layout Constraint Manager to display all the placement constraints, and see they have all been met by the Analog Placer, as indicated by the green checkmarks in the status column. To see the status of the Modgen constraints, expand the templates (notice that the top Current Mirror Modgen shows up as differing from the Schematic constraint because it has been routed in the Layout).



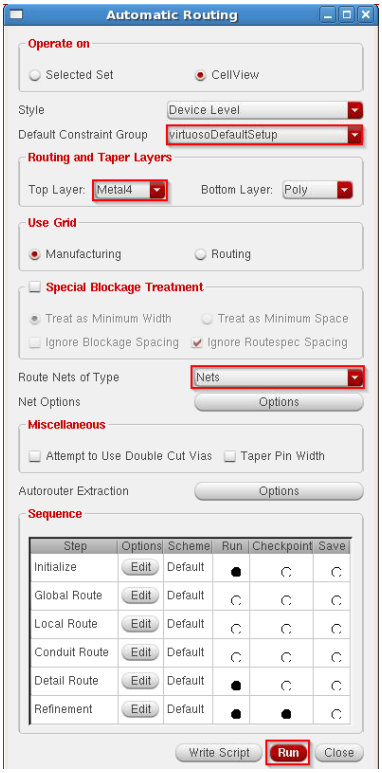
Note: You can interactively modify the placement at this point without violating the constraints (in CAE mode), and fix the placement incrementally using the 'Fix DRC' command of the Analog Placer.

Action 13: We will now use the automatic routing functionality to route the nets in the Layout. Click on the 'All Routing' filter in the Constraint Manager to view the routing constraints in the design. You can expand the constraints to view the member nets.

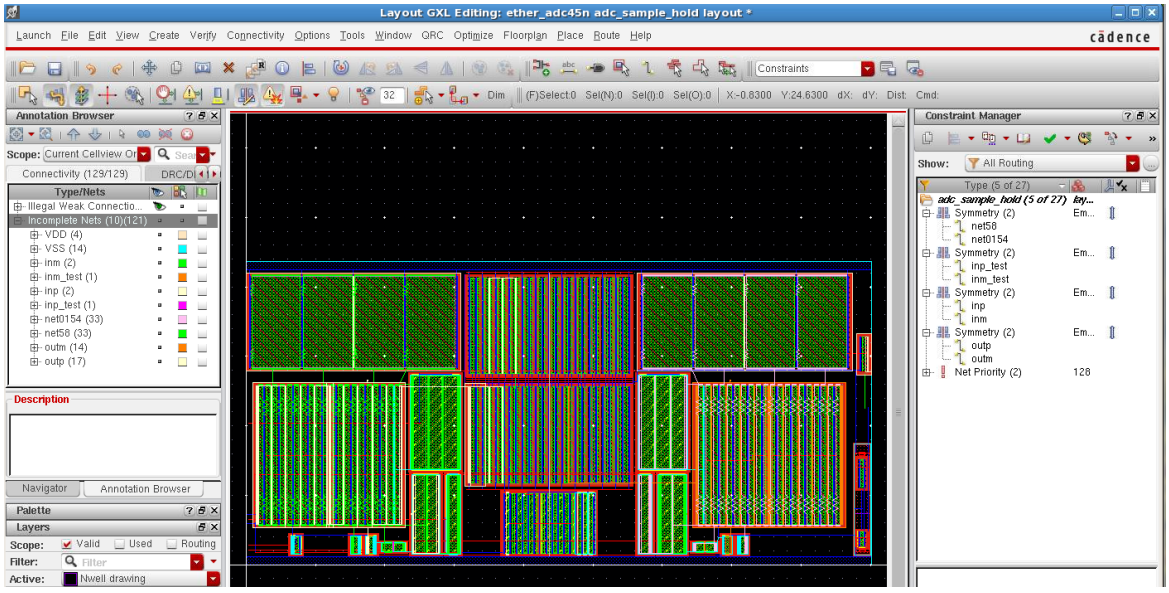


Action 8: We will now use the automatic routing functionality once again to route the nets in the Layout. Click on 'Automatic Routing' icon in the 'Space Based Router' toolbar at the bottom of the Layout window or the 'Automatic Routing' command from the 'Route' Menu in the Layout window.

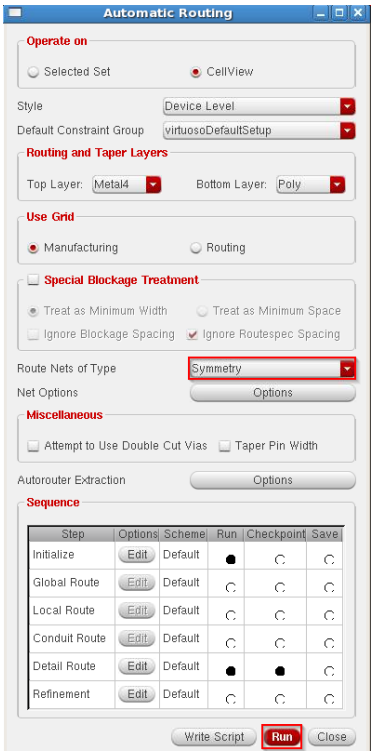
Action 15: In the form, specify the Default Constraint Group as ‘virtuosoDefaultSetup’, and enter the top and bottom layers as Metal4 and Poly respectively. Click on the ‘Run’ button the run the automatic router through the default steps as shown.



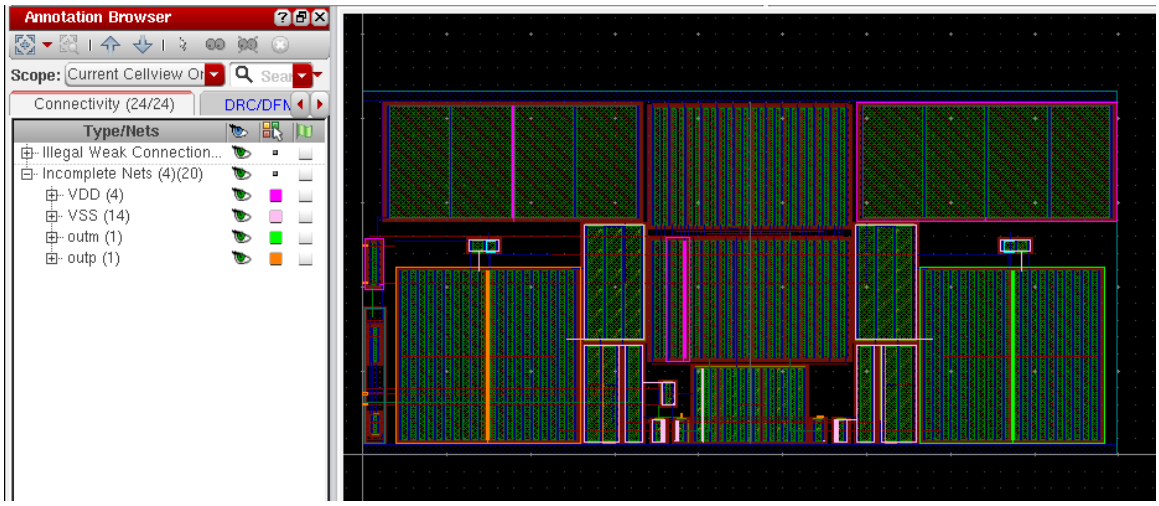
Action 16: When the routing completes, click on the Annotation Browser Assistant and expand the ‘Incomplete Nets’. Notice that all nets (except for VDD and VSS) that show up as opens are those that have specialty routing constraint (in this case, symmetry) on them. We will now route the symmetric nets.



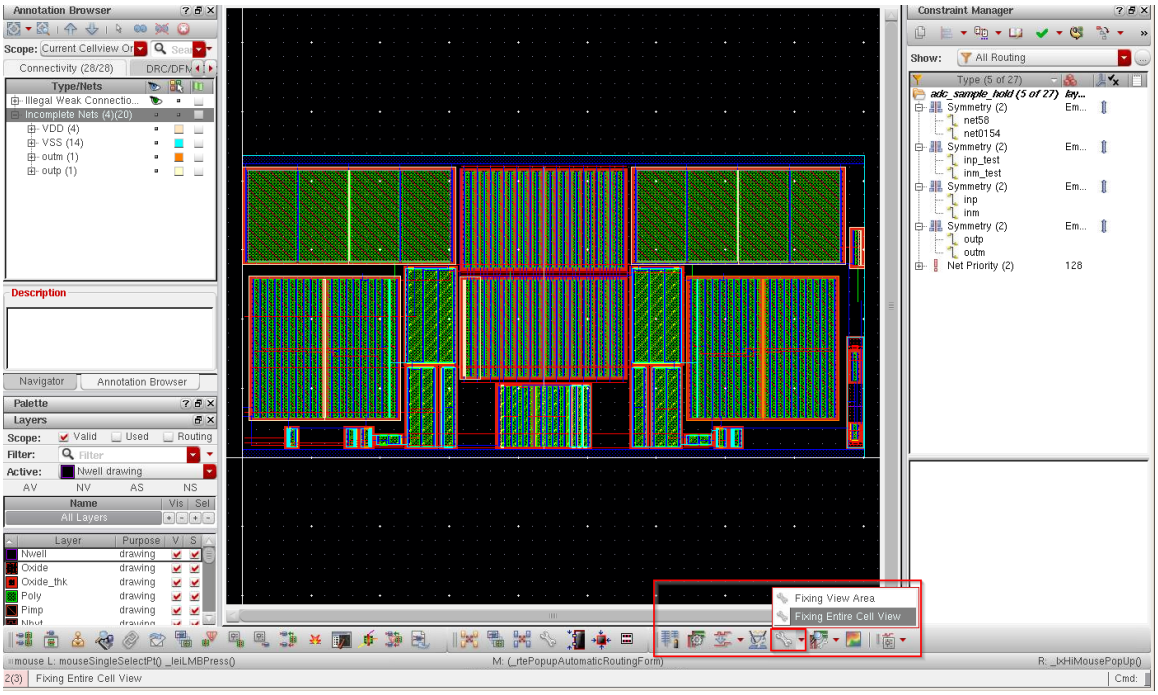
Action 17: In the ‘Automatic Routing’ form that is still up, click on the cyclic pull-down under the field ‘Route Nets of Type’ and select ‘Symmetry’ as shown in the screenshot below. Click once again on the ‘Run’ button.



Action 18: You should now see that the nets with symmetry constraints have been routed as well and the only opens left in the design, as displayed in the Annotation Browser, are nets whose wells have not been merged and taps inserted. You may also see some illegal weak connect violations in the Annotation Browser which we will clean up in the next step. Go ahead and close the ‘Automatic Routing’ form.

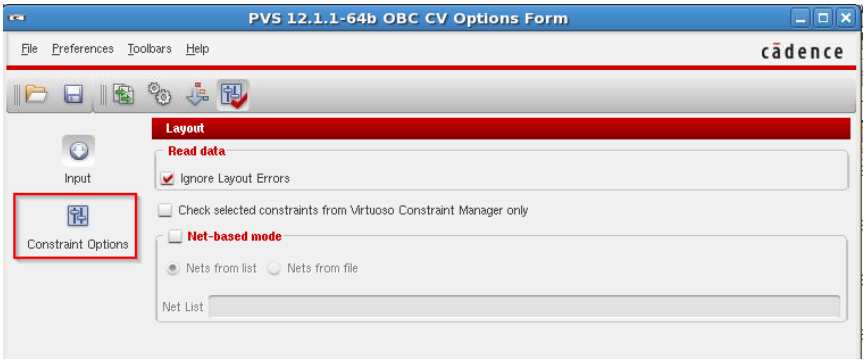


Action 19: Click on the ‘Fix Violations’ icon in the Space Based Router toolbar in the bottom of the Layout window, and select ‘Fix Entire Cell View’ from the cyclic field. Once this operation is completed, you will see that the illegal weak connection violations have been fixed and no longer appear in the Annotation Browser.

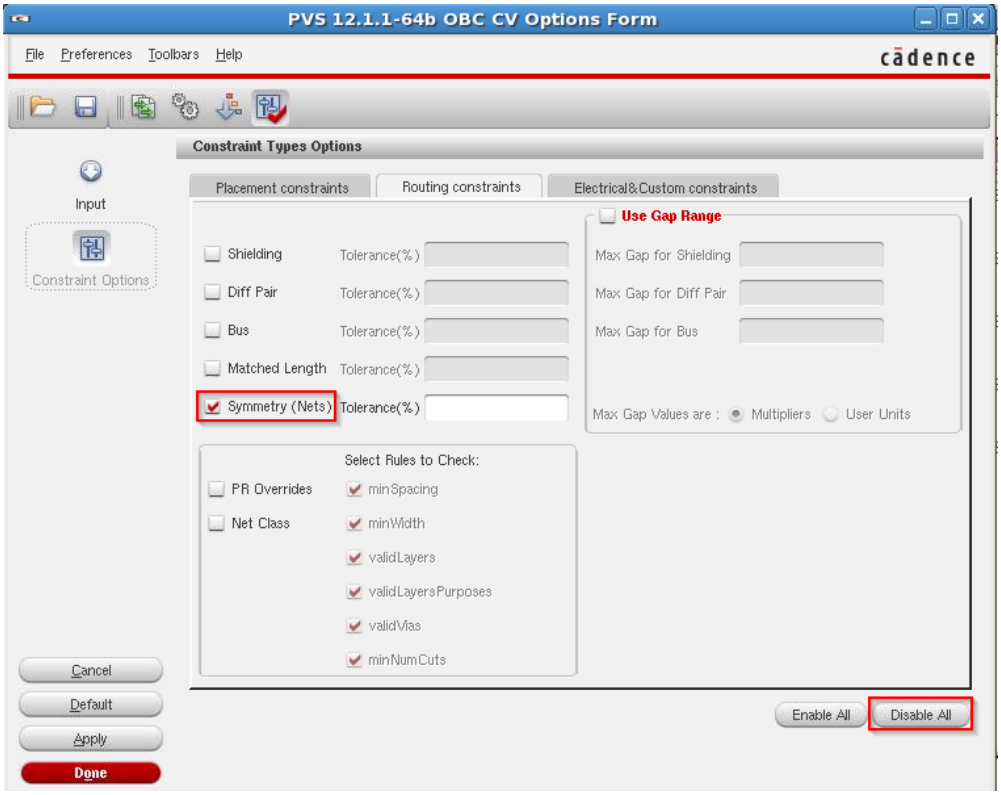


Action 20: From the (Check Constraints) pulldown in the Constraint Manager Toolbar, click on ‘Options’. This will bring up the Options form for the PVS based constraint checker.

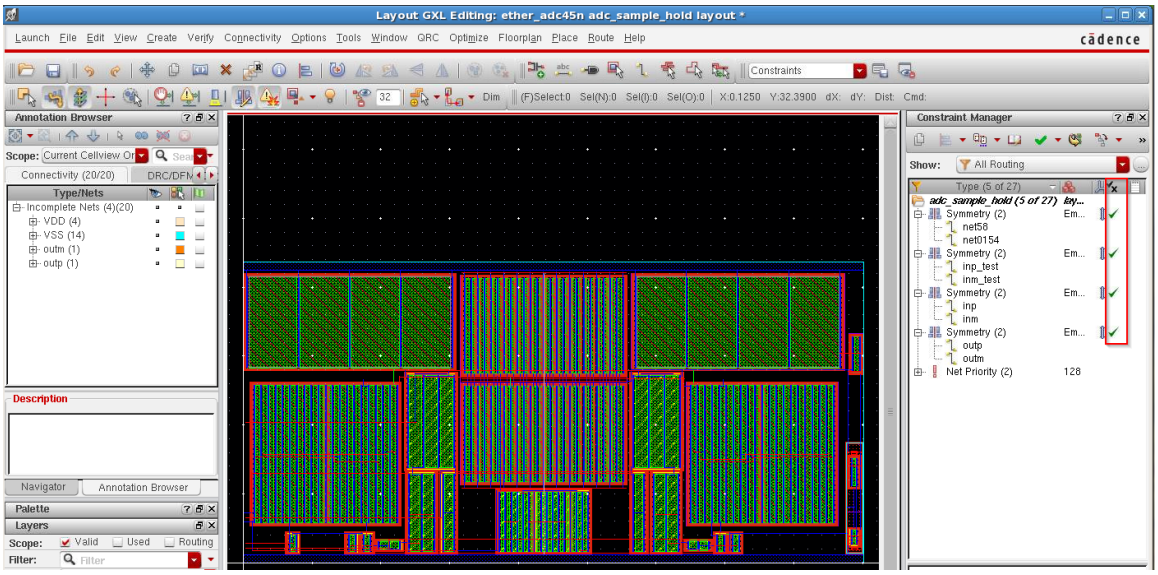
Action 21: Click on the ‘Constraint Options’ button in the form that comes up.



Action 22: Click on the ‘Routing Constraints’ tab. Click on ‘Disable All’ and select Symmetry (Nets) as that is the only constraint we want to check at this point. Click on ‘Done’ to apply the settings.



Action 23: In the Constraint Manager, click again on the ‘Check’ button to run the Constraint checker with the above settings. You can see messages in the CIW indicating the start and finish of the check. We see that all net symmetry constraints in the design have been met as indicated by the green check marks in the status column.



Action 24: At this point, we are ready to run LVS and extract the Layout. We just need to draw the well and insert the well and substrate taps. This has been done already in layout view ‘layout1’. Close the current layout without saving.

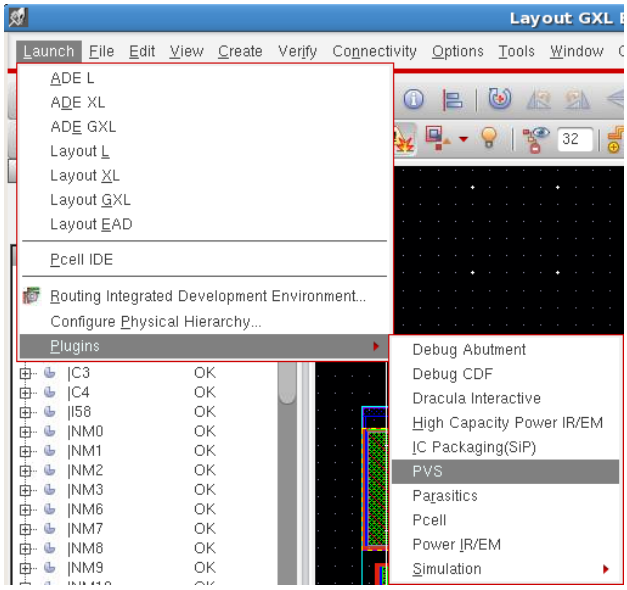
6 Verification, Extraction and Parasitic Re-simulation

In this section...

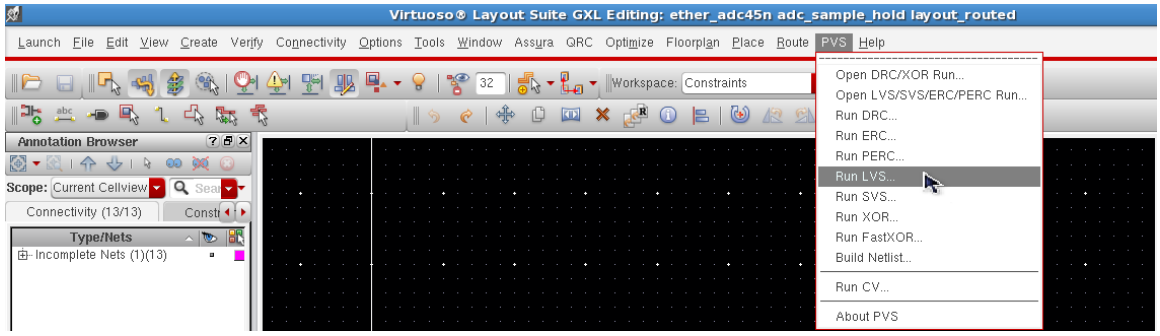
We will run LVS on the layout generated in the previous section and extract the parasitics. We will then rerun the tests in ADE-XL using the extracted view, to check for any design or layout issues.

Action 1: From the Schematic window, click on 'Launch > Layout-GXL' to open layout view 'layout1'.

Action 2: From the Layout window, click on 'Launch > Plugins > PVS'. This will install the PVS menu in the Layout window.



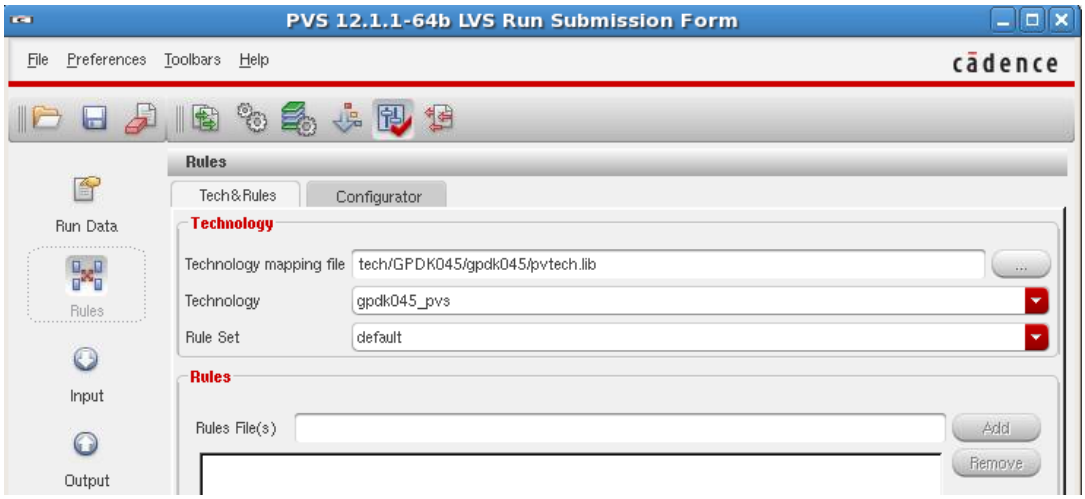
Action 3: From the PVS pull-down menu that appears, click on 'Run LVS …'.



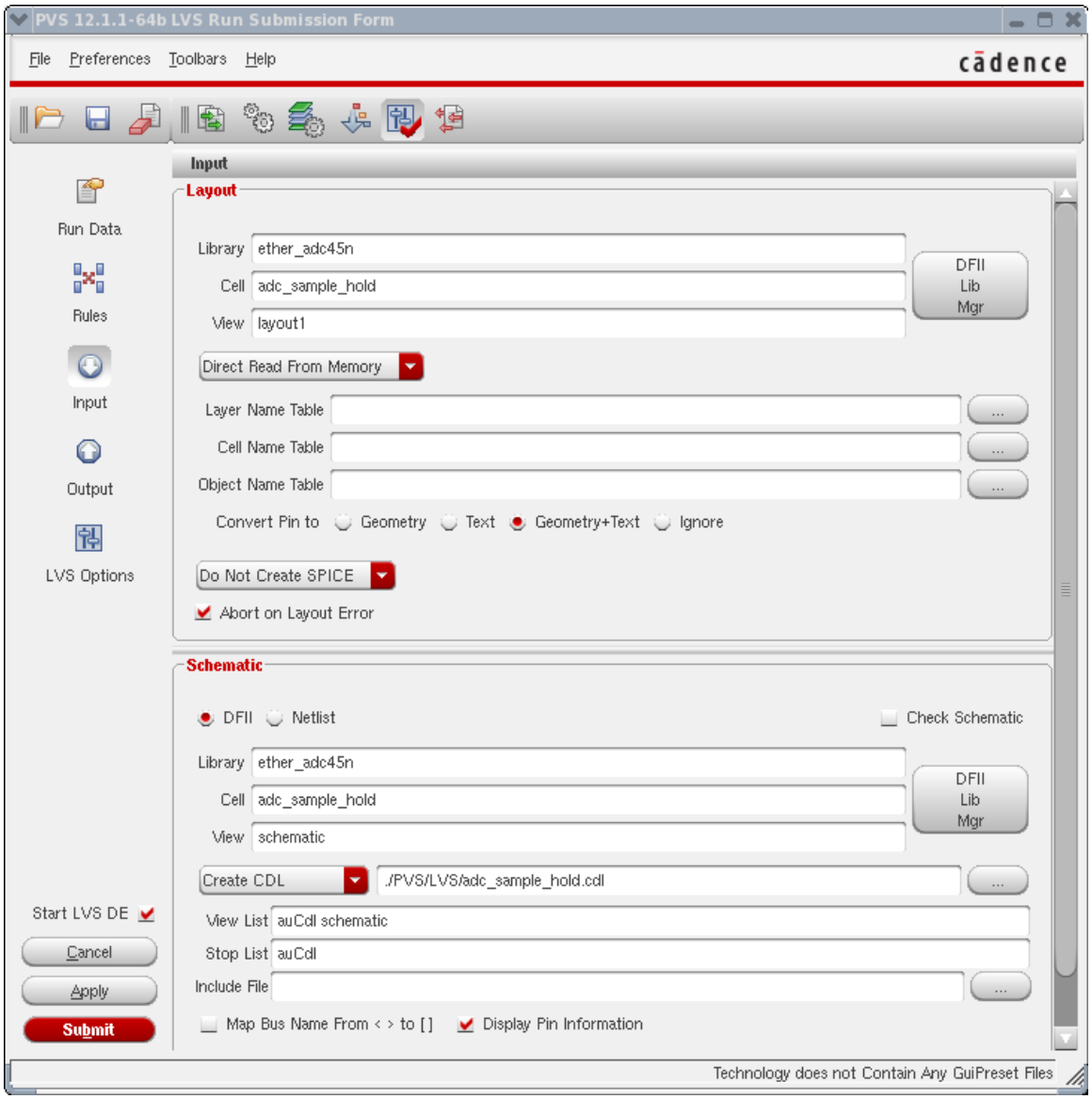
Action 4: In the LVS Run Submission form that comes up, set up the necessary fields as follows.

- Under 'Run Data', click on the browse button to specify the path to the Run Directory (./PVS/LVS/).

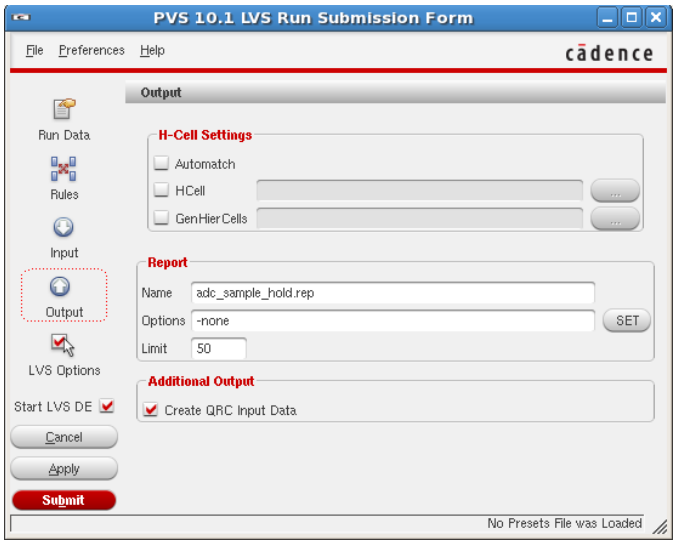
- Under 'Rules', specify the path to the Technology Mapping File for this PDK (./tech/GPDK045/gpdk045/pvtech.lib). Click on the Technology pull-down to select gpdk045_pvs, with the default Rule Set.



- Under ‘Input’, make sure that the Library, Cell and View names are populated correctly for the given Layout. Choose ‘Direct Read From Memory’, ‘Convert Pin to Geometry + Text’, ‘Do Not Create SPICE’ (we will create an extracted view instead) and ‘Create CDL’ for the given Schematic.



- Under ‘Output’, make sure you click on ‘Create QRC Input Data’ as this will be required to run QRC for extraction following the LVS run.



Action 5: Click on the ‘Submit’ button in the LVS Run Submission form to start the LVS run. You may get a pop-up saying that a non-empty directory exists and if it is OK to overwrite. Click ‘Yes’ in that form. You may also get a form that asks to save any unsaved cell view. Click ‘Yes’ also in that form.

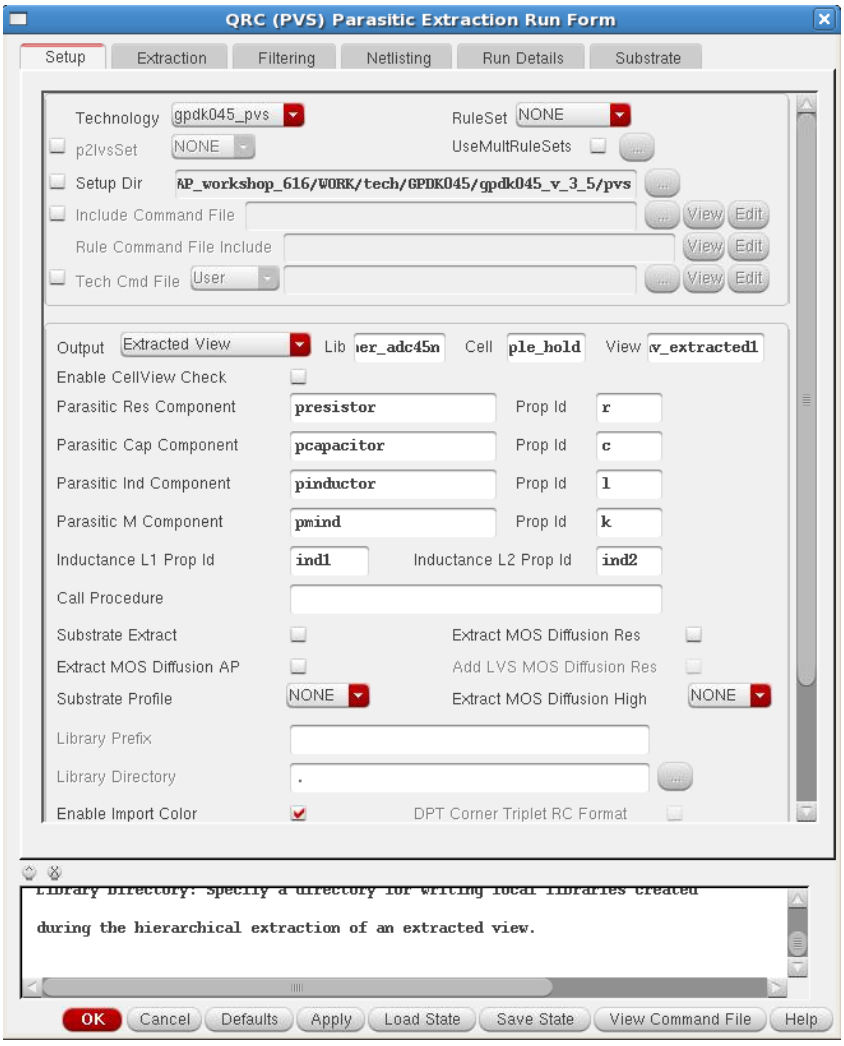
Action 6: This will open up a report window displaying the progress of the LVS run. Once the run is complete, an LVS Run Status form will come up to show that the comparison resulted in a match. Click ‘No’ to the prompt in the status form since there is no mismatch and hence no need to start the LVS debugger. Also close the report window.

Action 7: From the QRC menu pull-down in the Layout window, click on ‘Run (PVS) QRC …’.

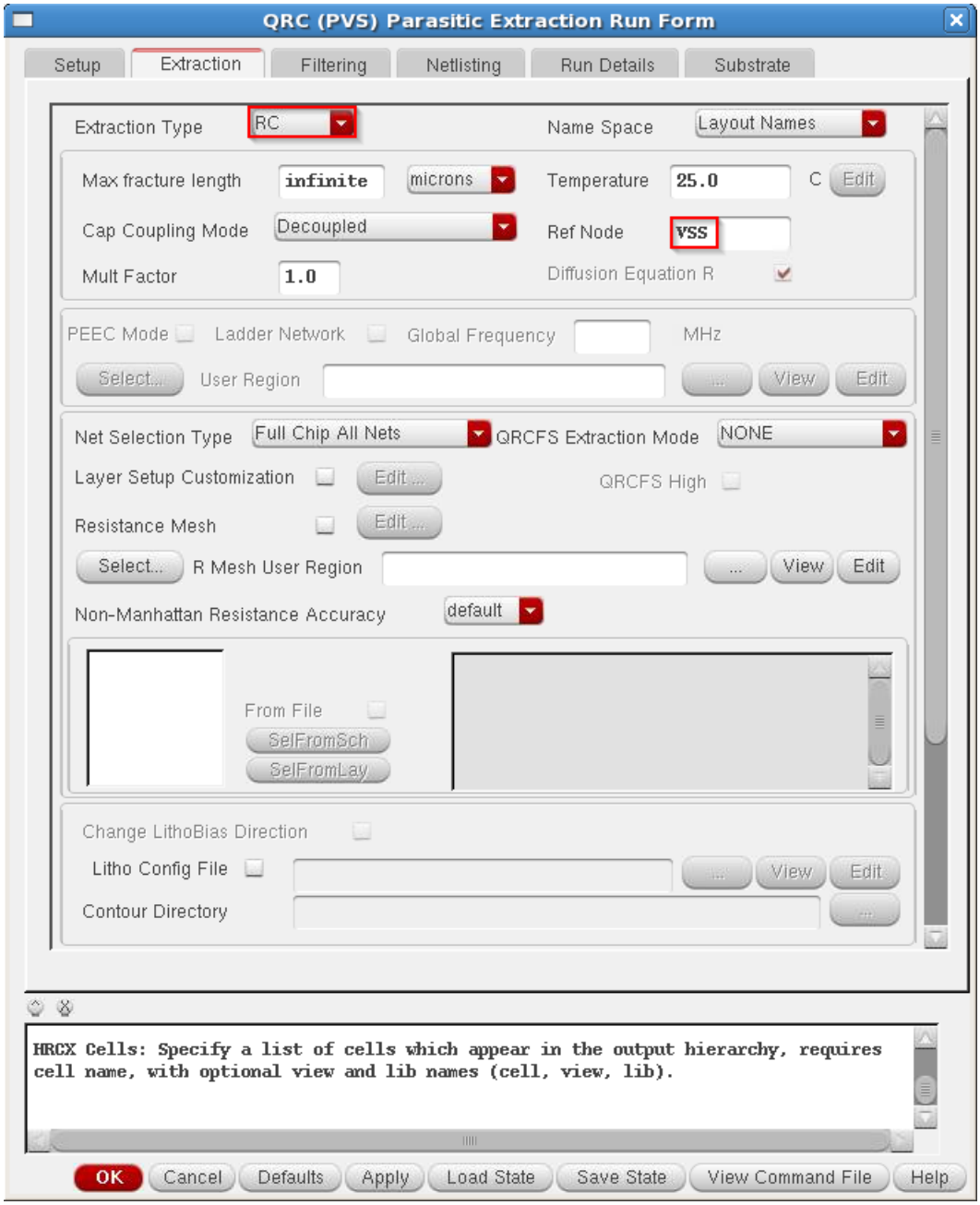
Action 8: In the form that comes up, check to make sure all fields are correctly populated as shown below. The PVS/ QRC Data Directory should point to the same directory as the LVS run, the QRC Tech Lib should point to ./tech/GPDK045/gpdk045/pvtech.lib and the Technology should be gpdk045_pvs. Click 'OK' in this form.

Action 9: This will bring up the QRC (PVS) Parasitic Extraction Run form.

- Under the 'Setup' tab, make sure that the Technology is set to gpdk045_pvs and nothing else is checked (everything else should be set to default values, including the output view name which is 'av_extracted1', as shown in the snapshot below).

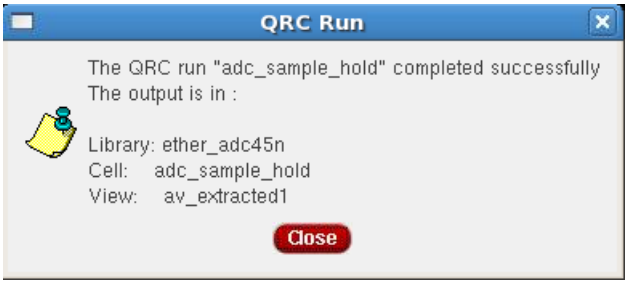


- Under the 'Extraction' tab, make sure that the Extraction Type is set to RC and the Ref node is VSS (everything else should be set to default values, as shown in the snapshot below).



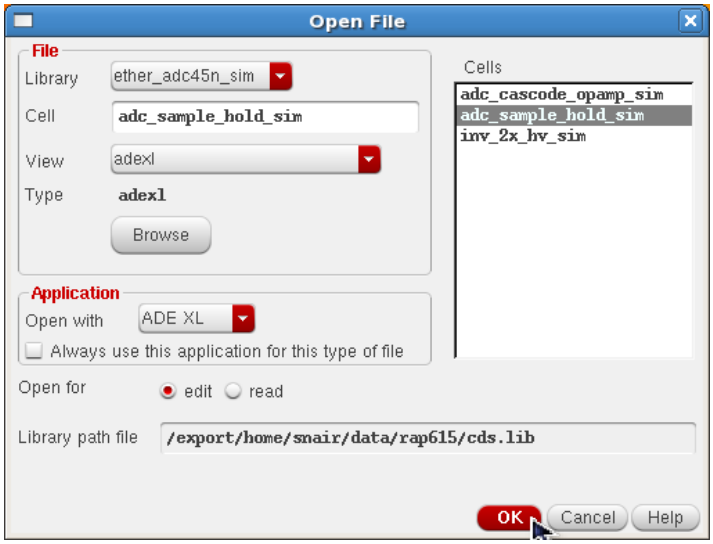
Action 10: Click 'OK' in the Parasitic Extraction Form to start the run. (A prompt may appear to ask if an existing 'av_extracted1' view should be overwritten in which case you should click 'Yes'). A form will then appear to show the progress of the run.

Action 11: When complete, a pop-up form will be displayed indicating that the QRC run has completed successfully and the specified output (view 'av_extracted1' for the cell has been generated. Click on the 'Close' button to close the form.

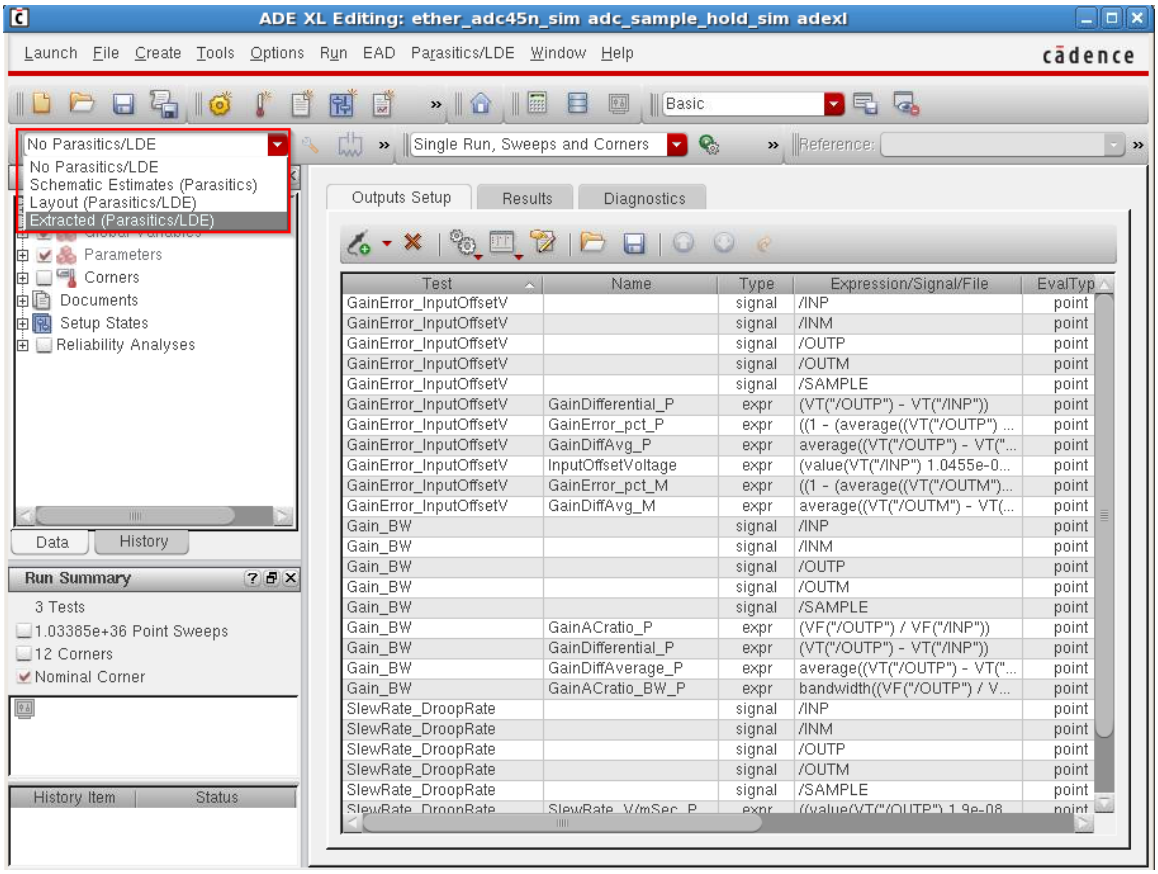


Action 12: Close all Layout and Schematic windows without saving. From the CIW menu pull-down, click on 'File > Open' to open the following:

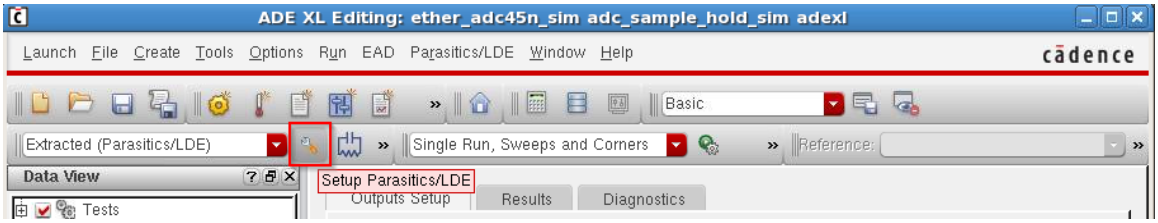
Library: ether_adc45n_sim
Cell: adc_sample_hold
View: adexl



Action 13: In the ADE-XL window, click on the leftmost pull-down to set the parasitic mode to ‘Extracted (Parasitics/ LDE)’.



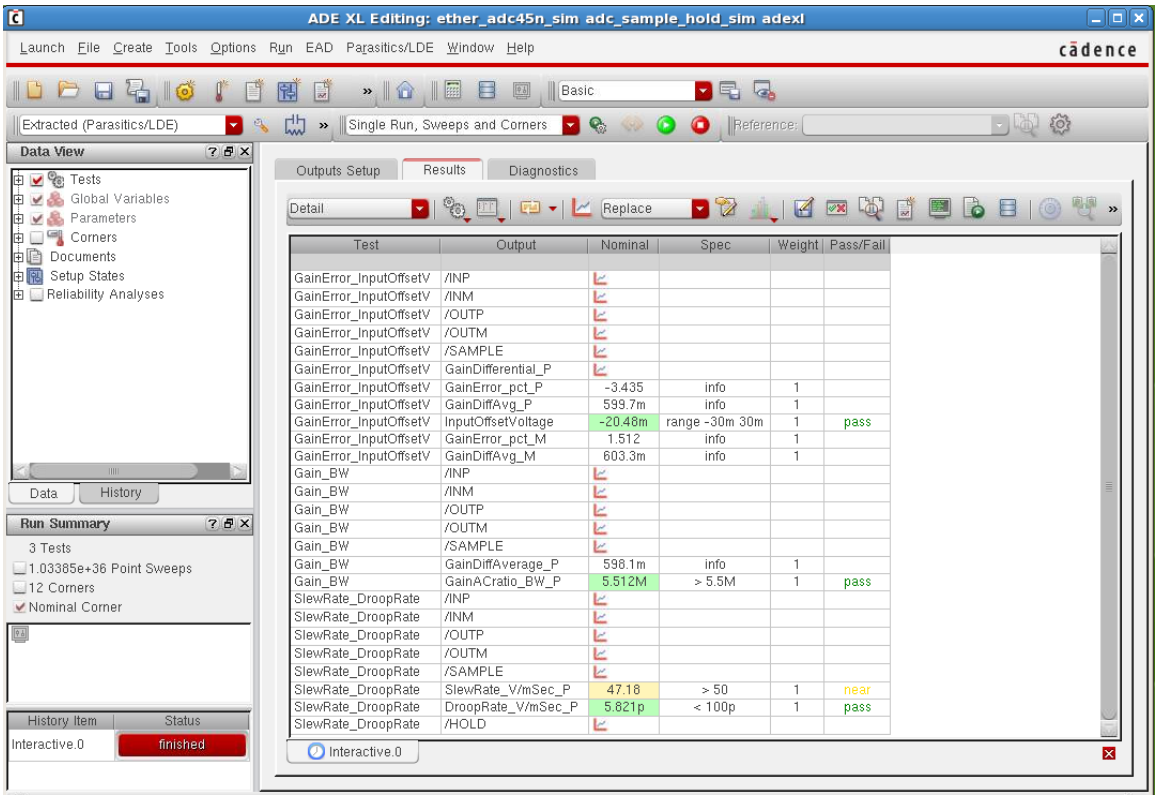
Action 14: Click on the ‘Setup Parasitics’ button to the right of this field.



Action 15: In the form that comes up, make sure that the view for ‘Extracted Parasitics’ is set to ‘av_extracted1’. Click ‘OK’ in this form.

Action 16: Click on the ‘Run Simulation’ button to re-run the same tests as before (as covered in Section 1) but with extracted parasitics.

Action 17: Once the simulation run completes, you can see that the ‘Results’ tab in the ADE-XL window update to reflect the outputs taking into account parasitics and other device effects. The results are also plotted for all the tests in tabs in the ViVA window.



Action 18: You can see that the specs are still met but with some degradations. We can improve the results through minor changes such as adding dummies to the Modgens as covered in the next section.

Action 19: Close the ADE-XL and ViVA windows without saving the setups.

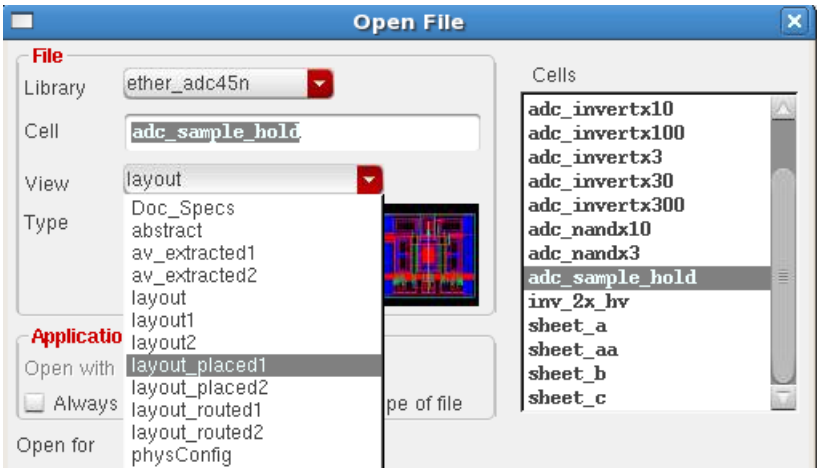
7 Quick iteration following changes in Schematic or Layout

In this section...

We will see how the test results can be improved by modifying the constraints in Schematic or Layout, and going though the flow very quickly.

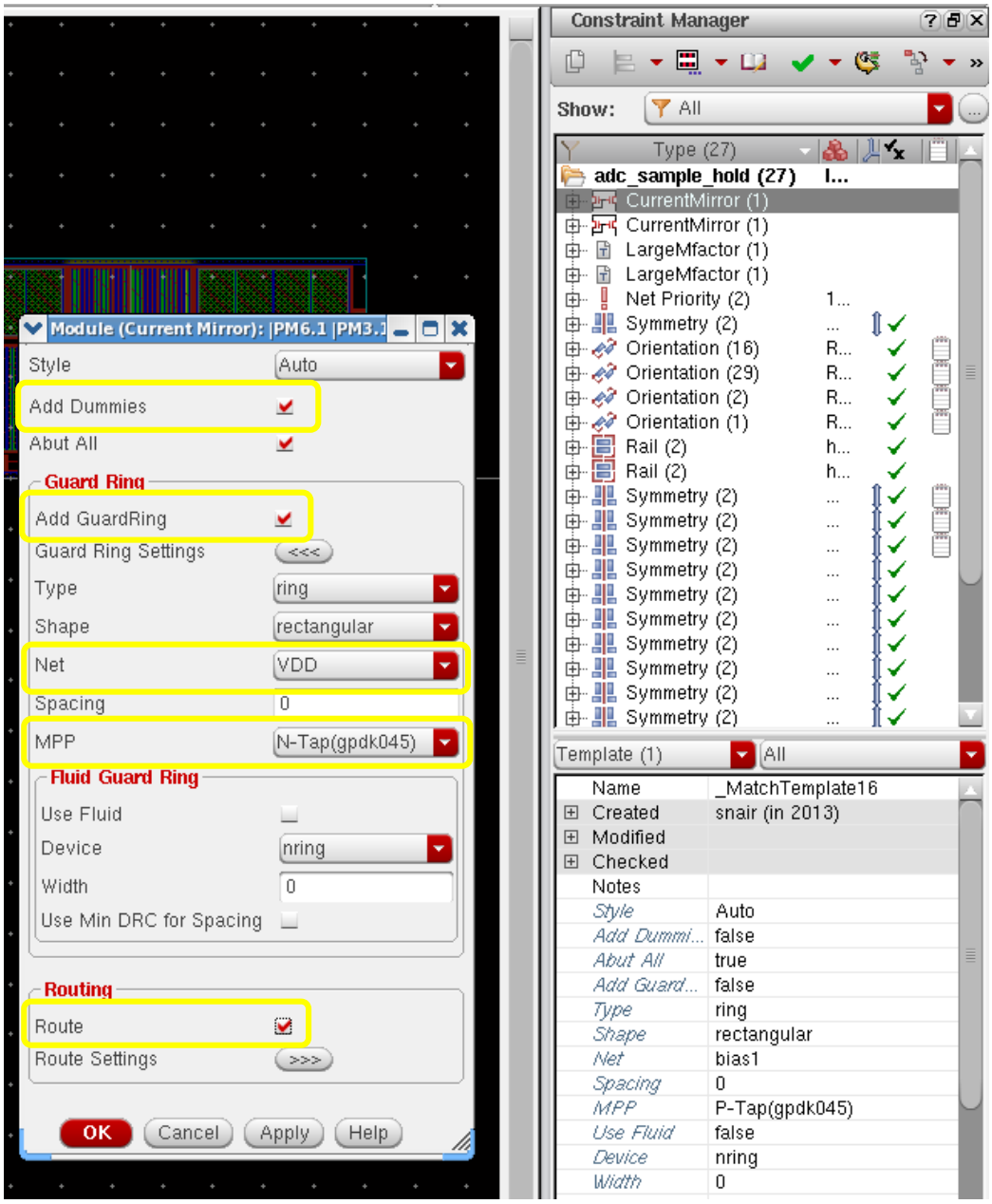
Action 1: Reopen Schematic 'ether_adc45n adc_sample_hold schematic'. (You can bring it up directly from 'File' menu in the CIW that shows the recently opened views).

Action 2: From the Schematic window, click on 'Launch > Layout-GXL'. In the form that comes up, select 'Open Existing'. We will start with the previously generated layout (placed only and not routed as it will need to be rerouted after the changes). Pick the view 'layout_placed1' which is the saved placement from the last run and click 'OK'.

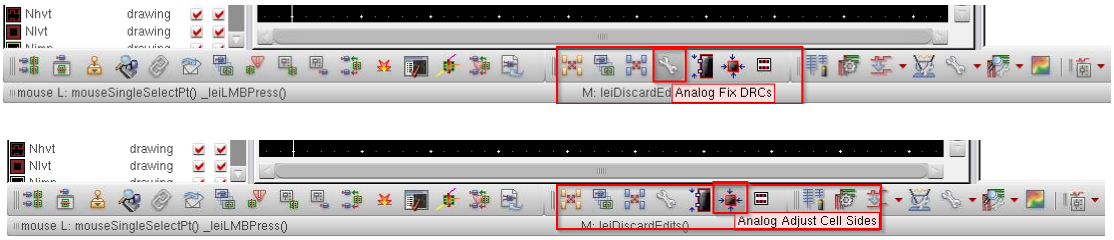


Action 3: In the Constraint Manager of the Layout Window after it comes up, click on the top Current Mirror. Modify its parameters as follows, in the same order to minimize unnecessary triggers. (This will create some overlaps in the Layout which we will fix in the next step).

MPP: N-Tap
Net: VDD
Add Guardring: True
Add Dummies: True
Route: True

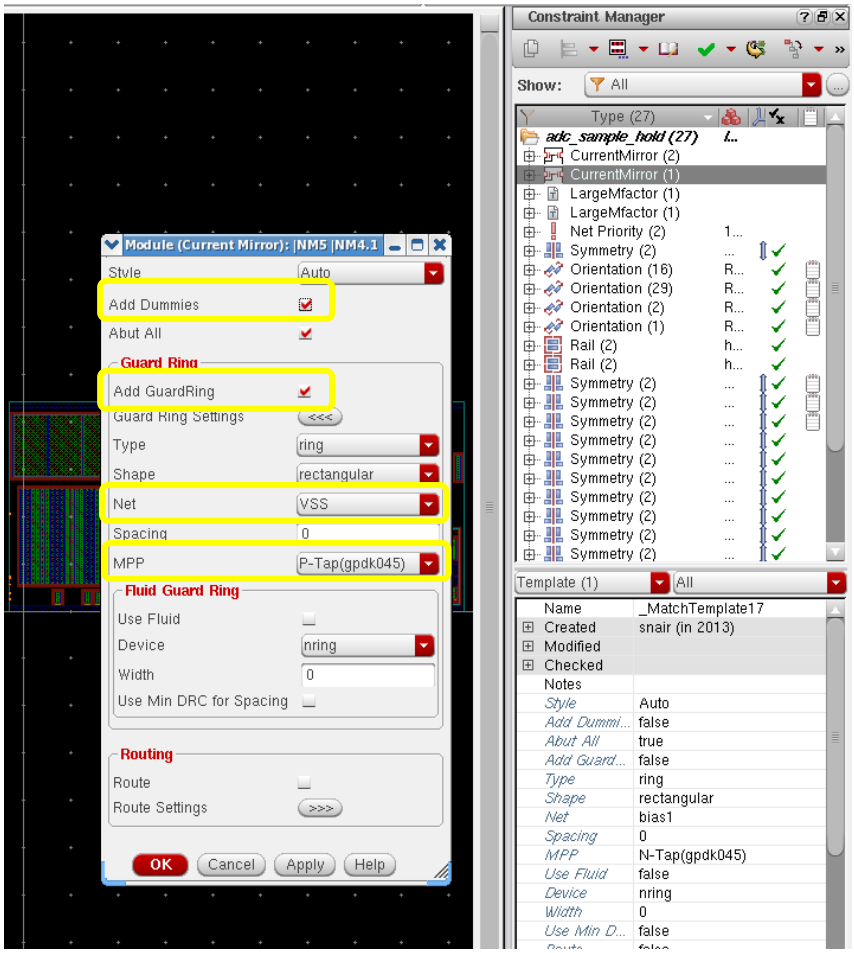


Action 4: Click on the ‘Analog Fix DRC’ button in the Analog Placer toolbar followed by ‘Analog Adjust Cell Sides’ also in the same toolbar, as shown below, or from the ‘Place - Analog’ Menu in the Layout window. This will fix the overlaps and extend the cell boundary to accommodate the changes.



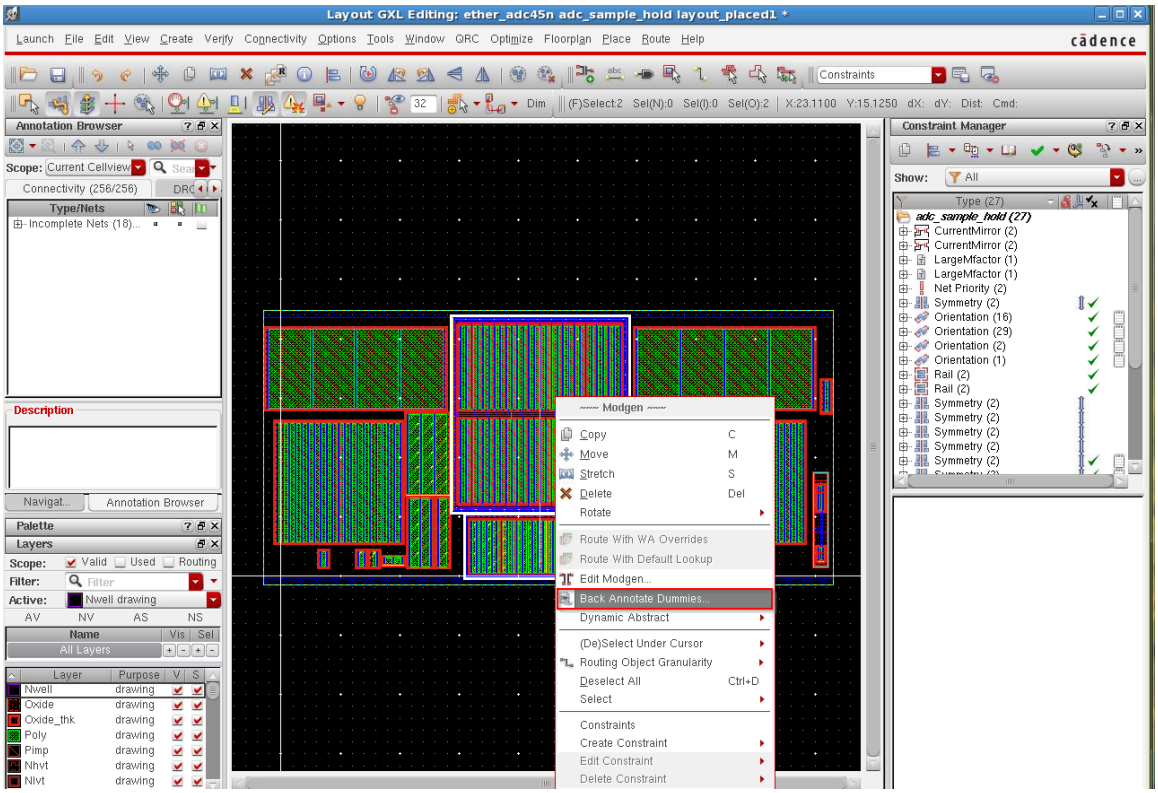
Action 5: Now select the bottom Current Mirror in the Constraint Manager and make changes to it as follows, in the same order (to minimize unnecessary triggers). Again, this will create some overlaps in the Layout but we will fix them in subsequent steps.

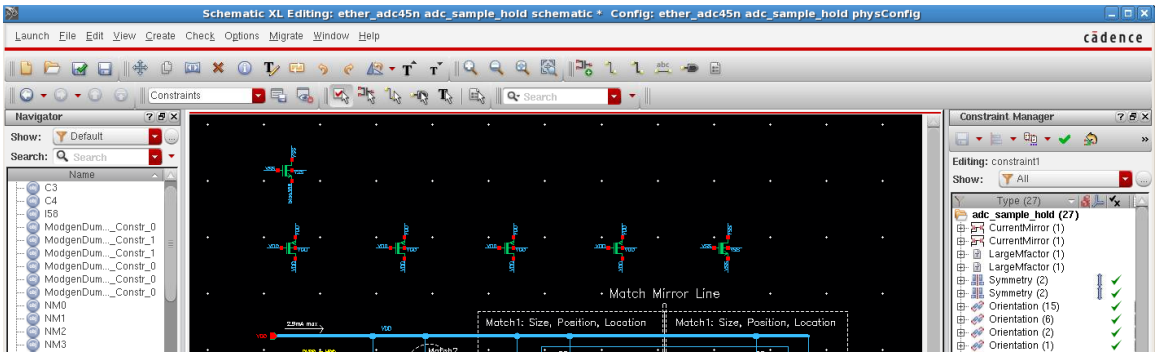
MPP: P-Tap *Net: VSS*
Add Guardring: True *Add Dummies: True*



Action 6: Repeat step 4 (‘Analog Fix DRC and ‘Analog Adjust Cell Sides’) to fix the overlaps.

Action 7: While adding dummies does not affect the netlist, we will back-annotate them into the schematic for LVS purposes. Click and select the two Modgens in the Layout and from the RMB options, select ‘Back Annotate Dummies’. You can see the Schematic getting updated with the dummy instances at the top, with the same connectivity as the dummies inside the Modgen.

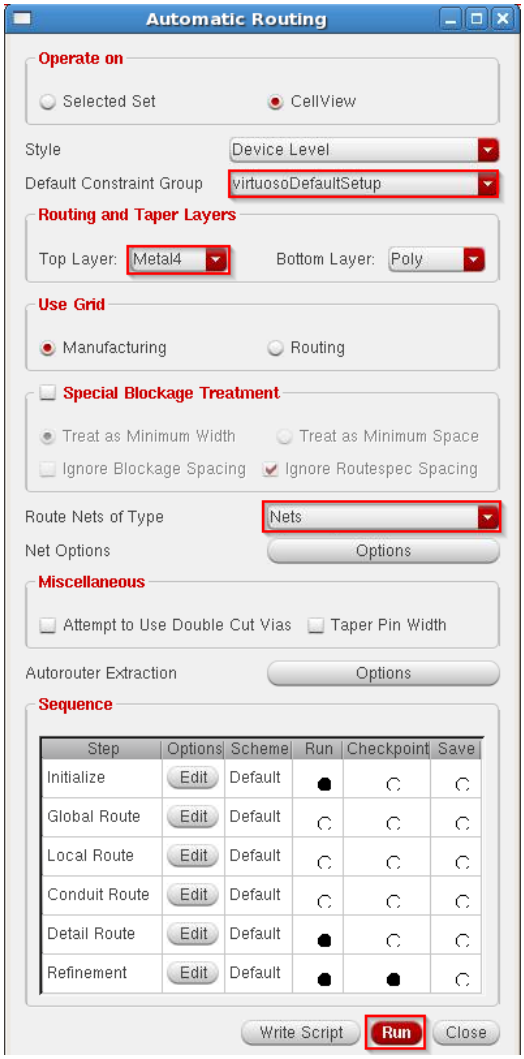




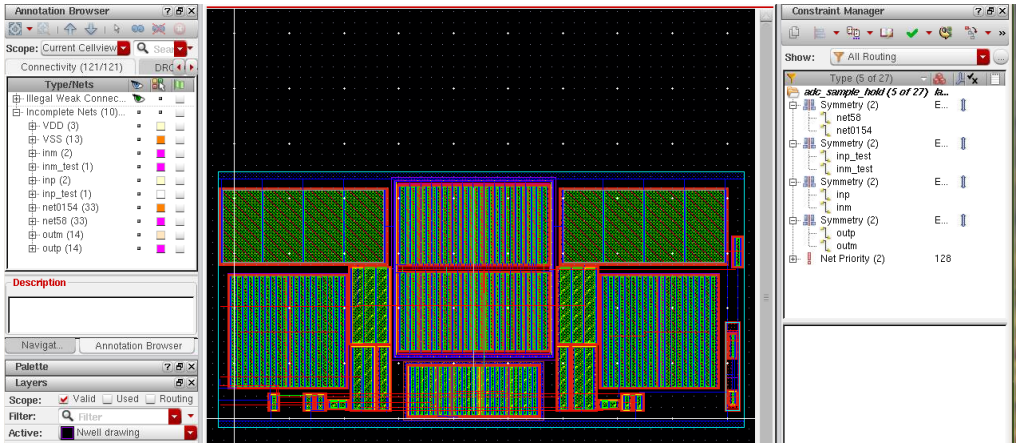
Note: This operation requires the Schematic to be editable, otherwise the operation will fail. Make sure that the Schematic is in ‘Edit’ mode.

Action 8: We will now use the automatic routing functionality once again to route the nets in the Layout. Click on ‘Automatic Routing’ icon in the ‘Space Based Router’ toolbar at the bottom of the Layout window.

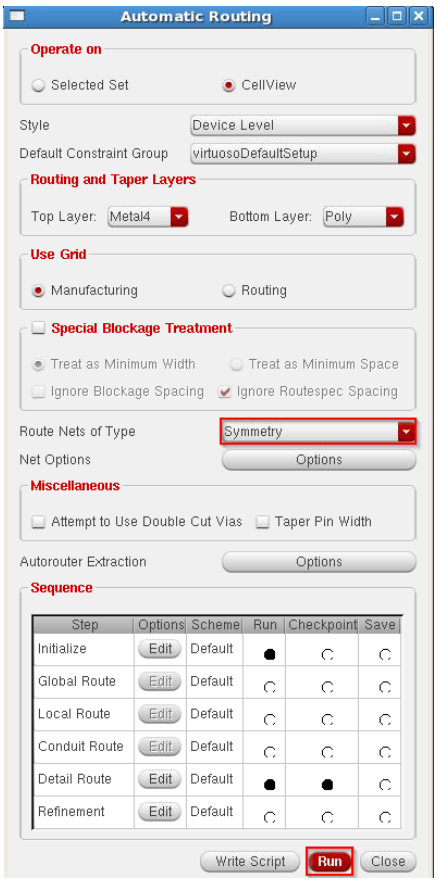
Action 9: In the form, specify the Default Constraint Group as ‘virtuosoDefaultSetup’, and enter the top and bottom layers as Metal4 and Poly respectively. Click on the ‘Run’ button the run the automatic router through the default steps as shown.



Action 10: When the routing completes, click on the Annotation Browser Assistant and expand the ‘Incomplete Nets’. Notice that all nets (except for VDD and VSS) that show up as opens are those that have specialty routing constraint (in this case, symmetry) on them. We will now route the symmetric nets.



Action 11: In the ‘Automatic Routing’ form that is still up, click on the cyclic pull-down under the field ‘Route Nets of Type’ and select ‘Symmetry’ as shown in the screenshot below. Click once again on the ‘Run’ button.

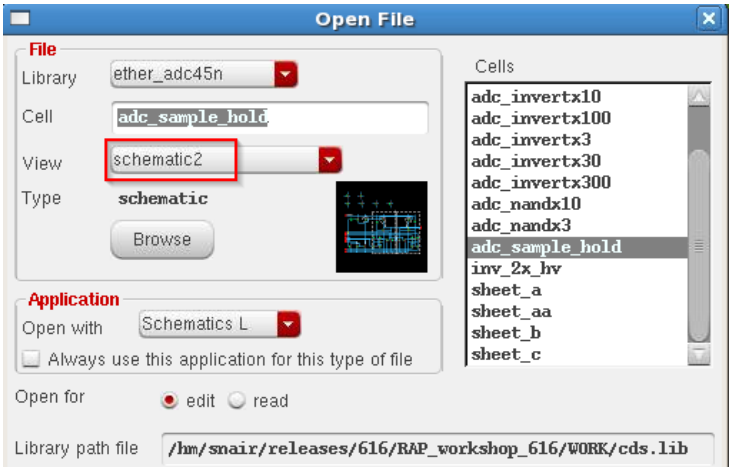


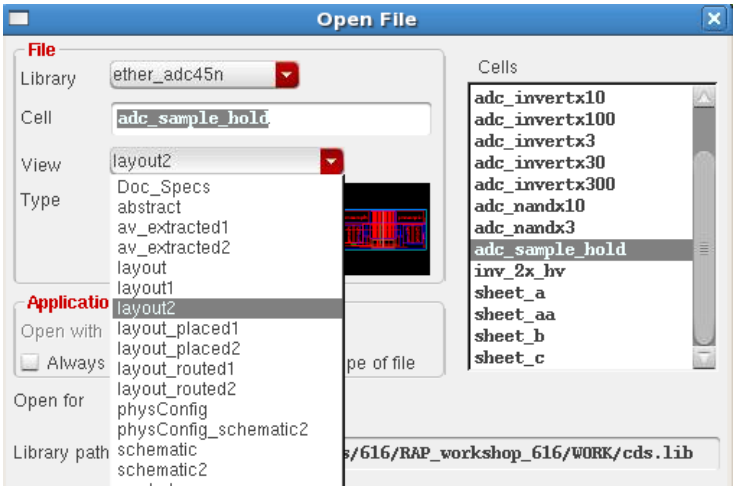
Action 12: You should now see that the nets with symmetry constraints have been routed as well and the only opens left in the design, as displayed in the Annotation Browser, are nets whose wells have not been merged and taps inserted. You may also see some illegal weak connect violations in the Annotation Browser which we can ignore for now. Go ahead and close the 'Automatic Routing' form.

Action 14: At this point, we are ready to run LVS and extract the Layout. We just need to draw the well and insert the well and substrate taps. This has been done already in layout view 'layout2'. Close the current Layout as well as the Schematic without saving.

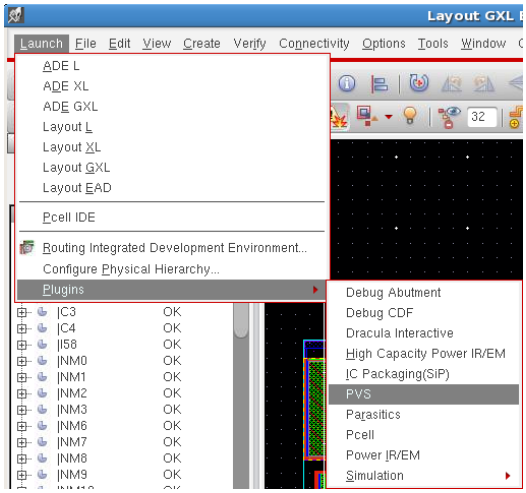
Note: A copy of the Schematic (view name 'schematic2') with the dummies has been saved for LVS purposes, and a constraint view 'constraint2' with constraints updated from the modified Layout constraints (Modgens) has been saved as well.

Action 15: Open Schematic 'ether_adc45n adc_sample_hold schematic2' and click on 'Launch > Layout-GXL' to open existing layout view 'layout2'.

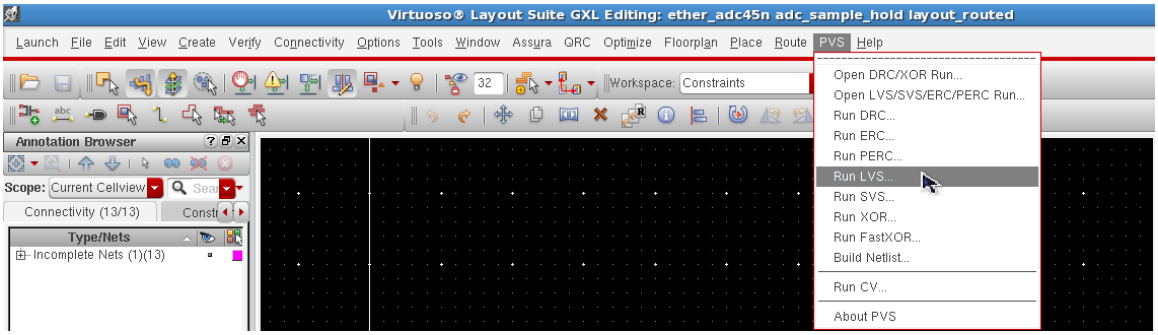




Action 16: From the Layout window, click on ‘Launch > Plugins > PVS’ to install the PVS menu in the Layout window.



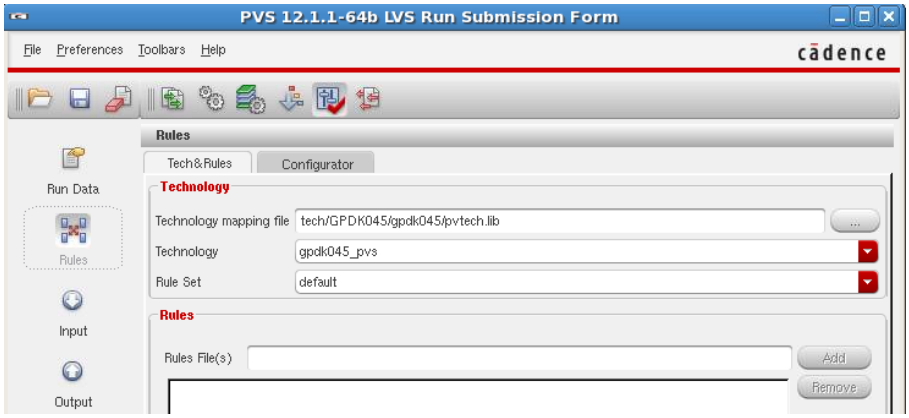
Action 17: From the PVS pull-down menu that appears, click on ‘Run LVS …’.



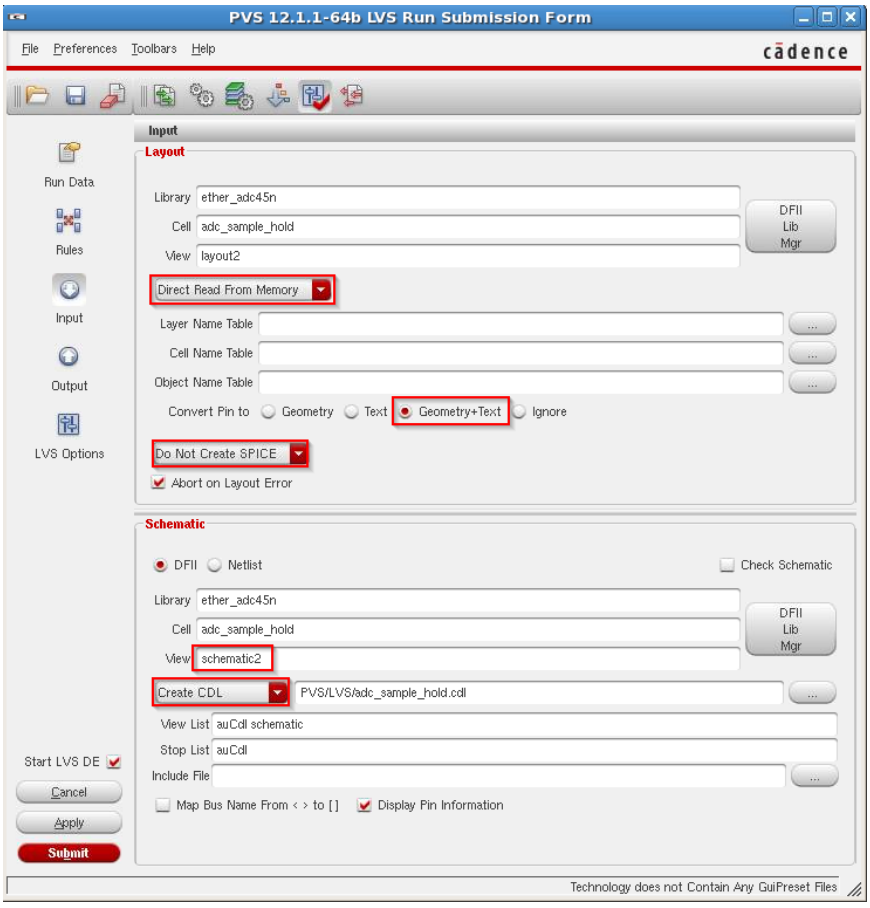
Action 18: In the LVS Run Submission form that comes up, set up the necessary fields as follows.

- Under ‘Run Data’, click on the browse button to specify the path to the Run Directory (./PVS/LVS/).

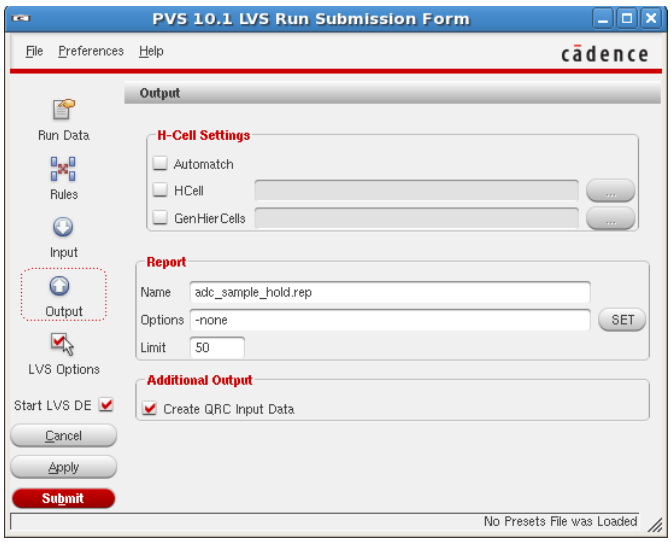
- Under ‘Rules’, specify the path to the Technology Mapping File for this PDK (./tech/GPDK045/gpdk045/pvtech.lib). Click on the Technology pull-down to select gpdk045_pvs, with the default Rule Set.



- Under ‘Input’, make sure that the Library, Cell and View names are populated correctly for the given Layout. Choose ‘Direct Read From Memory’, ‘Convert Pin to Geometry + Text’, ‘Do Not Create SPICE’ (we will create an extracted view instead) and ‘Create CDL’ for the given Schematic (**make sure the schematic view name is set to schematic2**).



- Under ‘Output’, make sure you click on ‘Create QRC Input Data’ as this will be required to run QRC for extraction following the LVS run.



Action 19: Click on the ‘Submit’ button in the LVS Run Submission form to start the LVS run. You may get a pop-up saying that a non-empty directory exists and if it is OK to overwrite. Click ‘Yes’ in that form. You may also get a form that asks to save any unsaved cell view. Click ‘Yes’ also in that form.

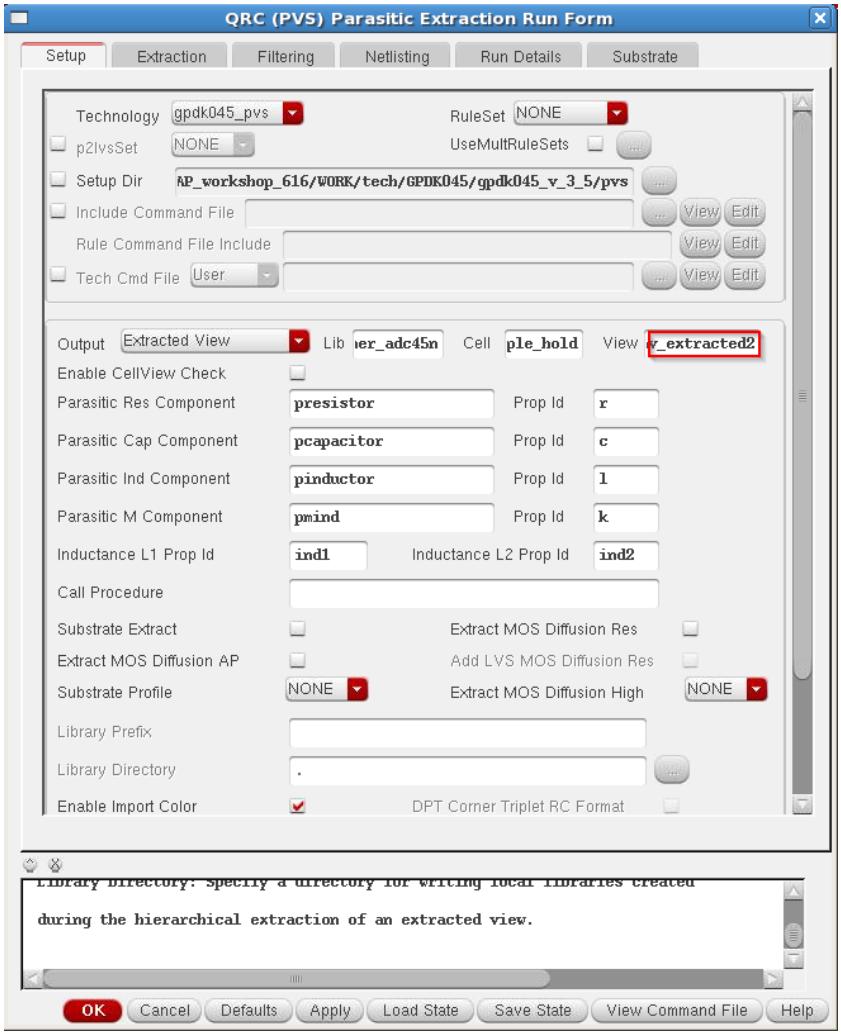
Action 20: This will open up a report window displaying the progress of the LVS run. Once the run is complete, an LVS Run Status form will come up to show that the comparison resulted in a match. Click ‘No’ to the prompt in the status form since there is no mismatch and hence no need to start the LVS debugger. Also close the report window.

Action 21: From the QRC menu pull-down in the Layout window, click on ‘Run (PVS) QRC …’.

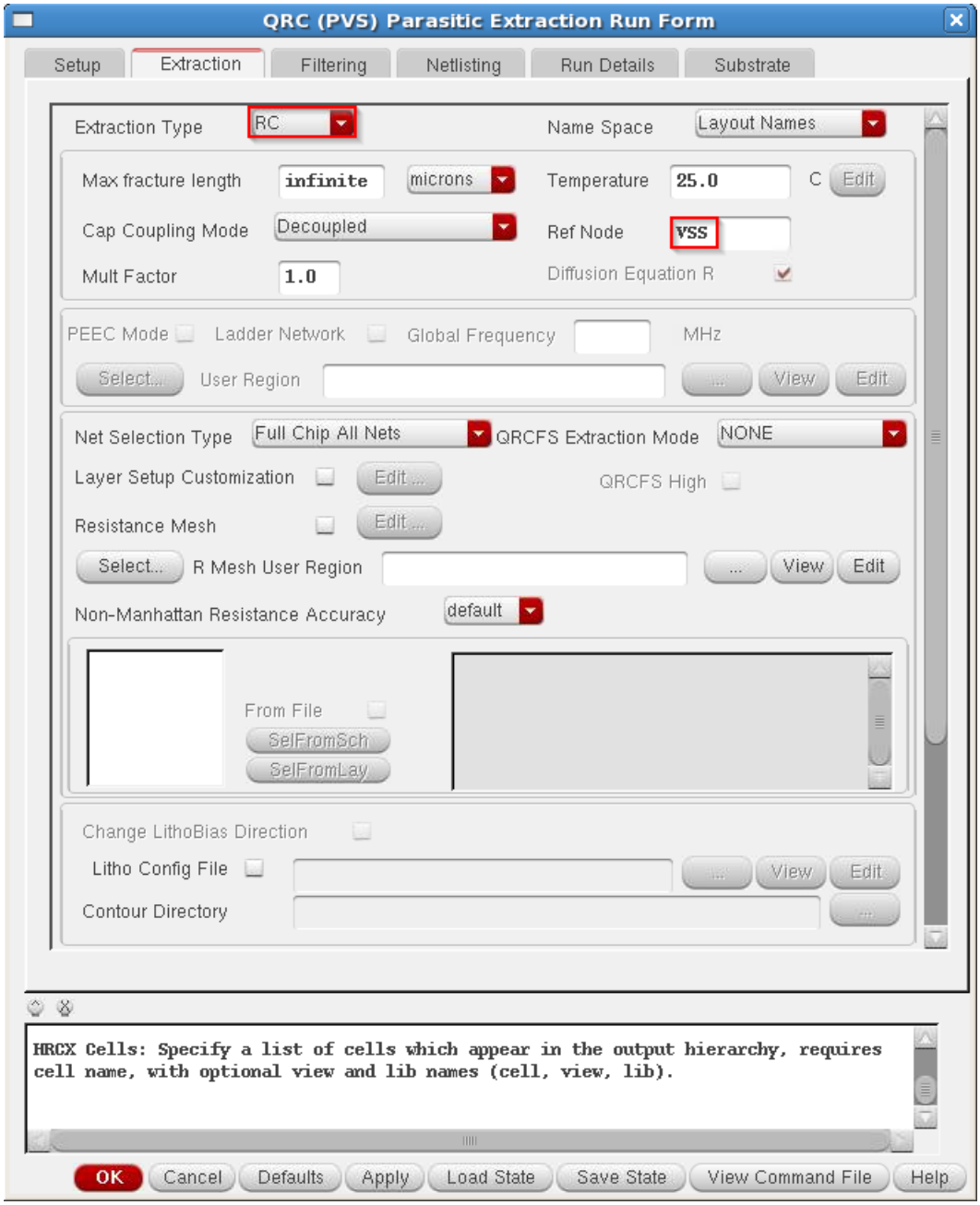
Action 22: In the form that comes up, check to make sure all fields are correctly populated as shown below. The PVS/ QRC Data Directory should point to the same directory as the LVS run, the QRC Tech Lib should point to ./tech/GPDK045/gpdk045/pvtech.lib and the Technology should be gpdk045_pvs. Click 'OK' in this form.

Action 23: This will bring up the QRC (PVS) Parasitic Extraction Run form.

- Under the 'Setup' tab, make sure that the Technology is set to gpdk045_pvs and nothing else is checked (everything else should be set to default values, including the output view name which is 'av_extracted2', as shown in the snapshot below).

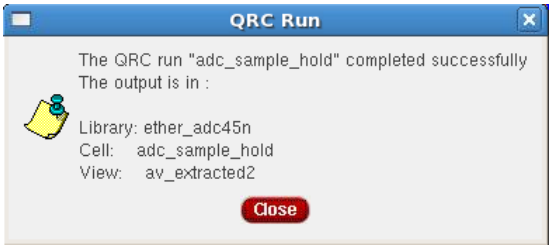


- Under the ‘Extraction’ tab, make sure that the Extraction Type is set to RC and the Ref node is VSS (everything else should be set to default values, as shown in the snapshot below).



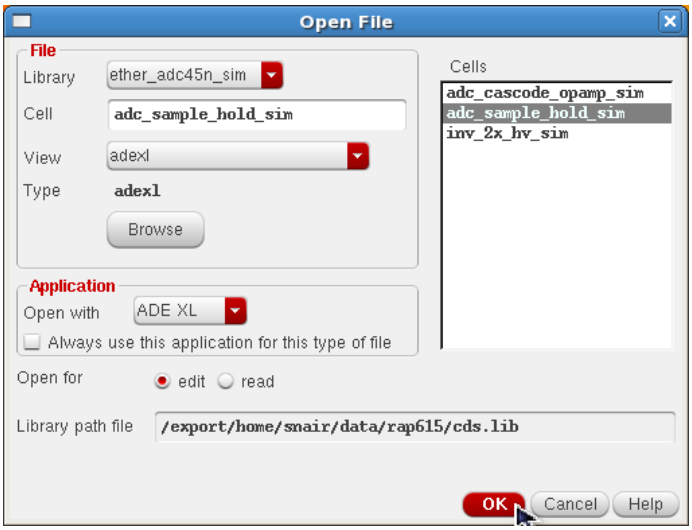
Action 24: Click ‘OK’ in the Parasitic Extraction Form to start the run. (A prompt may appear to ask if an existing ‘av_extracted2’ view should be overwritten in which case you should click ‘Yes’). A form will then appear to show the progress of the run.

Action 25: When complete, a pop-up form will be displayed indicating that the QRC run has completed successfully and the specified output (view 'av_extracted2' for the cell has been generated. Click on the 'Close' button to close the form.

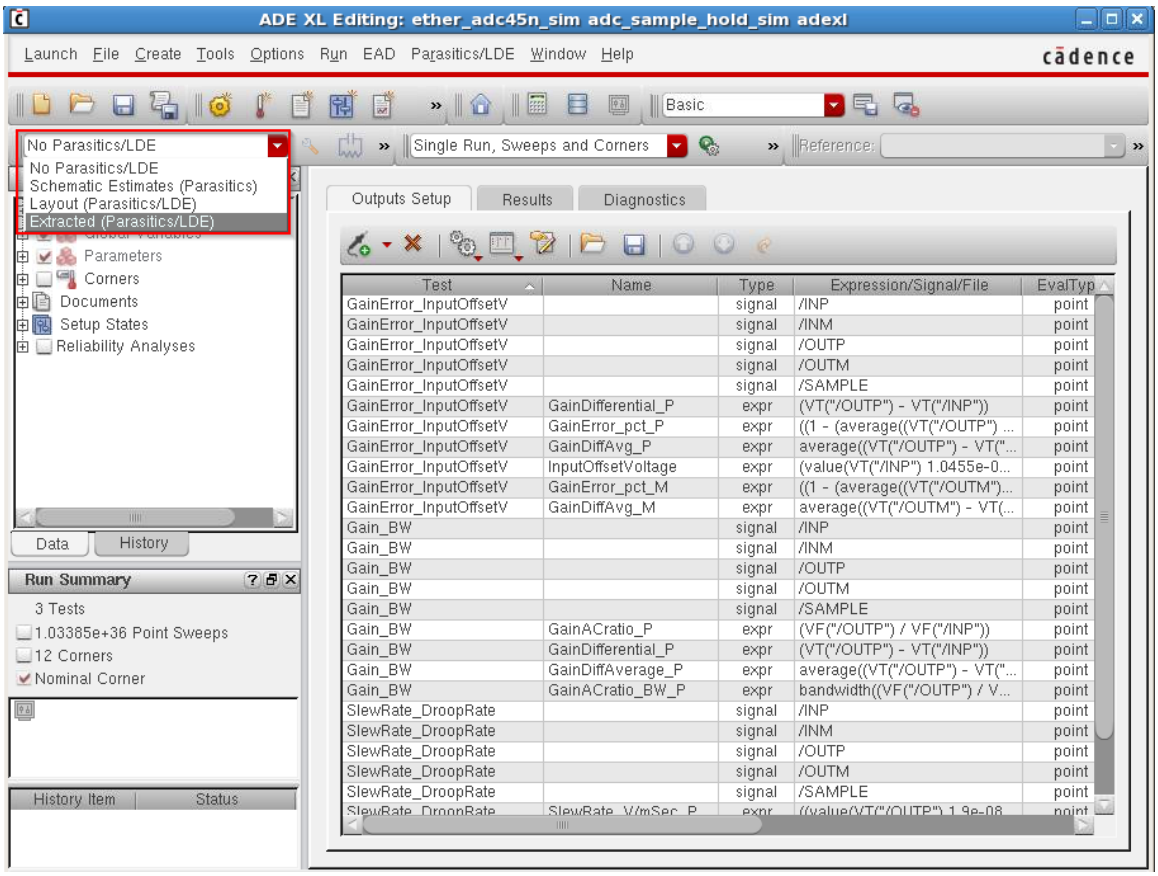


Action 26: Close all Layout and Schematic windows. From the CIW menu pull-down, click on 'File > Open' to open the following:

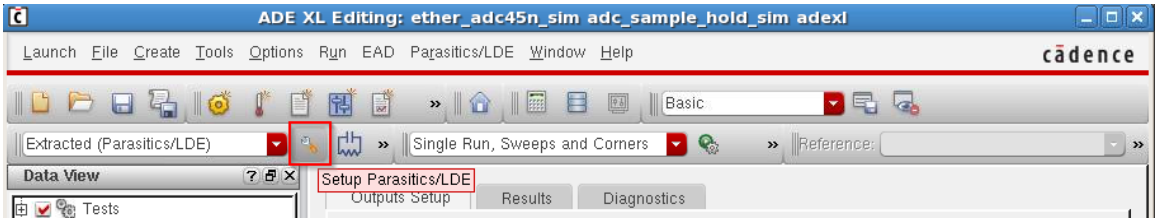
Library: ether_adc45n_sim
Cell: adc_sample_hold
View: adexl



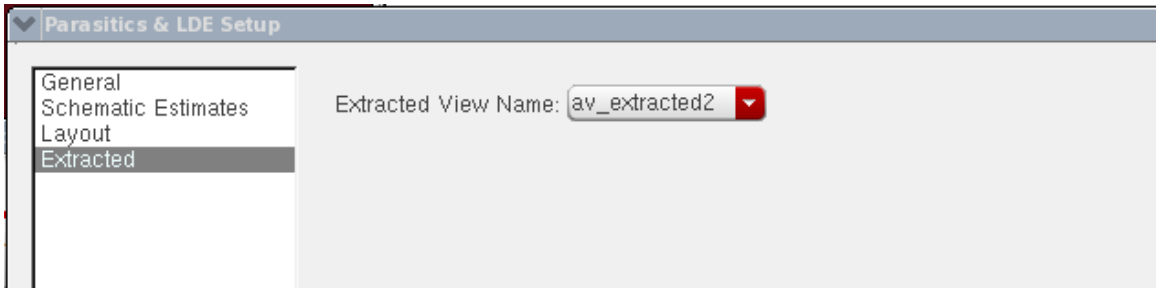
Action 27: In the ADE-XL window, click on the leftmost pull-down to set the parasitic mode to 'Extracted (Parasitics/ LDE)'.



Action 28: Click on the ‘Setup Parasitics’ button to the right of this field.

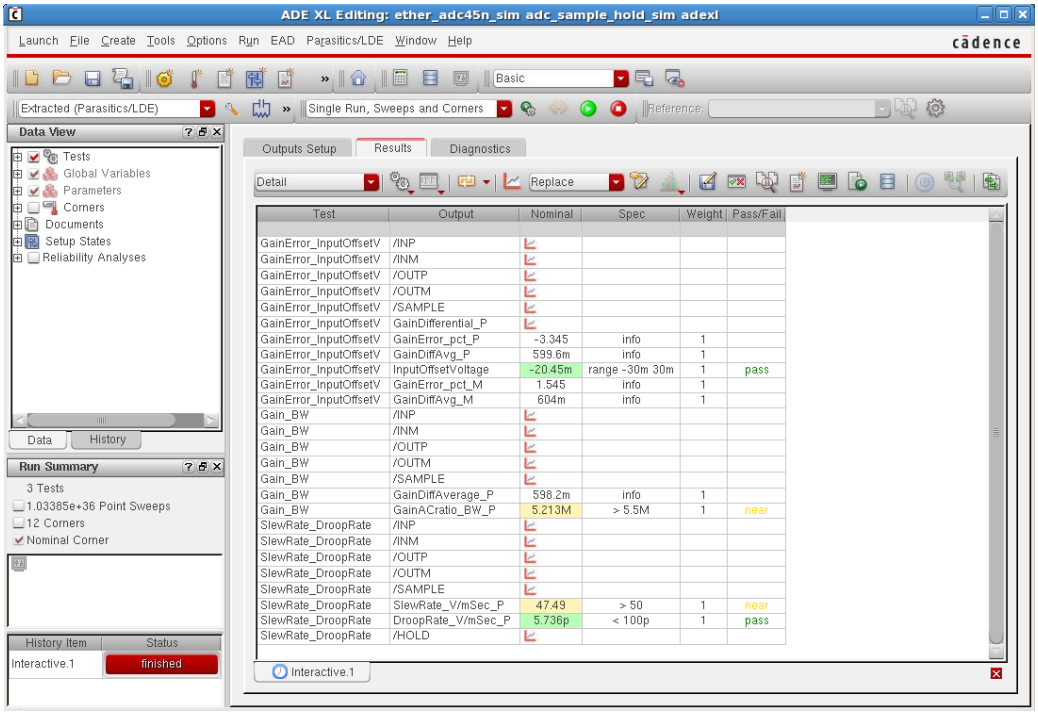


Action 29: In the form that comes up, make sure that the view for ‘Extracted Parasitics’ is set to ‘av_extracted2’. Click ‘OK’ in this form.



Action 30: Click the ‘Run Simulation’ button to run the same tests on this extracted view.

Action 31: Once the simulation run completes, you can see that the ‘Results’ tab in the ADE-XL window update to reflect the outputs taking into account parasitics and other device effects. The results are also plotted for all the tests in tabs in the ViVA window.



Action 32: You can see that the specs are still met but they can be tweaked further by following similar steps as followed in this section. Close the ADE-XL and ViVA windows without saving the setups.

This concludes a brief tour of the Rapid Analog Prototyping Flow in the IC6.1.6 release. We saw how instructions in the original schematic were captured more easily through constraints that were passed to the Layout, automatic Layout generation following these constraints, and the resulting LVS correct Layout extracted to re-simulate the design with parasitics and device effects.

Hopefully you will find these features useful and helpful in improving design productivity and analysis.