DRAFT

Pre-Characterized Device Extraction Flow Overview

26 April 2005
James Paris/Myron Lin

Description

The pre-characterized device (PCD) flow is an extraction methodology that supports parameterized simulation models that have parasitic information included within the device model. Physical layout of these devices is done through device generators or parameterized cell (Pcells). To insure accurate simulation results when using parameterized models, the extraction tool must not extract parasitics inside the specified devices.

General Requirements

To properly use this extraction flow, the design must use Parameterized cells and models that account for all of the parasitics in the Pcell. Even though this design is transistor level, the Gate Level option for Calibre xRC must be set. Each of these models must have an entry in the "hcell" file and identical entries in the "xcell" file. It's possible to use wildcards (*) in the hcell list, which makes it easier to create the file since Pcells have random numbers generated after the cell name once they have been translated to GDSII. Wildcards in the xcell file is released in Calibre version 2005.1.

Example hcell file:

```
pmos_rf*      pmos_rf
nmos_rf*      nmos_rf
ind_rf*       ind_rf
…
```

In Calibre 2005.1, you can use the SVRF command to switch YES/NO to decide would you like to use PCDEF in extraction:

```
PEX IDEAL XCELL YES
```

Calibre View Requirements

If the intent is to simulate this design within the Cadence Analog Design Environment, then a working Calibre View flow must be in place. The design kit must have a cell-mapping file; available from most foundry Cadence based design

kits. The LVS rules file used in the extraction flow must contain the following statements:

SOURCE CASE YES
LVS COMPARE CASE NAMES

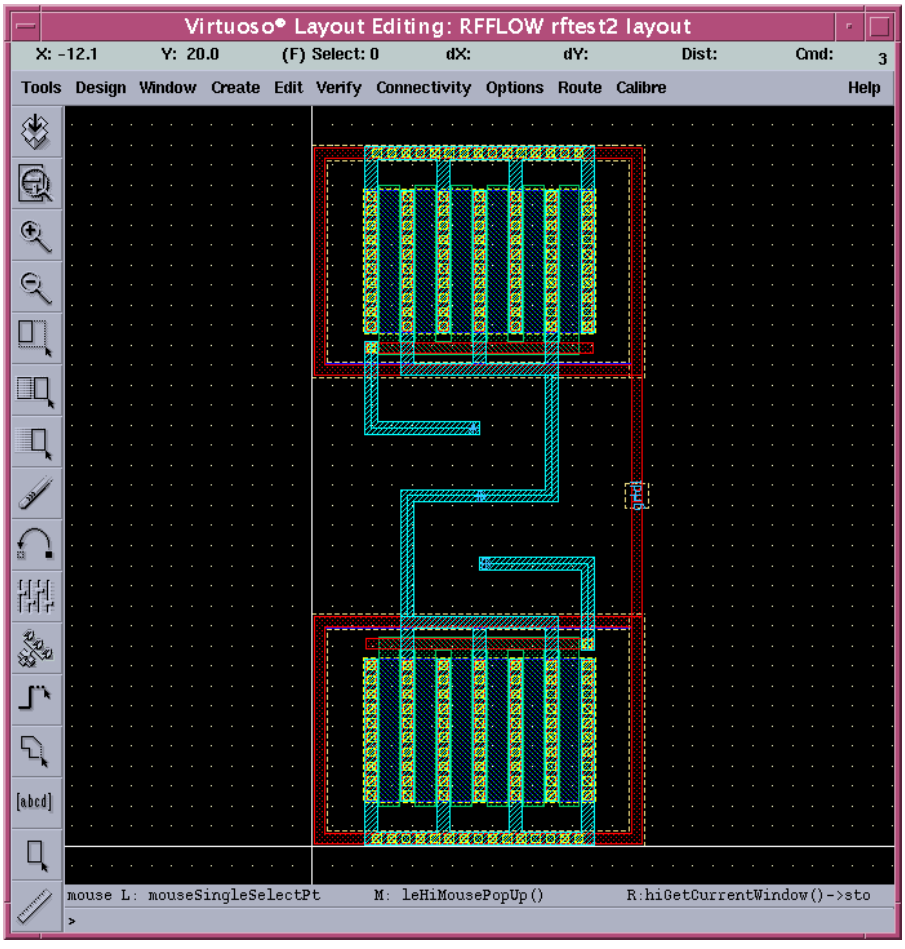
These will insure that the names are correct when back-annotating to the Calibre View schematic. Cadence versions 4.4.x –5.x are supported.

Creating Calibre View

1. Add PEX IDEAL XCELL YES in your rule file

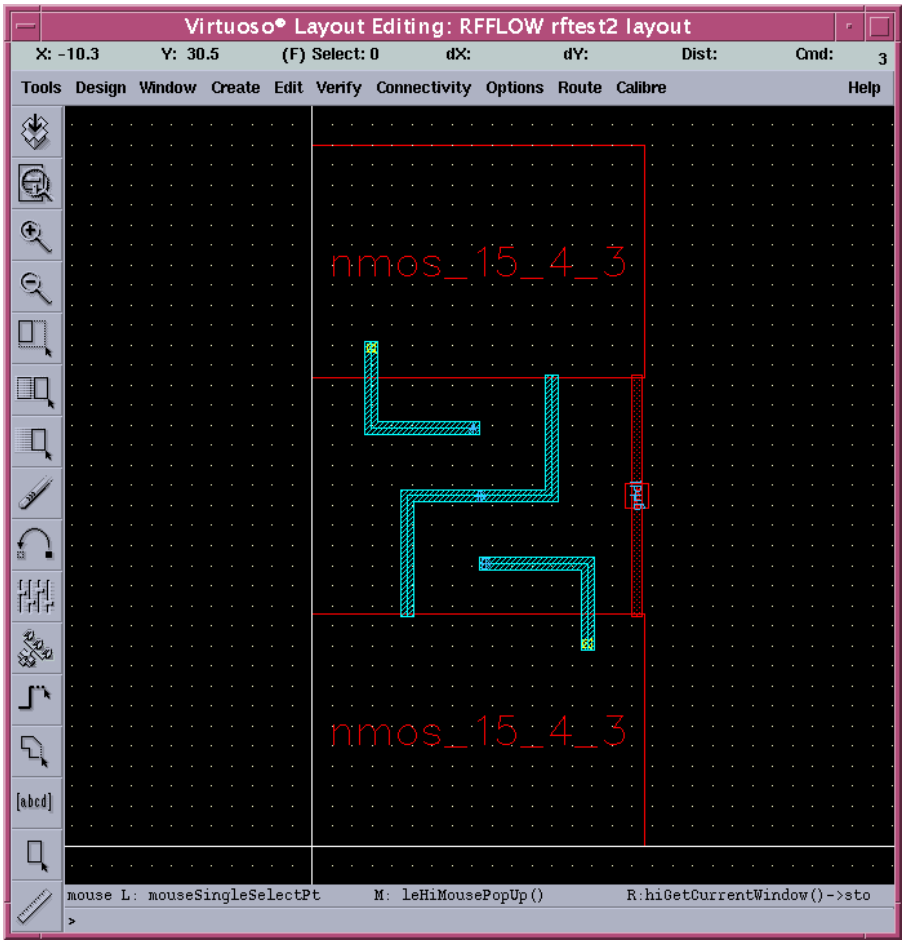
2. Start Cadence

3. Open the design layout view



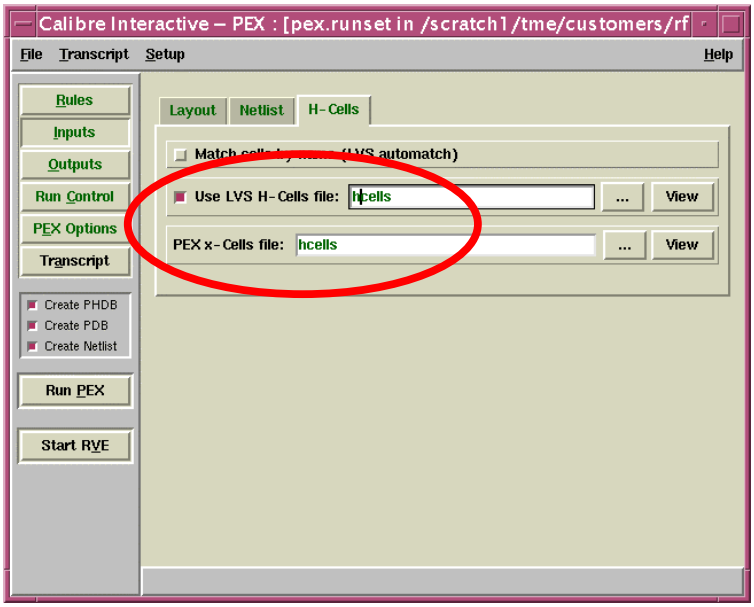
This design has two nmos Pcells in it. The next image displays only the top level of hierarchy so that it is easier to see. No parasitics for the polygons included

inside of the Pcell will be extracted, however, the extracted netlist will contain each device instance and the extracted properties such as L, W, M, etc.



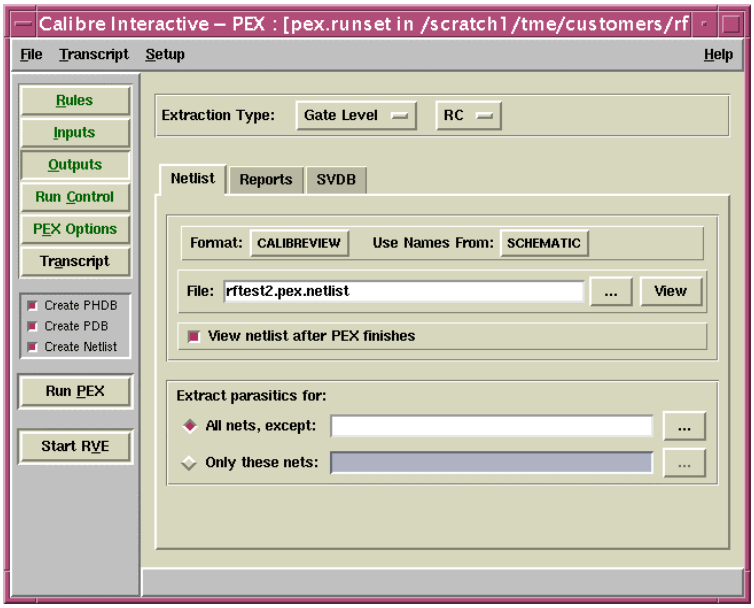
4. Start Calibre Interactive PEX
   Calibre → Run PEX

5. Set hcell/xcell files in Calibre Interactive GUI
   Inputs→ H-Cells



6. Output

   Run Control→Extraction Type→ Gate Level
   Outputs→ Format→CALIBREVIEW
   Outputs→Use names from→ Schematic

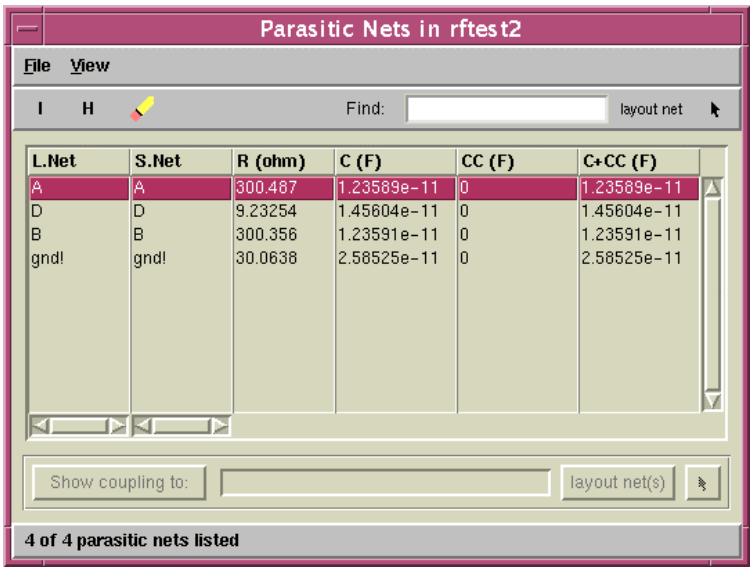


7. Execute Calibre Interactive xRC

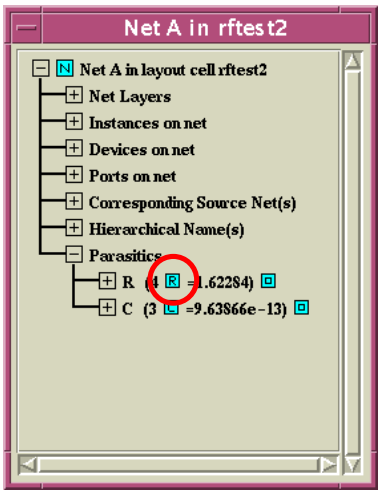
   Run PEX

After the job completes, start Calibre RVE. We can use RVE's Parasitic Visualization capability to verify that no parasitics were extracted inside of the Pcell.

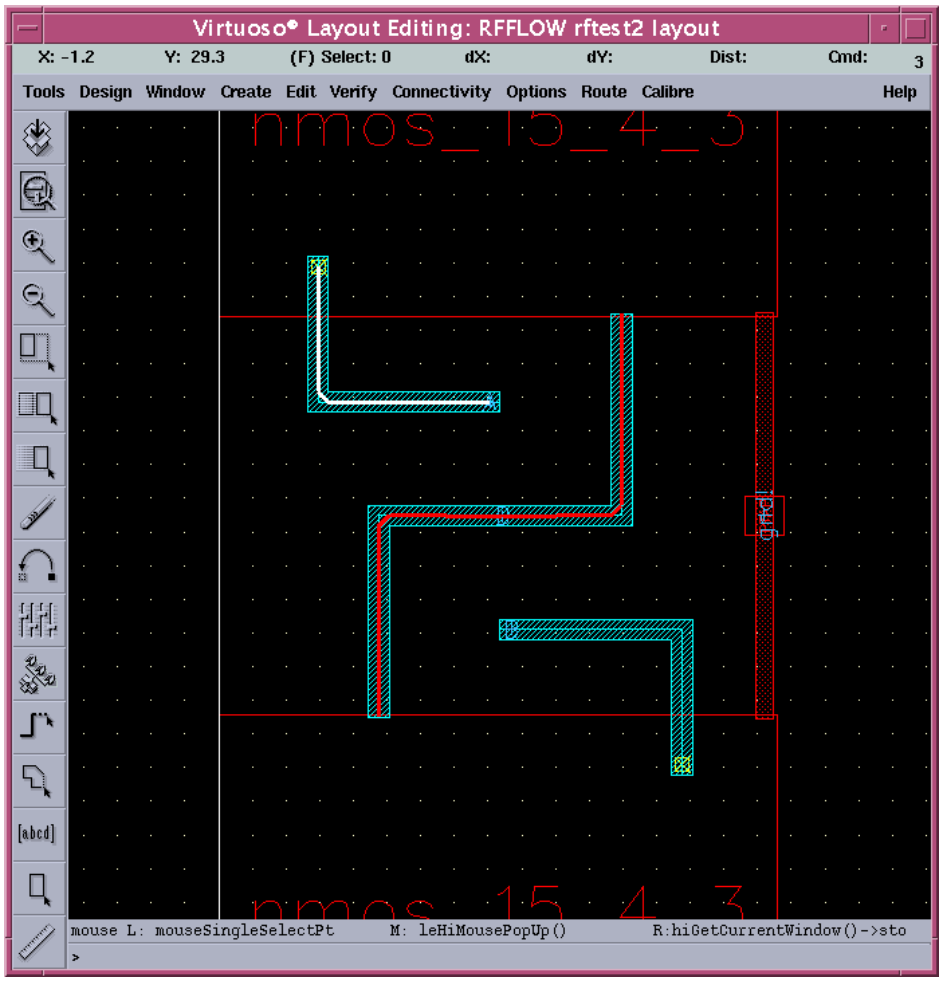
8. Open the Parasitic Visualization window in Calibre RVE
   Layout→Net Parasitics…



In this window are listed all the nets in the design and the extracted parasitic values. To see detailed parasitics on a net, select it, then click on the "I" button. This opens the Net Info window. At the bottom of this window, select "Parasitics". To highlight all the parasitic resistor segments in the layout window, click the blue R in the Net Info window.



Notice in the layout window, each resistor segment is highlighted, but the highlights only go to the pins of each Pcell. Check the other nets by using the Parasitic Visualization window.



Below is the abbreviated extracted netlist showing the intentional devices that were extracted.

```
mgc_rve_cell_start "rftest2" "A" "D" "B" "gnd!"
mr_pi "n" "MN1__1" '("X03" "X02" "X01" "X02") '(("l" 1) ("w" 7) ("m" 1)) '(3.36 2.26)
mr_pi "n" "MN1__2" '("X03" "X02" "X01" "X02") '(("l" 1) ("w" 7) ("m" 1)) '(5.16 2.26)
mr_pi "n" "MN1__3" '("X03" "X02" "X01" "X02") '(("l" 1) ("w" 7) ("m" 1)) '(6.96 2.26)
mr_pi "n" "MN1__4" '("X03" "X02" "X01" "X02") '(("l" 1) ("w" 7) ("m" 1)) '(8.76 2.26)
mr_pi "n" "MN1__5" '("X03" "X02" "X01" "X02") '(("l" 1) ("w" 7) ("m" 1)) '(10.56 2.26)
mr_pi "n" "MN1__6" '("X03" "X02" "X01" "X02") '(("l" 1) ("w" 7) ("m" 1)) '(12.36 2.26)
mr_pi "n" "MN0__1" '("X13" "X12" "X11" "X12") '(("l" 1) ("w" 7) ("m" 1)) '(3.36 32.7)
mr_pi "n" "MN0__2" '("X13" "X12" "X11" "X12") '(("l" 1) ("w" 7) ("m" 1)) '(5.16 32.7)
mr_pi "n" "MN0__3" '("X13" "X12" "X11" "X12") '(("l" 1) ("w" 7) ("m" 1)) '(6.96 32.7)
```

mr_pi "n" "MN0__4" '("X13" "X12" "X11" "X12") '(("l" 1) ("w" 7) ("m" 1)) '(8.76 32.7)
mr_pi "n" "MN0__5" '("X13" "X12" "X11" "X12") '(("l" 1) ("w" 7) ("m" 1)) '(10.56 32.7)
mr_pi "n" "MN0__6" '("X13" "X12" "X11" "X12") '(("l" 1) ("w" 7) ("m" 1)) '(12.36 32.7)