

3DSoC PDK + Libraries: Carbon Nanotube FETs + Resistive RAM



skywater
TECHNOLOGY FOUNDRY



Nanointegris
RAYMOR



“Quick Reference” user guide for PDK “n1”, release 2020_08_03

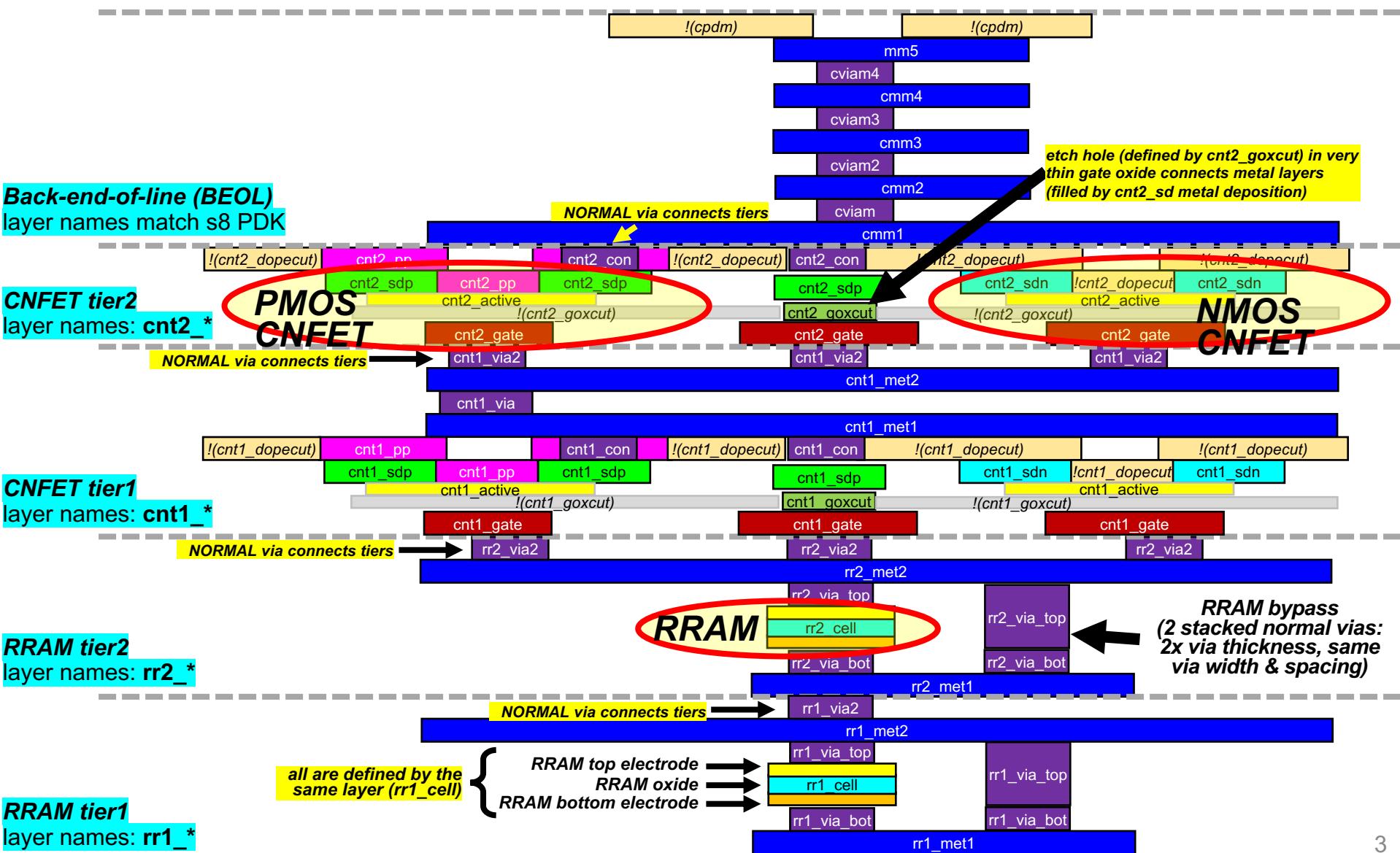
contains confidential information, do not distribute
August 3, 2020

Quick Ref: Overview

- 1) “n1” process: three-dimensional (3D) layer stack (“mask layers” corresponding to physical photomasks)
- 2) “n1” process: 3D layer stack (“drawn layers” used by the designer)
- 3) PDK layer descriptions
- 4) PDK install & tool dependencies
- 5) PDK directory structure
- 6) Carbon nanotube FET (CNFET) experimental calibration + compact model
- 7) Resistive RAM (RRAM) experimental calibration + compact model
- 8) Cadence Virtuoso® Parameterized Cells (PCells): CNFET + RRAM
- 9) Pre-defined vias
- 10) Design Rule Check (DRC)
- 11) Layout vs. Schematic (LVS)
- 12) Parasitics Extraction (PEX) Using GUI
- 13) Parasitics Extraction (PEX) Using GUI: Option 2
- 14) Circuit simulations with parasitics
- 15) Example designs: included for reference & testing
- 16) Example design: 1T-1R (CNFET-RRAM)
- 17) Standard cell libraries: overview, timing/power (.lib), abstract view (.lef), PEX netlists (.pex.sp)
- 18) Standard cell libraries: table of all cells
- 19) Example synthesis, place & route scripts
- 20) Stream out GDS from Cadence Innovus®, stream into Cadence Virtuoso®
- 21) Questions, release notes

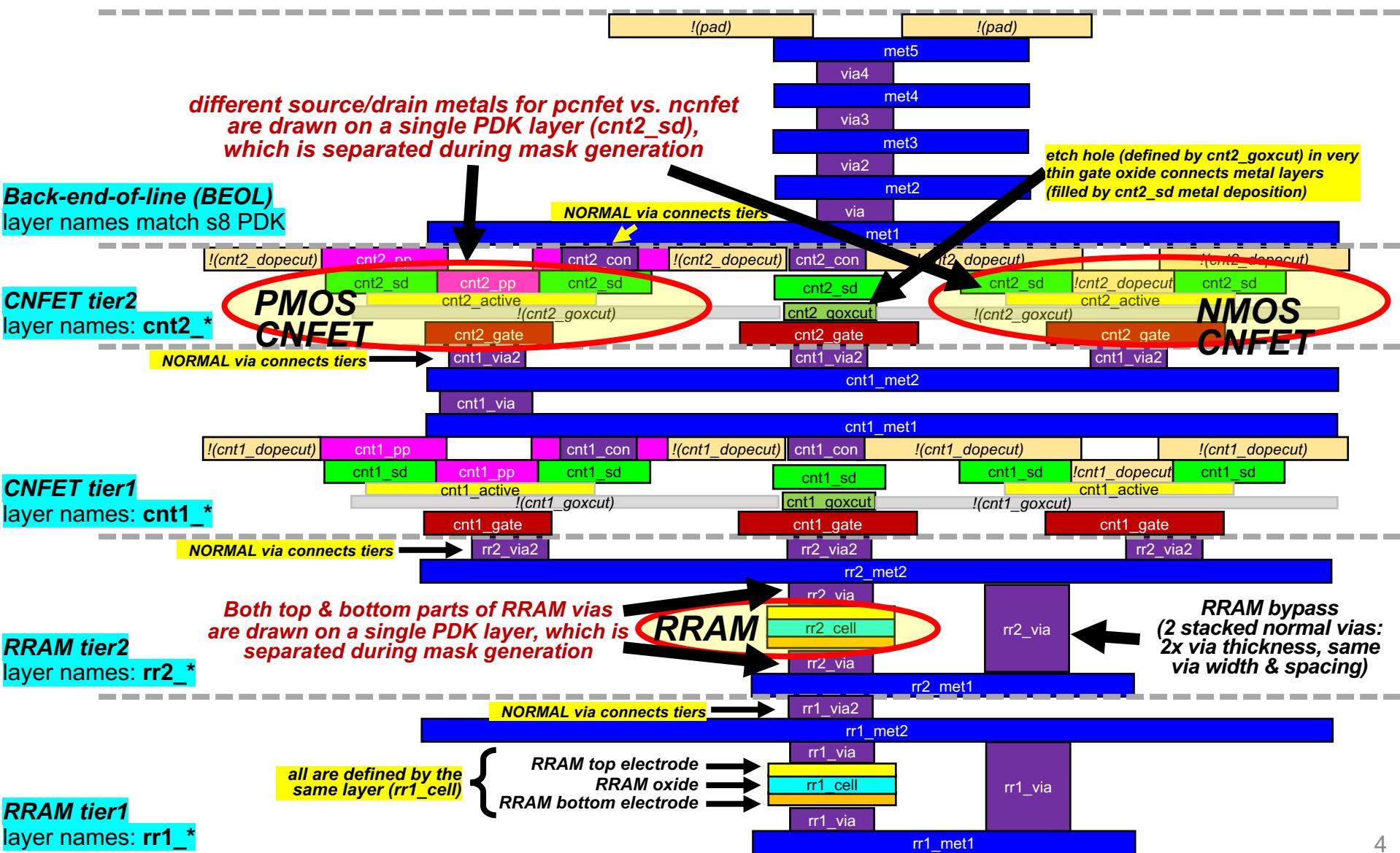
"n1" process: 2 tiers RRAM + 2 tiers CNFET

mask layers (corresponding to physical photomasks)



"n1" process: 2 tiers RRAM + 2 tiers CNFET

drawn layers (used by the designer)



PDK Layer Descriptions

- “drawn layers” are used by the designer, and are ultimately converted to “mask layers” that correspond to physical photomasks

PDK layer name (n1)	Fab acronym	Drawn layer	Mask layer	Expanded name	Function
rr1_metal1	R1BM	✓	✓	RRAM tier1 metal1	defines first level of metal interconnects for RRAM tier1, underneath RRAM cells
rr1_via	-	✓	-	RRAM tier1 via	defines both top & bottom contacts to RRAM tier1 cell (when enclosed by rr1_cell), or between rr1_metal1 and rr1_metal2
rr1_via_bot	R1BVM	-	✓	RRAM tier1 via bottom	defines bottom part of contacts to RRAM tier1 cell (when enclosed by rr1_cell), or between rr1_metal1 and rr1_metal2
rr1_cell	R1RRM	✓	✓	RRAM tier1 cell	defines RRAM cell between rr1_metal1 and rr1_metal2
rr1_via_top	R1CVM	-	✓	RRAM tier1 via top	defines top part of contacts to RRAM tier1 cell (when enclosed by rr1_cell), or between rr1_metal1 and rr1_metal2
rr1_metal2	R1TM	✓	✓	RRAM tier1 metal2	defines second level of metal interconnects for RRAM tier1, above RRAM cells
rr1_via2	R1TVM	✓	✓	RRAM tier1 via2	defines contact between rr1_metal2 and rr2_metal1
rr2_metal1	R2BM	✓	✓	RRAM tier2 metal1	defines first level of metal interconnects for RRAM tier2, underneath RRAM cells
rr2_via	-	✓	-	RRAM tier2 via	defines both top & bottom parts of contacts to RRAM tier2 cell (when enclosed by rr2_cell), or between rr2_metal1 and rr2_metal2
rr2_via_bot	R2BVM	-	✓	RRAM tier2 via bottom	defines bottom part of contacts to RRAM tier2 cell (when enclosed by rr2_cell), or between rr2_metal1 and rr2_metal2
rr2_cell	R2RRM	✓	✓	RRAM tier2 cell	defines RRAM cell between rr2_metal1 and rr2_metal2
rr2_via_top	R2CVM	-	✓	RRAM tier2 via top	defines top part of contacts to RRAM tier2 cell (when enclosed by rr2_cell), or between rr2_metal1 and rr2_metal2
rr2_metal2	R2TM	✓	✓	RRAM tier2 metal2	defines second level of metal interconnects for RRAM tier1, above RRAM cells
rr2_via2	R2TVM	✓	✓	RRAM tier2 via2	defines contact between rr1_metal2 and rr2_metal1
cnt1_gate	C1GM	✓	✓	CNFET tier1 gate	defines CNFET tier1 gates, interconnects and resistors
cnt1_goxcut	C1GCM	✓	✓	CNFET tier1 gate oxide cut	defines contact between CNFET source/drain (cnt1_sd) and CNFET back gate (cnt1_gate) through gate oxide
cnt1_active	C1ISOM	✓	✓	CNFET tier1 active	defines active regions of CNFETs on CNFET tier1
cnt1_sd	-	✓	-	CNFET tier1 source/drain	defines local interconnect to CNFET tier1 source/drain, mapped to cnt1_sdp and cnt1_sdn during mask generation
cnt1_sdp	C1PCM	-	✓	CNFET tier1 PMOS source/drain	defines local interconnect to CNFET tier1 PMOS source/drain
cnt1_pp	C1NM	✓	✓	CNFET tier1 PMOS passivation	defines passivation oxide over CNFET tier1 PMOS
cnt1_sdn	C1NCM	-	✓	CNFET tier1 NMOS source/drain	defines local interconnect to CNFET tier1 NMOS source/drain
cnt1_dopecut	C1N2M	✓	✓	CNFET tier1 doping oxide cut	defines etch regions in the doping oxide that electrostatically dopes CNFET tier1 NMOS
cnt1_con	C1CVM	✓	✓	CNFET tier1 via to source/drain	defines contact between CNFET tier1 source/drain and CNFET tier1 met1
cnt1_metal1	C1MS1M	✓	✓	CNFET tier1 metal1	defines first level of metal interconnects on CNFET tier1, above CNFET source/drain
cnt1_via	C1VS1M	✓	✓	CNFET tier1 via	defines contact between CNFET tier1 met1 and CNFET tier1 met2
cnt1_metal2	C1MS2M	✓	✓	CNFET tier1 metal2	defines second level of metal interconnects on CNFET tier1, above CNFET source/drain
cnt1_via2	C1VS2M	✓	✓	CNFET tier1 via2	defines contact between cnt1_metal2 (below) to cnt2_gate (above)
cnt2_gate	C2GM	✓	✓	CNFET tier2 gate	defines CNFET tier2 gates, interconnects and resistors
cnt2_goxcut	C2GCM	✓	✓	CNFET tier2 gate oxide cut	defines contact between CNFET source/drain (cnt2_sd) and CNFET back gate (cnt2_gate) through gate oxide
cnt2_active	C2ISOM	✓	✓	CNFET tier2 active	defines active regions of CNFETs on CNFET tier2
cnt2_sd	-	✓	-	CNFET tier2 source/drain	defines local interconnect to CNFET tier2 source/drain, mapped to cnt2_sdp and cnt2_sdn during mask generation
cnt2_sdp	C2PCM	-	✓	CNFET tier2 PMOS source/drain	defines local interconnect to CNFET tier2 PMOS source/drain
cnt2_pp	C2NM	✓	✓	CNFET tier2 PMOS passivation	defines passivation oxide over CNFET tier2 PMOS
cnt2_sdn	C2NCM	-	✓	CNFET tier2 NMOS source/drain	defines local interconnect to CNFET tier2 NMOS source/drain
cnt2_dopecut	C2N2M	✓	✓	CNFET tier2 doping oxide cut	defines etch regions in the doping oxide that electrostatically dopes CNFET tier2 NMOS
cnt2_con	C2CVM	✓	✓	CNFET tier2 via to source/drain	defines contact between CNFET tier2 source/drain and met1
met1	-	✓	-	metal1	defines first level of metal interconnects on CNFET tier2, above CNFET source/drain
via	-	✓	-	via	defines contact between met1 and met2
met2	-	✓	-	metal2	defines second level of metal interconnects on CNFET tier2, above CNFET source/drain
via2	-	✓	-	via2	defines contact between met2 and met3
met3	-	✓	-	metal3	defines third level of metal interconnects on CNFET tier2, above CNFET source/drain
via3	-	✓	-	via3	defines contact between met3 and met4
met4	-	✓	-	metal4	defines fourth level of metal interconnects on CNFET tier2, above CNFET source/drain
via4	-	✓	-	via4	defines contact between met4 and met5
met5	-	✓	-	metal5	defines fifth level of metal interconnects on CNFET tier2, above CNFET source/drain
pad	-	✓	-	pad	defines openings in the passivation
cmml1	C2MS1M	-	✓	mask layer for metal1	mask layer for metal1
cviam	C2VS1M	-	✓	mask layer for via	mask layer for via
cmm2	C2MS2M	-	✓	mask layer for metal2	mask layer for metal2
cviam2	C2VM1M	-	✓	mask layer for via2	mask layer for via2
cmm3	C2MM1M	-	✓	mask layer for metal3	mask layer for metal3
cviam3	C2VM2M	-	✓	mask layer for via3	mask layer for via3
cmm4	C2MM2M	-	✓	mask layer for metal4	mask layer for metal4
cviam4	C2VL1M	-	✓	mask layer for via4	mask layer for via4
cmm5	C2ML1M	-	✓	mask layer for metal5	mask layer for metal5
cpdm	PDM	-	✓	mask layer for pad	mask layer for pad
areaaid	-	✓	-	area identification	defines various identification regions for specialized purposes
capacitor	-	✓	-	capacitor	defines boundaries of capacitors
prBoundary	-	✓	-	place-and-route boundary	defines boundaries of place & route objects

 Total: 42 mask layers

PDK Install & Tool Dependencies

- PDK & libraries released in a single package:
 - n1_top.release_2020_08_03.tar.gz
- Access to the following electronic design automation (EDA) tools is assumed:
 - Cadence: Virtuoso®, Spectre®, Quantus QRC®, Genus®, Innovus®
 - Mentor Graphics: Calibre®
- To install the PDK and get started quickly (in Cadence Virtuoso®):

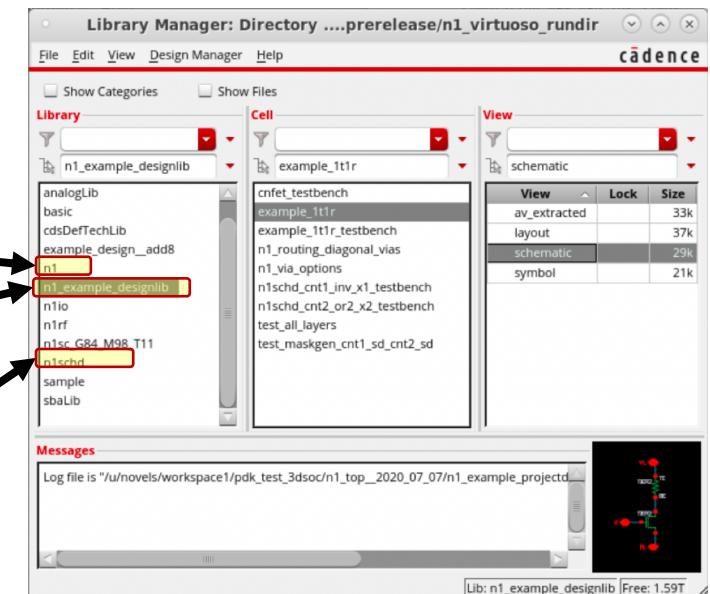
```
>> tar -zxvf n1_top.release_2020_08_03.tar.gz
>> cd n1_top
>> ./runme_extract_all
>> setenv PDK_HOME $PWD/n1/VLATEST
>> cd $PDK_HOME/../../n1_example_projectdir/VLATEST/n1_virtuoso_rundir
>> virtuoso &
```
- This will bring up the Library Manager in Cadence Virtuoso®:
 - Enjoy!

Library

n1: base technology library with CNFETs & RRAM

n1_example_designlib: reference designs

n1schd: standard cell library (high density version)

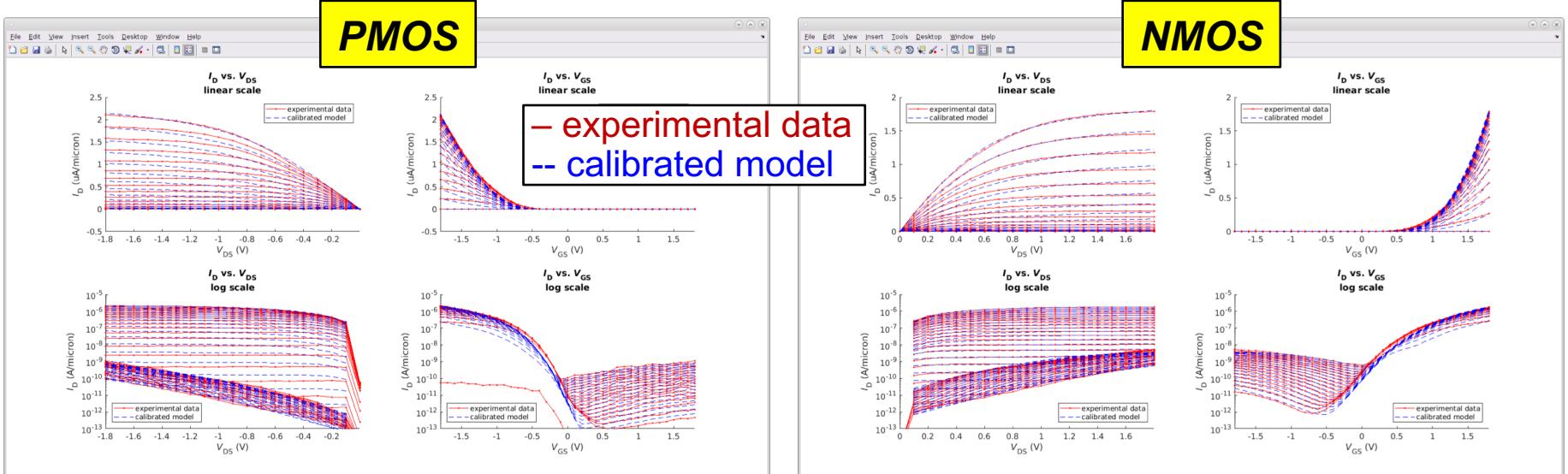


PDK Directory Structure

- Key locations are provided below, directories are **bold**, descriptions are *italicized*
- **n1_top**: top-level directory
 - **n1**: base PDK and technology library
 - **VLATEST**: symbolic link to latest version (V1.2.0.prerelease)
 - **doc**: documentation
 - **DRC**: rules and runsets for Design Rule Check using Mentor Graphics Calibre®
 - **innovus**: contains n1.innovus.layermap file to stream out GDS from Cadence Innovus®
 - **lef**: contains n1.tech.lef (technology .lef file)
 - **LVS**: rules and runsets for Layout vs. Schematic using Mentor Graphics Calibre®
 - **MODELS**: contains CNFET and RRAM compact models calibrated to experimental data
 - **PEX**: contains files for Parasitic Extraction using Quantus QRC (e.g., qrcTechFile)
 - **VirtuosoOA**: for use with Cadence Virtuoso®
 - **libs**: contains technology library "n1" with CNFET and RRAM cells
 - **techfiles**: contains n1.tf technology file defining parameters for Cadence Virtuoso®
 - **n1sc**: standard cell libraries
 - **VLATEST**: symbolic link to latest version (V1.2.0.prerelease)
 - **cdl**: directory with Circuit Description Language netlists
 - **hne**: netlists for all cells (no parasitics)
 - **pex**: parasitic extraction netlists for all cells
 - **cells**: contains files listing all cells (for use with various tools and scripts)
 - **lef**: contains abstract cell views (.macro.lef)
 - **lib**: contains liberty files (.lib) with timing/power information
 - **verilog**: contains functional Verilog files (.v) for all cells
 - **VirtuosoOA**: for use with Cadence Virtuoso®
 - **libs**: n1schd & n1sc_G84_M98_T11 libraries, with cell views for all cells: layout, schematic, functional, av_extracted, symbol, abstract
 - **n1sc**: input/output libraries
 - **VLATEST**: symbolic link to latest version (V1.2.0.prerelease)
 - **lef**: contains abstract cell views (.macro.lef)
 - **VirtuosoOA**: for use with Cadence Virtuoso®
 - **libs**: n1io library with output pad
 - **n1_example_projectdir**: example designs for DRC, LVS, PEX, simulation, synthesis, place & route
 - **VLATEST**: symbolic link to latest version (V1.2.0.prerelease)
 - **example_designs**: example Verilog designs with scripts for synthesis, place-and-route, and importing into Virtuoso®
 - **add8**: 8-bit adder for quick synthesis & design flow testing
 - **n1_virtuoso_rundir**: run Cadence Virtuoso® from this directory, paths for libraries and PEX are setup here
 - **VirtuosoOA**: for use with Cadence Virtuoso®
 - **libs**: reference designs (layouts, schematics) in Cadence Virtuoso® for running DRC, LVS, PEX, simulation

CNFET Calibrated Model

- Experimental data + calibration



calibration using MIT virtual source compact model equations for field-effect transistors

- Verilog-A module with SPICE/spectre wrapper interface
 - 4 combinations: nmos/pmos on tier1/tier2

```

File Edit View Terminal Help
File Edit Options Buffers Tools Help
simulator lang=spectre insensitiveviews
Mate Terminal

section tt
  include "cnfet3_params.scs" section=tt
  include "cnfet3.scs" section=nmos
endsection tt

section mos
andl_include "cnfet3.va"

// -----
// ----- NMOS -----
subckt cnfet3 ( D G S )
parameters
  // give user control over the following parameters
  w           = 1.000          // transistor width [microns]
  lch         = 0.280          // channel length [microns]
  lc          = 0.280          // contact length [microns]
  lun         = 0.030          // gate underlap length under contact [microns]
  wgeext      = 0.030          // width extension over edge [microns]
  wceext      = 0.030          // contact width edge extension over active [microns]
  lctact      = 0.030          // contact to active edge length [microns]
  + gidl_mod   = 0             // gate-induced drain leakage (gidl) mod, 0: off, 1: on
cnfet3 inst ( D G S S ) cnfet3
  FEtype      = 1
  + w         = 1.000 * w
  + lch_m     = 1e-6 * lch
  + lg_m      = 1e-6 * (lch + 2*lun)
  + lc_m      = 1e-6 * lc
  + gidl_mod  = gidl_mod
+ parameters in cnfet3_params.scs user can't override
+ mu_app_cm_per_Vs = nmos_mu_app_cm2_per_Vs
+ vxo_cm_per_s  = nmos_vxo_cm_per_s
+ tox_m        = nmos_tox_m
+ kox           = nmos_kox
+ kspacer_sd    = nmos_kspacer_sd
+ ddbl_V_per_V = nmos_ddbl_V_per_V
.UU (DOS)---File cnfet3.scs Top_L1 (Fundamental)
Loading vc-git...done

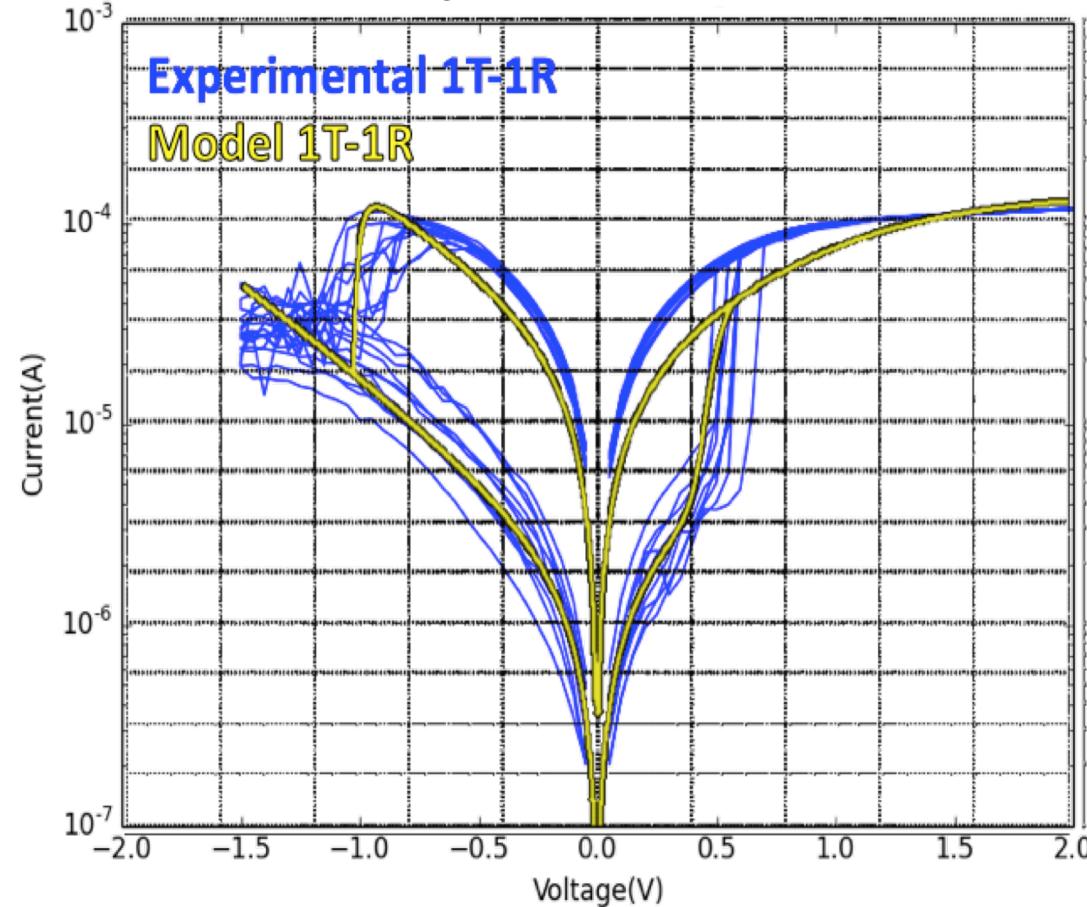
```

Cadence Spectre® interface
for circuit simulations

CNFET sub-circuit (subckt) wrappers:
CNFET is a 3-terminal device
(Drain, Gate, Source), but there are
4-terminal options (with a dummy
Bulk terminal) for compatibility with
some EDA tools

RRAM Calibrated Model

- Experimental data + calibration
 - Calibration using CNFET model for “1T”
- Verilog-A module with Cadence Spectre® wrapper interface



```
File Edit View Search Terminal Help
File Edit Options Buffers Tools Minibuf Help
simulator lang=spectre insensitive=yes

section tt
    include "rram0.scs" section=rram
endsection tt

section rram
ahdl_include "rram0.va"

// =====
// RRAM
// -----
subckt rram0 ( TE BE )
parameters
// --- device parameters
+ L      = 5e-9          // Oxide thickness (m)
+ gap_min = 0.1e-9        // Min. gap distance (m)
+ gap_max = 0.8e-9//1.7e-9 // Max. gap distance (m)
+ gap_ini = 0.1e-9        // Initial gap distance (m)
+ a0     = 0.25e-9        // Atomic distance (m)
+ Eag    = 1.501           // Activation energy for vacancy generation (eV)
+ Ear    = 1.5             // Activation energy for vacancy recombination (eV)
// --- I-V characteristics
+ I0     = 6.14e-5         // Steady-state current (A)
+ g0     = 2.7505e-10       // Conductance (S)
+ V0     = 0.43            // Threshold voltage (V)
// --- gap dynamics
+ Velo   = 150             // Velocity (m/s)
+ gamma0 = 16.5            // Gamma parameter
+ g1     = 1e-9             // Gap narrowing coefficient
+ beta   = 1.25            // Beta parameter
// --- temperature dynamics
+ T0     = 273+25          // Ambient temperature (K)
+ Cth    = 3.1825e-16        // Effective thermal capacitance (J/K)
+ Tau_th = 2.3e-10          // Effective thermal time constant (s)
// --- simulation time control
+ tstep   = 1e-9            // Max. internal timestep (s)
// --- resistance
+ Rread   = 0.1              // Read voltage (V)

rram0_inst ( TE BE ) rram0
+ L      = L
+ gap_min = gap_min
+ gap_max = gap_max
+ gap_ini = gap_ini
+ a0     = a0

-UU-(DOS)---F1 rram0.scs      Top L1      (Fundamental) -----
Find file: /gitroot/3dsoc/pdk/n1_top/n1/VLATEST/MODELS/SPECTRE/n1/Models/
```

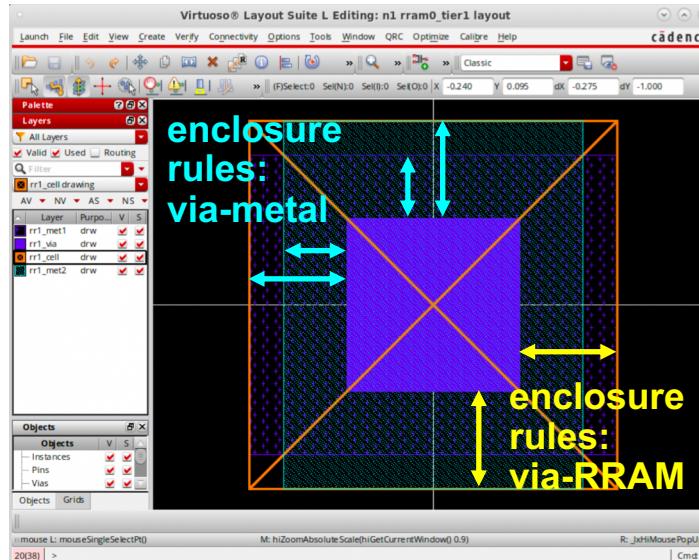
```
[novels4]: ~/gitroot/3dsoc/pdk/n1_top/n1/VLATEST/MODELS/SPECTRE/n1/Models> grep subckt rram0.scs
```

subckt rram0 (TE BE)
subckt rram0_tier1 (TE BE) } RRAM subcircuit (subckt) wrappers (tier1/tier2)
subckt rram0_tier2 (TE BE)

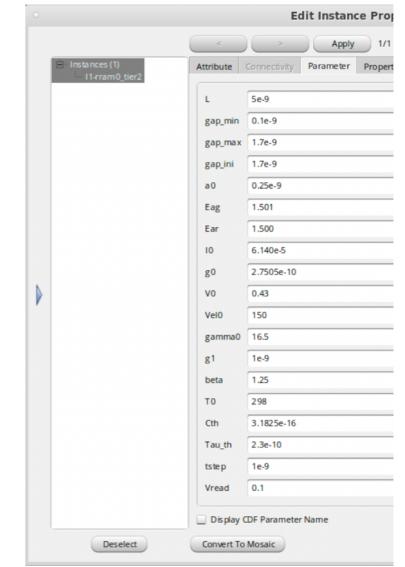
Parameterized Cells (PCells)

- Cadence Virtuoso® cells: CNFET + RRAM
- 2 RRAM cells (2 tiers)
 - rram0_tier1
 - rram0_tier2
- 4 CNFET cells (N/P, 2 tiers)
 - ncnfet3_tier1
 - pcnfet3_tier1
 - ncnfet3_tier2
 - pcnfet3_tier2
- CNFET cells include routing blockages
 - No goxcut under source/drain
 - No gate routing under source/drain
 - No via on top of NMOS source/drain

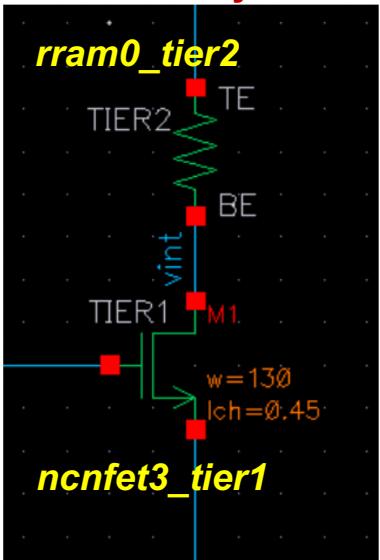
RRAM cell layout (tier 1)



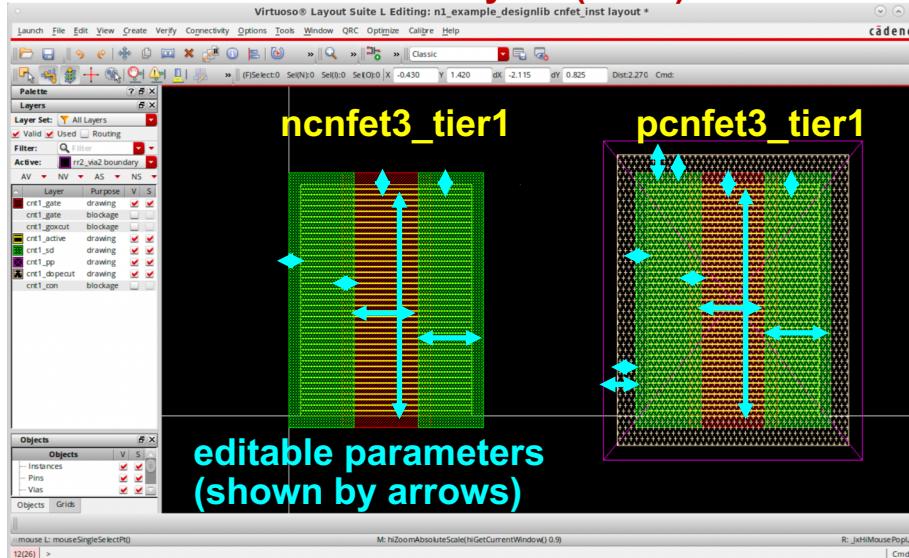
RRAM model parameters



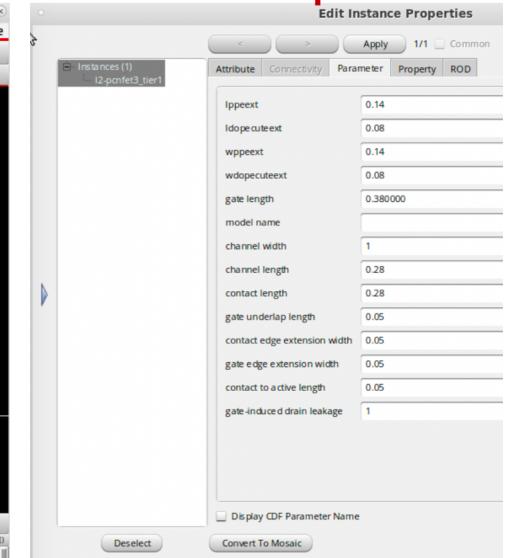
schematic symbols



CNFET PCell layout (tier 1)

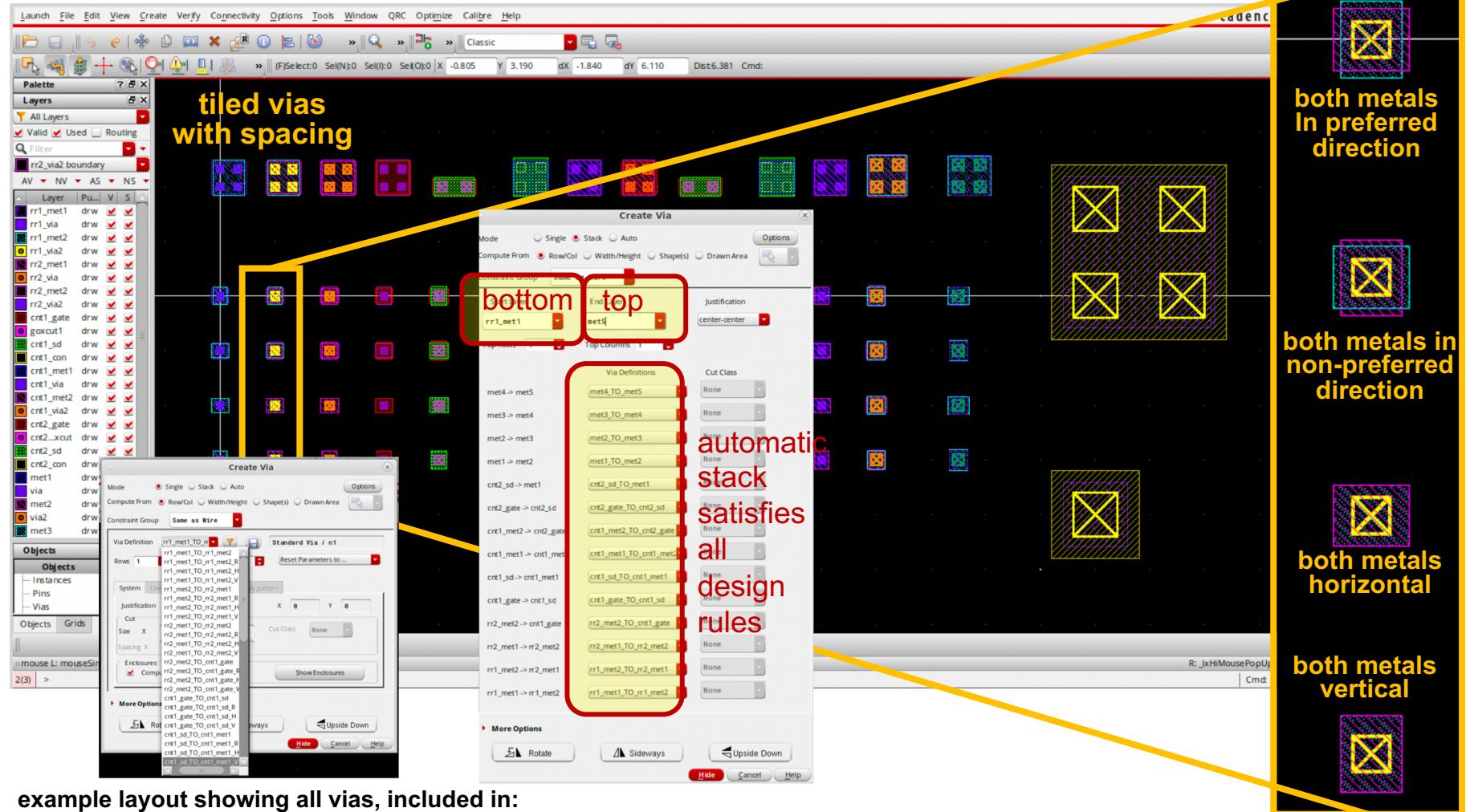


CNFET editable parameters



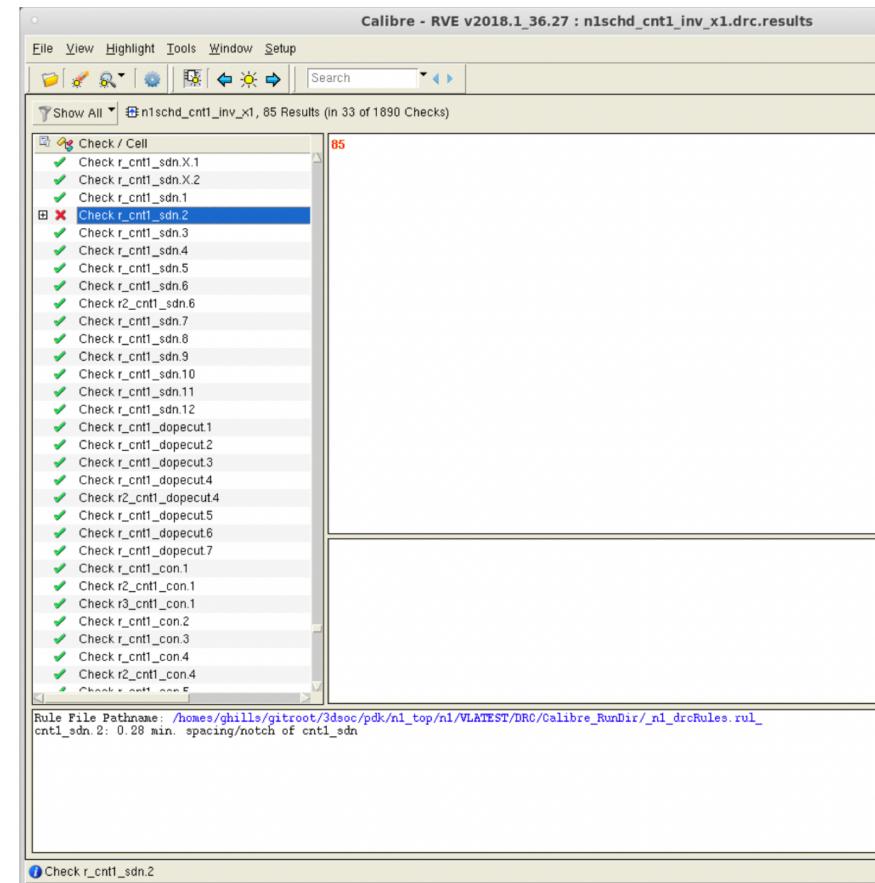
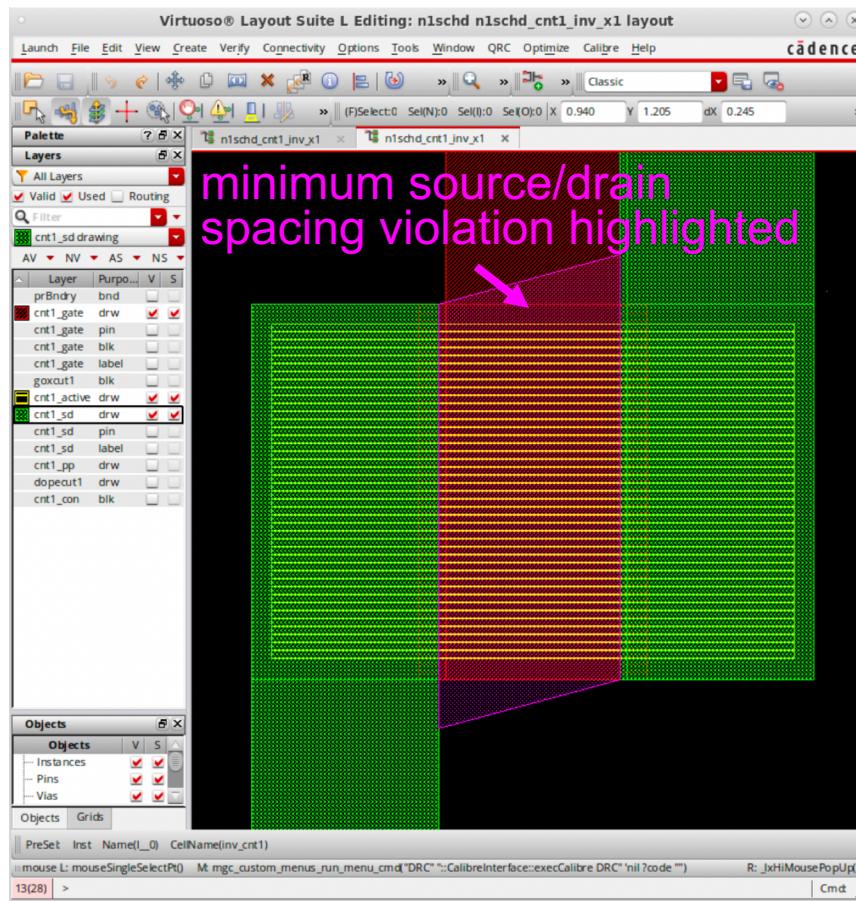
Pre-defined Vias

- Integrated in Cadence Virtuoso®, select from “Create Via” drop down menu
 - Supports “via stacking” (specify top & bottom layer)
 - Supports “via tiling” (specify number of via rows/columns)
 - All vias satisfy Design Rule Checks (DRC) (e.g., enclosure, spacing, area rules)



Design Rule Check (DRC)

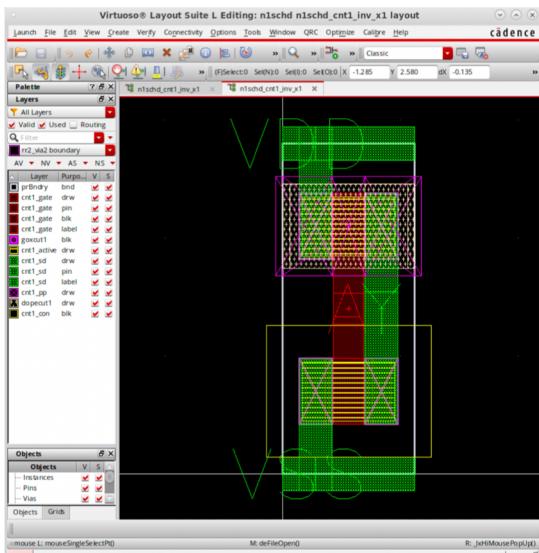
- From layout editor in Cadence Virtuoso®
 - 1) From toolbar, select “Calibre” > “Run nmDRC”
 - 2) At the prompt, load the runset file: \$PDK_HOME/DRC/Calibre/n1_drc_runset
 - 3) Click “Run DRC”
- DRC rules are defined in: \$PDK_HOME/DRC/Calibre/n1_drcRules.rul
- Example DRC violation highlighted in tools



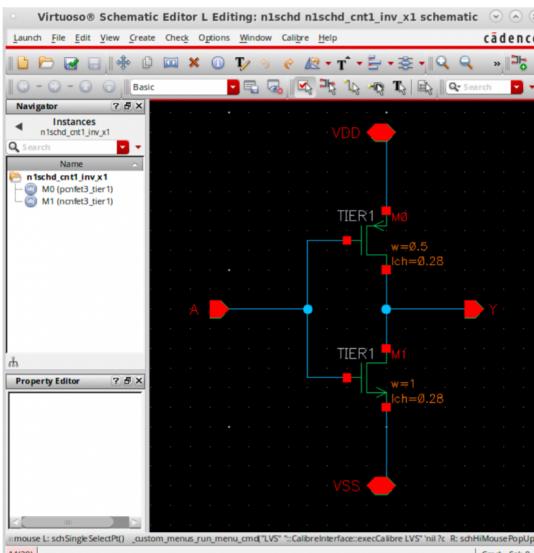
Layout vs. Schematic (LVS)

- From layout editor or schematic editor in Cadence Virtuoso®
 - 1) From toolbar, select “Calibre” > “Run nmLVS”
 - 2) At the prompt, load one of the following runset files:
 - \$PDK_HOME/LVS/Calibre/n1_lvs_runset_HNE: Hierarchical Netlist Extraction (HNE) – extract netlist from layout only (no schematic needed)
 - \$PDK_HOME/LVS/Calibre/n1_lvs_runset_LVS: Layout vs. Schematic (LVS) – compare extracted netlists from layout & schematic views
 - 3) Click “Run LVS”
- LVS rules are defined in: \$PDK_HOME/LVS/Calibre/n1_lvsRules.rul
- Example (n1schd_cnt1_inv_x1 standard library cell):

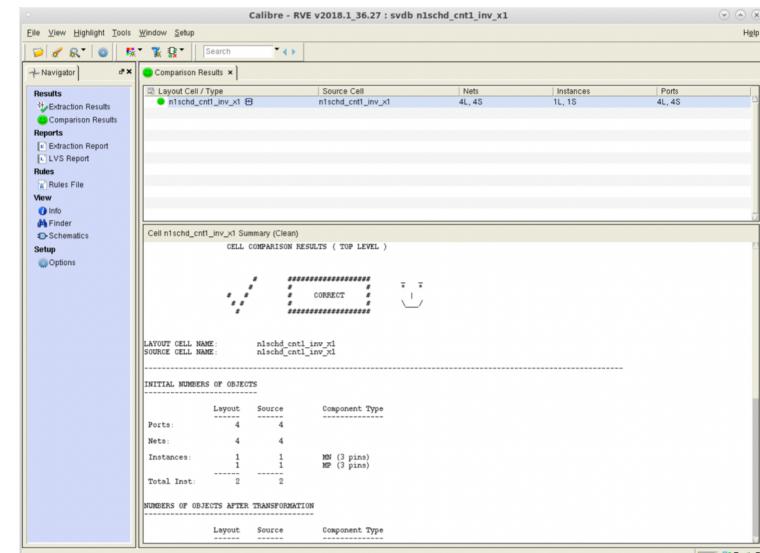
layout



schematic

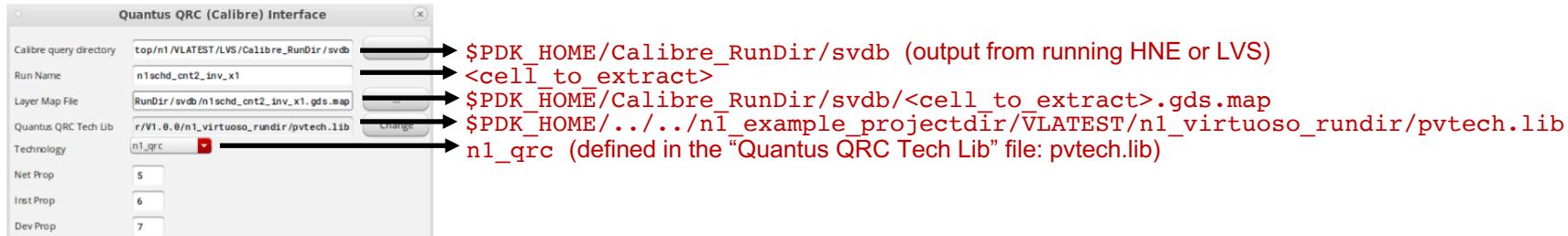


LVS Pass



Parasitic Extraction (PEX) Using GUI

- From the layout or schematic editor, select “Run nmLVS”, and use the runset:
 - \$PDK_HOME/LVS/Calibre/n1_lvs_runset_QRC
 - This will automatically run the “runQuery” utility using the HNE/LVS results to setup PEX
- From the layout editor toolbar, select “QRC” > “Run Calibre - Quantus QRC”, and fill out the form as follows:



- In the Quantus QRC form, “Setup” tab, set “Technology” to “n1_qrc” and “RuleSet” to “typical” (this loads all values in the form)
 - To generate the parasitic extraction netlist, set “Output” to “Spice” and click “Apply” (or “OK” to close GUI)
 - To generate the extracted cell view, set “Output” to “Extracted View” and click “Apply” (or “OK” to close GUI)

Output = “Spice”

```
File Edit View Search Terminal Help
File Edit Options Buffers Tools Help
SUBCKT n1schd_cnt2_inv_x1 A VSS VDD Y
*
* caps2d version: 10
*
* TRANSISTOR CARDS
*
MM0 Y#2 A#1 VDD#1 pcfet3_tier2 L=0.34 W=0.5
+ lch=0.28 lun=0.03 lc_s=0.25 lc_d=0.25 gidl_mod=1 nf=1
MM1 Y#4 A#3 VSS#1 ncfet3_tier2 L=0.34 W=0.5
+ lch=0.28 lun=0.03 lc_s=0.25 lc_d=0.25 gidl_mod=1 nf=1
*
* RESISTOR AND CAP/DIODE CARDS
*
Rg1 A#1 A 2.4654 $cnt2_gate SW=0.26
Rg2 A A#3 2.4654 $cnt2_gate SW=0.26
RF_2_1 VDD#1 VDD#2 0.2488 $cnt2_sd SW=0.5
RF_2_2 Y#1 Y#2 0.2488 $cnt2_sd SW=0.5
RF_1_1 Y#5 VSS#2 0.2488 $cnt2_sd SW=0.5
RF_1_2 Y#3 Y#4 0.2488 $cnt2_sd SW=0.5
RF1 Y Y#3 2.4975 $cnt2_sd SW=0.28
RF2 Y Y#1 2.4975 $cnt2_sd SW=0.28
RF3 VDD#2 VDD 2.5571 $cnt2_sd SW=0.28
RF4 VSS#2 VSS 2.5571 $cnt2_sd SW=0.28
*
* CAPACITOR CARDS
*
C1 Y A 1.38587E-16
C2 VSS VDD 3.21935E-18
C3 VSS Y#3 2.67484E-17
C4 VSS VDD 1.66119E-17
```

Quantus QRC form

Output = “Extracted View”

cell view: “av_extracted”

GUI = “Graphical User Interface”

Parasitic Extraction (PEX) Using GUI: Option 2

1) From the layout or schematic editor, select “Run nmLVS”, and use the runset:

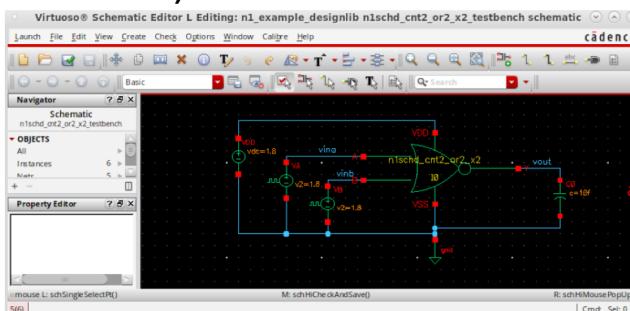
- \$PDK_HOME/LVS/Calibre/n1_lvs_runset_runQueryAndQRC
 - a) This will automatically run the “runQuery” utility using the HNE/LVS results to setup PEX
 - b) Then it will automatically run QRC using the template files in \$PDK_HOME/PEX/QRC
 - c) (see \$PDK_HOME/PEX/QRC/runQueryAndQRC for details)
 - d) This will generate the extracted view (av_extracted) and the extracted netlist, in
./calibre_rundir/pex/<cell_name>.pex.sp

Circuit Simulations with Parasitics

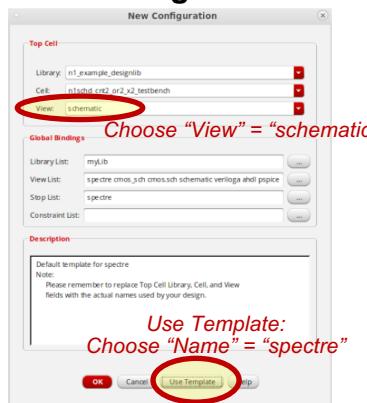
- To run simulations on the extracted cell view in Cadence ADE®:

- 1) Ensure the cell has the following views (or create them): schematic, symbol, av_extracted (e.g., n1schd_cnt2_or2_x1)
- 2) Create a new cell schematic view (the testbench) to instantiate the cell to test (e.g., n1schd_cnt2_or2_x1_testbench)
 - Setup sources (e.g., voltage sources) to stimulate the cell
- 3) Launch ADE and run simulations on the schematic (this runs the simulation *without* parasitics)
- 4) Create another cell view: “config” for the testbench
 - a) In the “New Configuration” form, select “View” = “schematic”, then click “Use Template”, choose “Name” = “spectre”, then click “OK”
 - b) In the “Virtuoso® Hierarchy Editing” form, right click the cell to test, to change the view from “schematic” to “av_extracted”, then save
- 5) In the ADE window, now load the config view by selecting “Setup” > “Design”, then change “View Name” from “schematic” to “config”
- 6) Re-run the simulation, and it will use the av_extracted view (e.g., the delay is longer in the simulation traces below)

2) testbench schematic

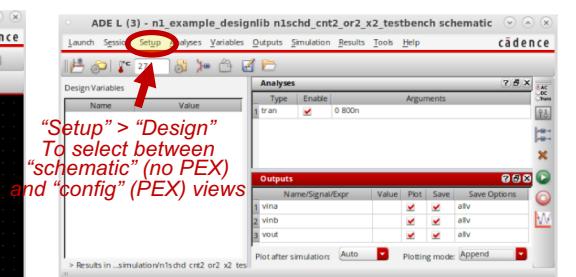


4a) “New Configuration” form



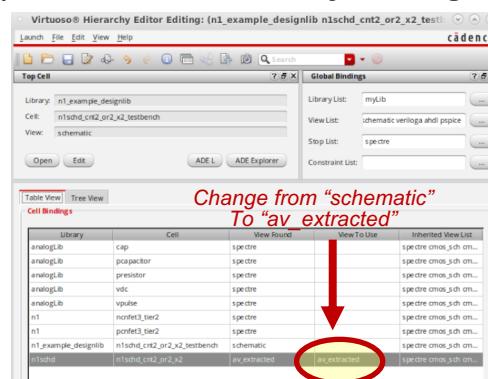
Use Template:
Choose “Name” = “spectre”

3) ADE to run simulations

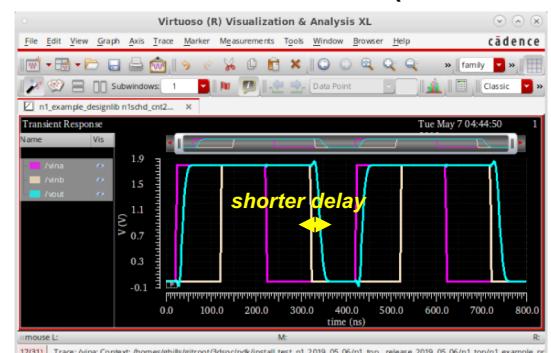


*NOTE: ensure in “Setup” > “Simulator/Directory/Host”, that “Simulator” = “spectre” (and not “ams”)

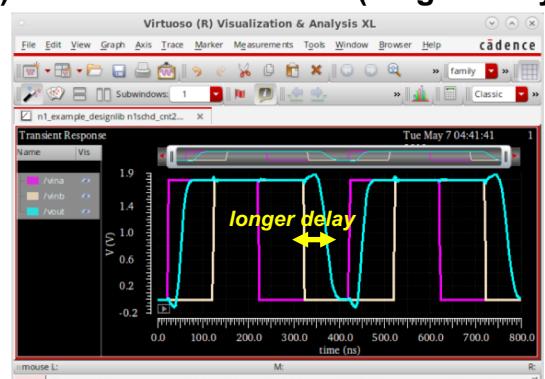
4b) “Virtuoso® Hierarchy Editing” form



3) simulations without PEX (shorter delay)



6) Simulations with PEX (longer delay)



Example Designs

- Example designs are provided in the Virtuoso® library “n1_example_designlib”. These are provided as examples to run DRC, LVS, PEX, simulations, and to view example layouts and schematics

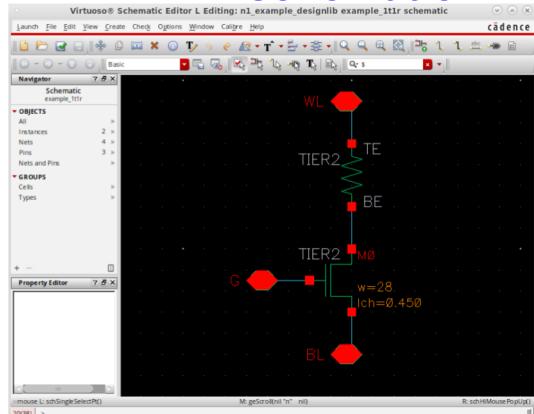
cell (in library n1_example_designlib)	description
cnfet_testbench	schematics and simulations to exercise the CNFET compact model (with different channel lengths as an example). Simulation setups for both DC analysis (I-V) and AC analysis (C-V) are included
example_1t1r	example 1T-1R memory cell using 1 CNFET and 1 RRAM. Cell views include layout and schematic (for LVS) and av_extracted for parasitic simulation
example_1t1r_testbench	testbench setup to illustrate switching of the RRAM state by modulating the bias voltages across the 1T-1R structure. Simulation setup for transient analysis is included.
n1_routing_diagonal_vias	example layout to show metal routing on all layers, including considerations for metal spacing considering via enclosures
n1schd_cnt1_inv_x1_testbench	testbench setup to measure propagation delay of the library cell n1schd_cnt2_inv_x1, with and without parasitic extraction. Simulation setup for transient analysis is included.
n1schd_cnt2_or2_x2_testbench	testbench setup to measure propagation delay of the library cell n1schd_cnt2_or2_x2, with and without parasitic extraction. Simulation setup for transient analysis is included.
test_all_layers	example circuit layout with all drawn layers (e.g., to test netlist extraction and mask generation)
test_maskgen_cnt1_sd_cnt2_sd	example circuit layout to test mapping of the drawn CNFET source/drain layers (cnt*_sd) into the mask CNFET source/drain layers (cnt*_sdp for PMOS and cnt*_sdn for NMOS)

- As an example, the example design “n1schd_cnt2_or2_x2_testbench” corresponds to the cell that was created to illustrate “Circuit Simulations with Parasitics” on the previous slide
- Furthermore, all of the standard cells in the n1schd library can also be used for testing (DRC, LVS, PEX, simulation, etc.)

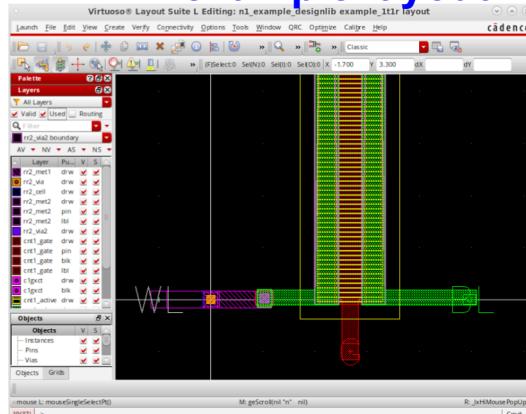
Example Design: 1T-1R (CNFET-RRAM)

- As described in the table on the previous slide, the cell “example_1t1r” includes simulations showing programming of a 1T-1R memory cell using 1 CNFET and 1 RRAM, with the following cell views:

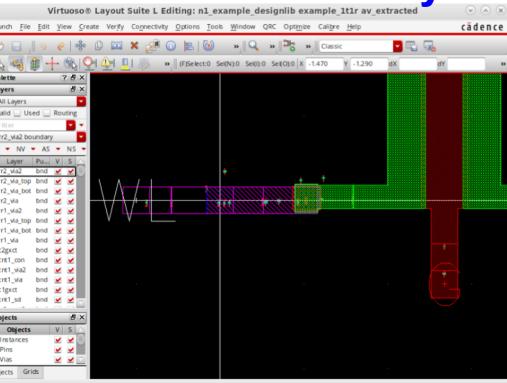
1T-1R schematic



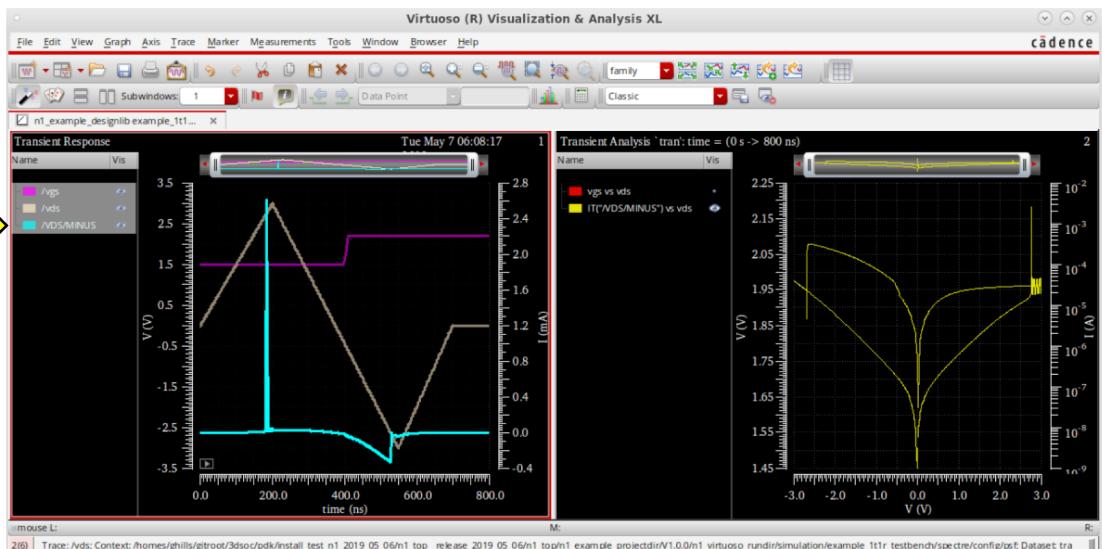
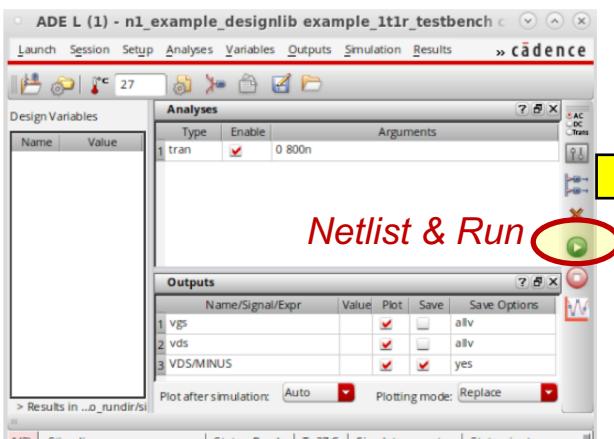
1T-1R example layout



**parasitic extracted view
(R & C annotated on layout)**

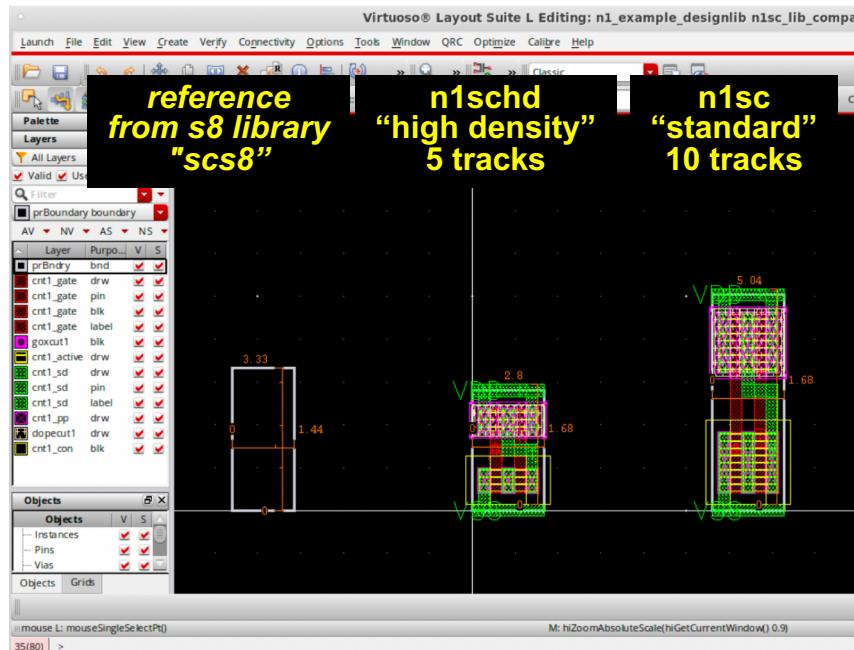


- Use the “example_1t1r_testbench” to run simulations
 - Open the “sim_tran_pex” cell view to open the simulation state in ADE, then click “Netlist and Run”
 - In the plot that opens, click “Axis” > “Y vs. Y”, then select the trace as “/vds”. This will plot the current through the 1T-1R cell as a function as the voltage across it. Change the axis to “log scale” to see the familiar RRAM current vs. voltage plot (shown on the right)



Standard Cell Libraries

- 3 libraries offered
 - **n1schd: high density**, local interconnect above and below CNFET gate/source/drain
 - 5 tracks, CGP/MP = 0.56 um (site size = $0.56 * 2.8 = 1.568$)
 - (reference: “scs8” library from s8: site size = $0.48 * 3.33 = 1.5984$)
 - **n1sc**: local interconnect above CNFET gate/source/drain
 - 10 tracks, CGP/MP = 0.56 um (site size = $0.56 * 5.04 = 2.8224$)
- >300 total cells per library (multiple functions, CNFET tiers, drive strengths, circuit topologies)
- Example layout (nand2) for each library:



- Liberty timing/power files (.lib) located in: \$PDK_HOME/.../.../n1sc/VLATEST/lib
- Abstract views (.lef) located in: \$PDK_HOME/.../.../n1sc/VLATEST/lef
- Parasitic extraction (PEX) netlists (.pex.sp) located in: \$PDK_HOME/.../.../n1sc/VLATEST/cdl/pex
- Schematic netlists without parasitics (.hne.sp) located in: \$PDK_HOME/.../.../n1sc/VLATEST/cdl/hne

Standard Cell Libraries: Table of All Cells

- n1schd = high density library, n1sc_G84_M98_T11 = low leakage standard cell library
- n1schd: all cells offered on both CNFET tier 1 (n1schd_cnt1_*) and CNFET tier 2 (n1schd_cnt2_*)
 - 302 total cells (25 functions, 2 tiers, multiple drive strengths & circuit topologies)
 - all cells include cell views: layout, schematic, symbol, functional, abstract, av_extracted
- Same for n1sc_G84_M98_T11

Total Functions = 25			Drive strength options											
Function	Circuit topology mod	Name format	x1	x2	x3	x4	x5	x6	x8	x12	x16	x20	x24	x32
inverter		n1schd_cnt*_inv_x*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
tri-state inverter (active low)		n1schd_cnt*_einvn_x*	✓	✓			✓							
buffer		n1schd_cnt*_buf_x*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2-input nand	nand-based	n1schd_cnt*_nand2_x*	✓	✓	✓	✓	✓							
	nor-based	n1schd_cnt*_nand2nr2_x*	✓	✓										
2-input nor	nor-based	n1schd_cnt*_nor2_x*	✓	✓		✓	✓							
	nand-based	n1schd_cnt*_nor2nd2_x*	✓	✓										
2-input and	nand-based	n1schd_cnt*_and2_x*	✓	✓										
	nor-based	n1schd_cnt*_and2nr2_x*	✓	✓										
2-input or	nor-based	n1schd_cnt*_or2_x*	✓	✓										
	nand-based	n1schd_cnt*_or2nd2_x*	✓	✓										
2-input exclusive or	nand-based	n1schd_cnt*_xor2nd2_x*	✓	✓										
	nor-based	n1schd_cnt*_xor2nr2_x*	✓	✓										
2-input exclusive nor	nand-based	n1schd_cnt*_xnor2nd2_x*	✓	✓										
	nor-based	n1schd_cnt*_xnor2nr2_x*	✓	✓										
2-to-1 multiplexor	using transmission gates	n1schd_cnt*_mux2tg_x*	✓	✓										
	nand-based	n1schd_cnt*_mux2nd2_x*	✓	✓										
	nor-based	n1schd_cnt*_mux2nr2_x*	✓	✓										
and-or-invert (aoi21)		n1schd_cnt*_aoi21_x*	✓	✓										
or-and-invert (oai21)		n1schd_cnt*_oai21_x*	✓	✓										
3-input nand	nand-based	n1schd_cnt*_nand3_x*	✓	✓										
	nor-based	n1schd_cnt*_nand3nr3_x*	✓	✓										
3-input nor	nor-based	n1schd_cnt*_nor3_x*	✓	✓										
	nand-based	n1schd_cnt*_nor3nd3_x*	✓	✓										
full adder	nand-based	n1schd_cnt*_fand2_x*	✓	✓										
	nor-based	n1schd_cnt*_fanr2_x*	✓	✓										
half adder	nand-based	n1schd_cnt*_hand2_x*	✓	✓										
	nor-based	n1schd_cnt*_hanr2_x*	✓	✓										
fill cell		n1schd_cnt*_fill_x*	✓	✓										
decap cell		n1schd_cnt*_decap_x*			✓	✓	✓	✓	✓	✓			✓	
D-flip flop, positive edge triggered	using transmission gates	n1schd_cnt*_dffptg_x*	✓	✓			✓							
	cascaded d-latches	n1schd_cnt*_dffpdl_x*	✓	✓										
	nand-based	n1schd_cnt*_dffpn2d_x*	✓	✓										
	nor-based	n1schd_cnt*_dffpn2r_x*	✓	✓										
D-latch, active high enable	using transmission gates	n1schd_cnt*_dlptg_x*	✓	✓										
	nand-based	n1schd_cnt*_dlpnd2_x*	✓	✓										
D-latch, active low enable	using transmission gates	n1schd_cnt*_dlntg_x*	✓	✓										
	nor-based	n1schd_cnt*_dlnnr2_x*	✓	✓										
D-flip flop, with scan	using transmission gates	n1schd_cnt*_sdffptg_x*	✓	✓										
Level-sensitive scan latch, active high enable	nand-based	n1schd_cnt*_sdlpnd3_x*	✓	✓										
D-flip flop, asynchronous reset (reset low)	using transmission gates	n1schd_cnt*_dffprntg_x*	✓	✓										
Clock gating D-latch, positive edge triggered		n1schd_cnt*_cgdlptg_x*	✓	✓										

Synthesis, Place & Route

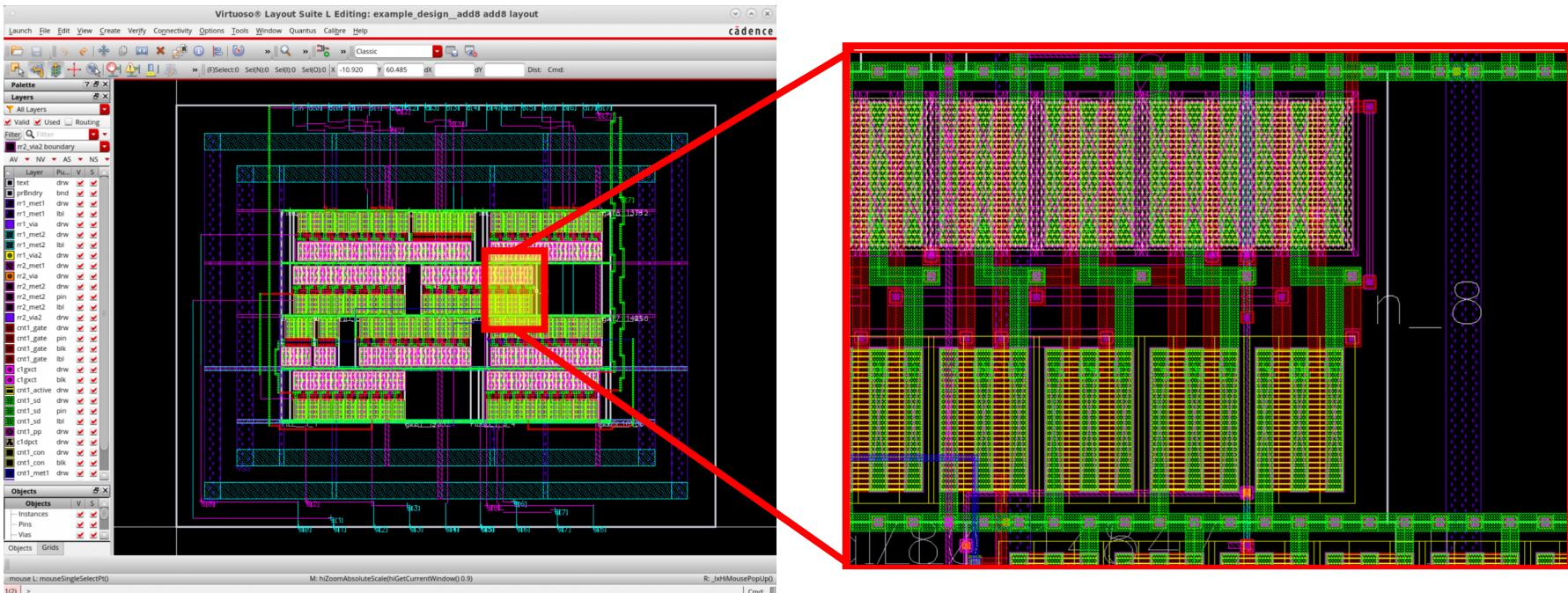
- Synthesis & place-and-route scripts are in:
`$PDK_HOME/../../n1_example_projectdir/VLATEST/example_designs/add8/syn_pnr`
- To run synthesis, run "make syn"
 - This will generate the synthesized netlist: `output_genus/add8.syn.v`
- To run place-and-route, run "make pnr"
 - This will generate the stream file: `output_innovus/add8.gds`
- Finally, run "make gds", to copy the .gds file into a location where it will be imported into virtuoso as library "example_design_add8"

file (in syn directory)	description
<code>n1_setup.tcl</code>	Setup variables common to both synthesis & place-and-route scripts
<code>n1_genus.unconstrained.tcl</code>	synthesis script (run in Cadence genus)
<code>n1_innovus.tcl</code>	place-and-route script (run in Cadence innovus)
<code>n1_innovus_ports.add8.tcl</code>	called by <code>ndk_innovus.tcl</code> to place pins on the border of the processor core (inputs on the top edge, outputs on the bottom edge)
<code>n1_view_definition.genus.no_sdc.tcl</code>	setup script for analysis views for genus
<code>n1_view_definition.innovus.qrc.tcl</code>	setup script for analysis views for innovus
<code>_reference_output</code>	directory containing reference output files from running scripts on my machine
<code>Makefile</code>	Makefile interface to invoke various design steps

- Note that, "make nuke" will remove all the generated design files, "make clean" will remove all of the intermediate files but will keep the necessary design files

Import design: Innovus® to Virtuoso®

- After running “make syn”, “make pnr”, and “make gds” (described on the previous slide), from the same directory, run:
 - run “make layout” to stream in the gds
 - run “make schematic” to import the Verilog netlist
 - run “make drc” to run DRC on the layout (the number of violations will be printed to stdout)
 - optional: run “make hne” to run HNE to extract the netlist from the layout (without performing LVS)
 - run “make lvs” to run LVS on the imported layout vs. imported schematic (the LVS result will be printed to stdout)
 - You can also run “make virtuoso” to open virtuoso from this directory and view the design “add8_example”
- NOTE: to remove all the files and then re-run everything from scratch, you can run “make nuke” (to remove everything), followed by “make all” (which runs “make syn pnr gds layout schematic drc hne lvs”)



GDS = database format for IC design

Questions

- Email support at SkyWater
- Release notes for `n1_top.release_2020_08_03.tar.gz`
 - Dense standard cell layouts in `n1schd` can cause routing congestion & DRC violations, the other included standard cell library (`n1sc_G84_M98_T11`) can be used when routing congestion is high
- Updates for release v1.2.0 in `release_2020_08_03`
 - Added “lvscap” purposes for many layers indicating that LVS should extract a capacitor from the layout (similar to “short” purpose that extracts a resistor). Corresponding updates in LVS rules
 - Added “run_drc” and “run_lvs” scripts to run DRC & LVS in batch mode from command line
- Updates for release v1.2.0.prerelease in `release_2020_07_07`
 - Updated CNFET calibrated compact model (higher current density vs. previous release)
 - Added `n1sc_G84_M98_T11` standard cell library (to replace `n1sc`)
 - Updated `qrcTechFile` for use with Cadence Innovus versions 19.1 (`qrcTechFile.pr` not needed)
- Updates for release v1.0.2 in `release_2019_05_21`
 - Added `n1sc` standard cell library (in addition to `n1schd`)
 - Added `qrcTechFile.pr` for use with Cadence Innovus versions 17.1 and 18.1
 - Fixed timing arcs for D-latches
 - Updated ncnfet3_tier1 label update in symbol view (“TIER1” instead of “TIER2”)
 - Fixed DRC violations in cells `n1schd_cnt*_mux2tg_x2`, `n1schd_cnt*_mux2tg_x4`
 - Fixed automatic spacing for `cnt*_goxcut` layers for tiled vias in Cadence Virtuoso®
- Original release: v1.0.2 in `release_2019_05_21`