

# Analog Flow PDK 28FDSOI

M2- Layout Verification

**FD-SOI**

CMOS & Derivative PDK



M0. Design Kit Overview

M1. Schematic and Simulation

M2. Layout Verification

M3. Post Layout Simulation

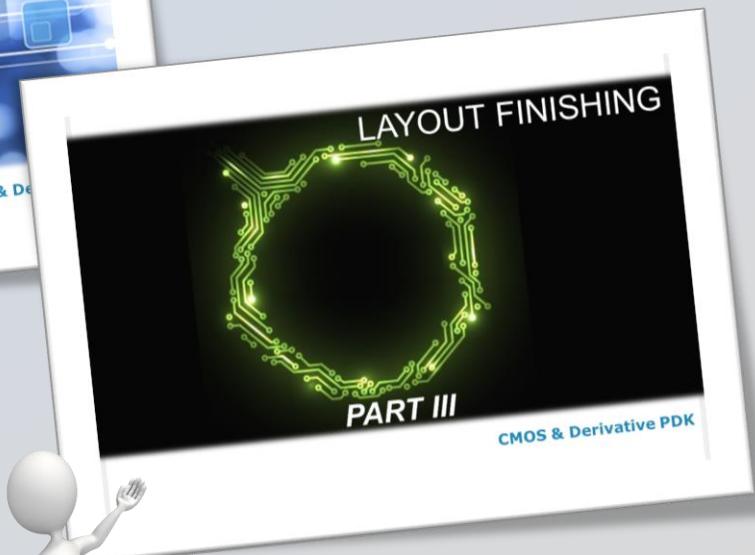
M4. Wrap-up & Conclusion



- Identify the sign-off layout verification flow
- Configure the CAD tools for DRC, LVS
- Perform the decomposition
- Identify the Layout Finishing steps



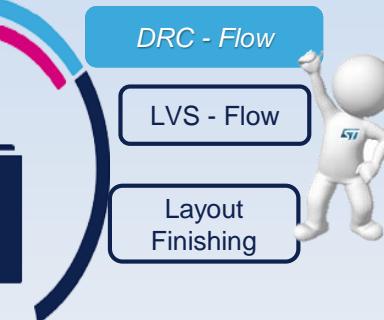
# AGENDA



# DESIGN RULES CHECK - FLOW

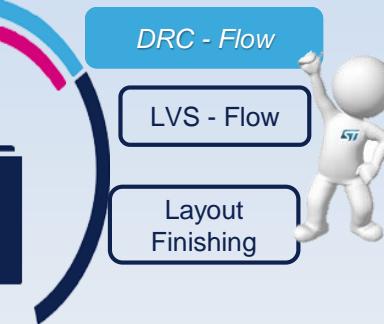
**PART I**

**CMOS & Derivative PDK**



## Tool Capabilities

	Calibre	PVS
DRC	○	○
Antenna	○	○
DRC In-Design	🚫	○
LVS	○	○
Tiling	○	○



- **Calibre Interactive**

- DRC, LVS, Antenna, PERC
- Recommended rules set by default
- Non mandatory checks can be set by default

- **PVS**

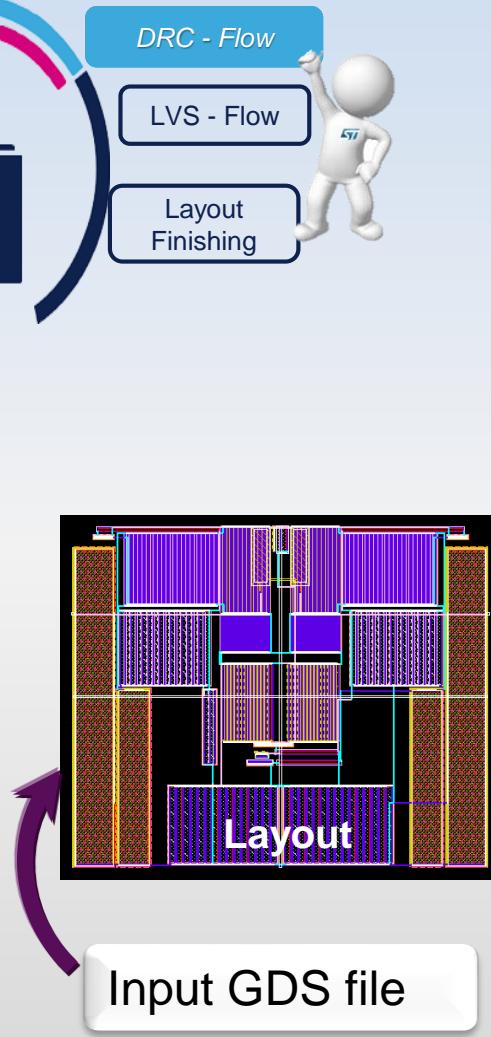
- DRC, LVS, Antenna, Tiling



# DESIGN RULES CHECK - FLOW

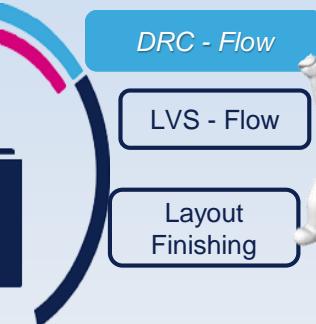
***Calibre DRC: Calibre gui***

**CMOS & Derivative PDK**

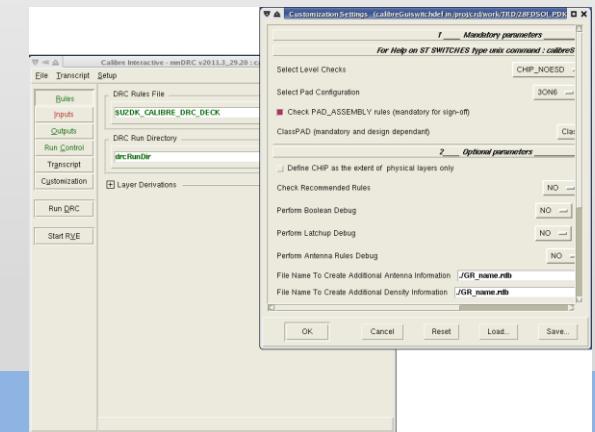


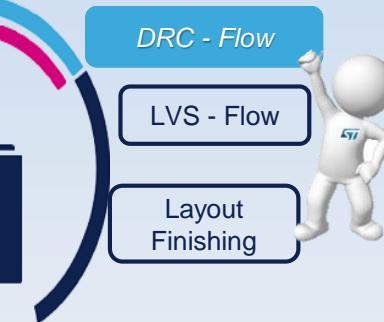
Customization file is encrypted in the PDK  
→ switches are sign-off .





- Refer to the README documentation:
    - `$PDKITROOT/DATA/DRC/CALIBRE/README`
  - Create DRC run directory
    - `mkdir -p drcRunDir`
  - Set configuration files from PDK (Runset and DRC customization)
    - `setenv MGC_CALIBRE_DRC_RUNSET_FILE $PDKITROOT/DATA/DRC/CALIBRE/calibreGuirunsetdrc`
    - `setenv MGC_CALIBRE_CUSTOMIZATION_FILE $PDKITROOT/DATA/DRC/CALIBRE/calibreGuiswitchdef`
    - `setenv U2DK_CALIBRE_DRC_DECK $PDKITROOT/DATA/DRC/CALIBRE/calibredrc_cgi`
  - Launch calibre gui:
    - `calibre -gui -drc -runset $MGC_CALIBRE_DRC_RUNSET_FILE -custom $MGC_CALIBRE_DRC_CUSTOMIZATION_FILE -drcLayoutPaths <full gds path> -drcLayoutPrimary <topcell> -batch > drc.log &`
- or
- `calibre -gui -drc -runset $MGC_CALIBRE_DRC_RUNSET_FILE -custom $MGC_CALIBRE_DRC_CUSTOMIZATION_FILE &`





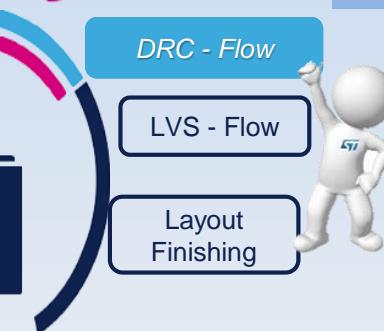
- Refer to the README documentation
  - \$PDKITROOT/DATA/DRC/CALIBRE/README
- Customization switches in  
**\$PDKITROOT/DATA/DRC/CALIBRE/calibredrc.ctrl**
  - Retrieve and copy the file in the working directory
  - Update it with the appropriate gds path and Topcell name
    - “myCell” fields to changed
- Launch calibre DRC
  - calibre –hier –drc calibredrc.ctrl > drc.log

# DESIGN RULES CHECK - FLOW

*Calibre DRC: DRC Switches description*

CMOS & Derivative PDK

# DRC calibre gui : Switches Description



**Calibre Interactive customization window**

**Customization Settings** (calibre|switchdef in /prj/upptpluscache/areas/dk/wip/DK\_cmos28FDSOI\_RF\_mmW\_6U1x\_2U2x\_2T8x\_LB/0.9-DEV-)

For Help on ST SWITCHES type unix command : calibreSTSwitchesHelp -drc

ClassPAD (mandatory and design dependant) ClassFC44S

Select Level Checks CUSTOM

Check PAD\_ASSEMBLY rules (mandatory for sign-off)

Perform Emetrology Density structures checks (mandatory for sign-off)

IP CHECKS (Density requirements for IPs to ensure compliance with SOC density rules)

Design Help

Define CHIP as the extent of physical layers only

Check Recommended Rules NO

Perform Boolean Debug NO

Perform Latchup Debug NO

Perform Antenna Rules Debug NO

Perform high voltage Rules Debug NO

Active specific post SWAP rules NO

File Name To Create Additional Antenna Information ./GR\_name.rdb

File Name To Create Additional Density Information ./GR\_name.rdb

File Name To Be Included To Execute EXCLUDE CELL Command (Inside file: EXCLUDE CELL cell1 cell2 cell3)

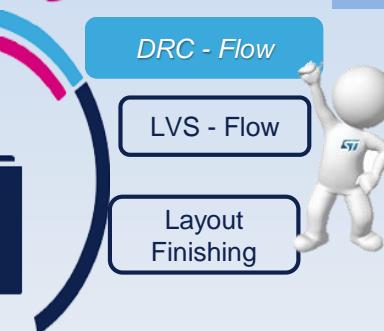
Perform Emetrology Density structures Helps

Check Density Statistic Rules

Expert parameters

Enable IP CHECKS customization no

OK Cancel Reset



Select Level Checks

- IP-Validation
- CHIP-SignOff
- CUSTOM

- **IP-Validation**

- Selects all rules except recommended rules, ID.BEOL, ID.MPW, ID.IP, ID.FEOL
- Use IP-tiling rules if any
- Recommended for DRC at IP level

- **CHIP-SignOff**

- Selects all rules except recommended rules (no IP-tiling rules)

- **CUSTOM**

- Possible to activate all the current switches (ex: lonely via, min density, ...)

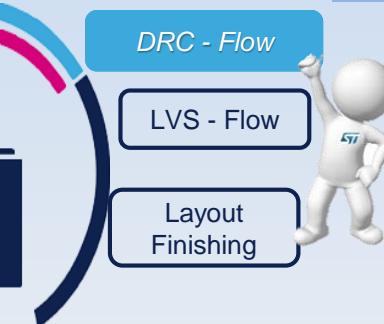
Check PAD\_ASSEMBLY rules (mandatory for sign-off)

ClassPAD (mandatory and design dependant)

ClassFC44S

- To activate or not PAD related checks

- ClassFC44S ; ClassFC61A ; ClassFC52A ; ClassWB47SR or ClassWB47SR



DRC - Flow

LVS - Flow

Layout  
Finishing

- Perform Emetrology Density structures checks (mandatory for sign-off)

- Enable Placement Design Rules for embedded metrology structures

- Perform Emetrology Density structures Helps

→ Click here to change rule

- Rules to help to debug embedded metrology structures rules

- IP CHECKS (Density requirements for IPs to ensure compliance with SOC density rules)

- To be select to enable density rules at IP level to ensure clean DRC integration at SoC level
  - Not Sign-Off
  - Only for the purpose of doing R&D
    - For IP designer, it makes sense to reduce the Step value (but increase runtime) but not the Window size

Enable IP CHECKS customization

3 - Expert parameters

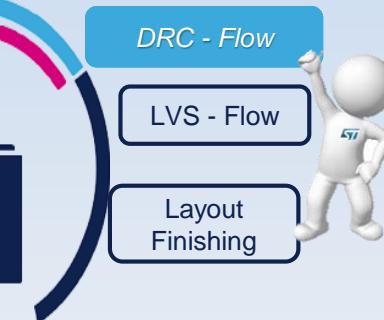
Enable IP CHECKS customization

	IP outer transition ring
P outer transition ring width (50μm)	50
M1.DEN.2: Minimum physical M1 density (35%)	35
M1.DEN.2: Minimum physical M1 density: Window (50'50)	50
M1.DEN.2: Minimum physical M1 density: Step (25)	25
M1.DEN.2.1: Minimum physical M1 density (55%)	55
M1.DEN.2.1: Minimum physical M1 density: Window (50'50)	50
M1.DEN.2.1: Minimum physical M1 density: Step (25)	25
Mx.DEN.2: Minimum physical M1 density (35%)	35
Mx.DEN.2: Minimum physical M1 density: Window (50'50)	50
Mx.DEN.2: Minimum physical M1 density: Step (25)	25
Mx.DEN.2.1: Minimum physical M1 density (55%)	55
Mx.DEN.2.1: Minimum physical M1 density: Window (50'50)	50
Mx.DEN.2.1: Minimum physical M1 density: Step (25)	25
Bx.DEN.2: Minimum physical M1 density (35%)	35
Bx.DEN.2: Minimum physical M1 density: Window (50'50)	50
Bx.DEN.2: Minimum physical M1 density: Step (25)	25
Bx.DEN.2.1: Minimum physical M1 density (55%)	55
Bx.DEN.2.1: Minimum physical M1 density: Window (50'50)	50
Bx.DEN.2.1: Minimum physical M1 density: Step (25)	25
Kx.DEN.2: Minimum physical M1 density (35%)	35

OK Cancel Reset

More information about rules customization in the DRM

# DRC calibre gui: Switches Description



DRC - Flow

LVS - Flow

Layout  
Finishing

Check Recommended Rules

- If set to YES runs rules in the manual with an "R" suffix

Perform Boolean Debug

- Output the generated boolean layers which touch a failing boolean check.

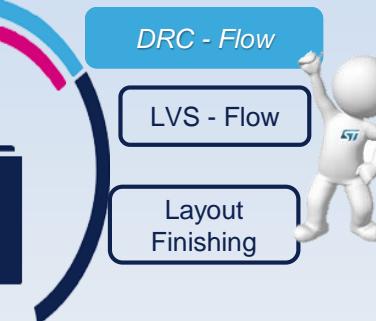
Perform Latchup Debug

- To help in Latch-up errors debug
  - Corresponding to each LUP derived geometry listed in DRM
  - Corresponding to IO Pads and Supply Pads
    - Highlight of emitters, 20µm area around emitters
    - Highlight all zones 20um around emitter derived geometries

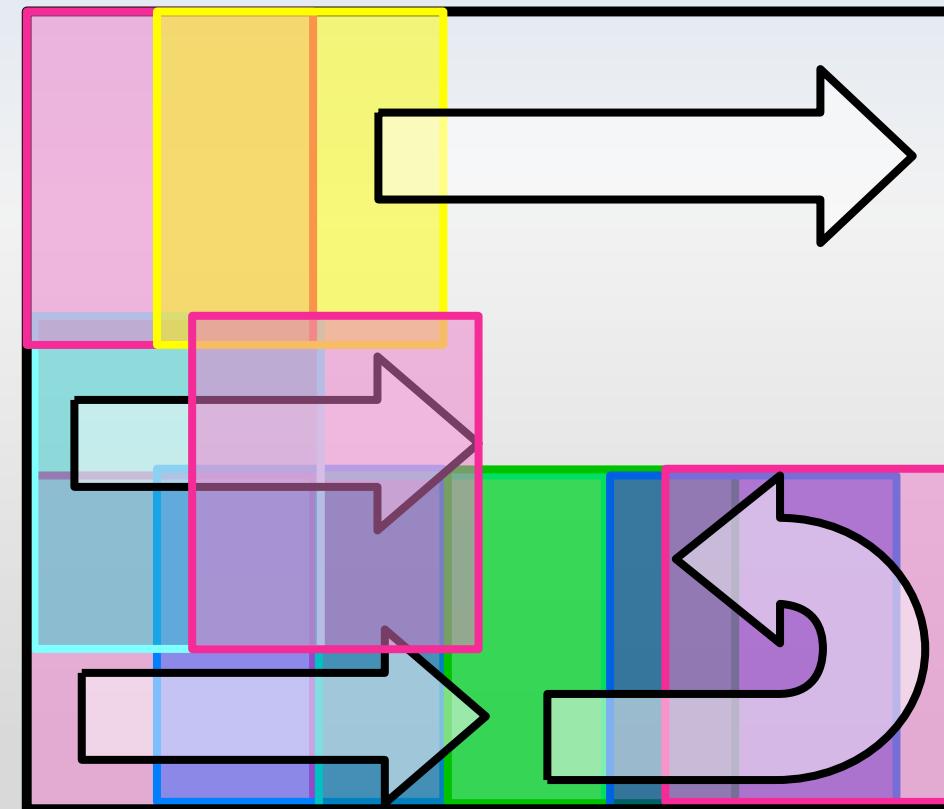
Perform Antenna Rules Debug

- To help in Antenna rules debugging
  - Permit to catch the considered connection in the design
    - have more values available in the rdb file
    - more results on intermediate metals

# Focus: Density Rules Checking

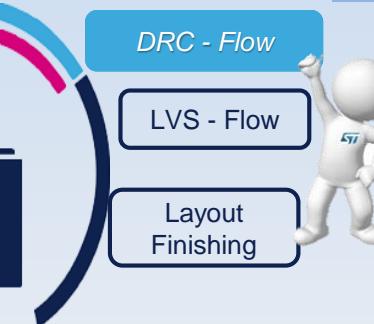


As required by DRM, the checks are done by windows, and gradient between windows is also taken into account.



**Circuit extent is used to check for “global” DENSITY**

**square windows with 50% overlap are used to check local DENSITY**

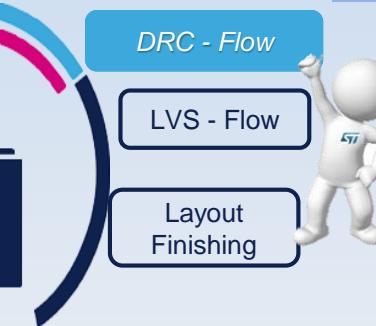


# Focus: Density Rules Checking

## 3 KINDS OF DENSITY CHECKS

- 1 min and max allowed density for all region.
- 2 maximum density on 3 consecutive metal levels on a same local area
- 3 density gradient (neighbouring windows or across chip)

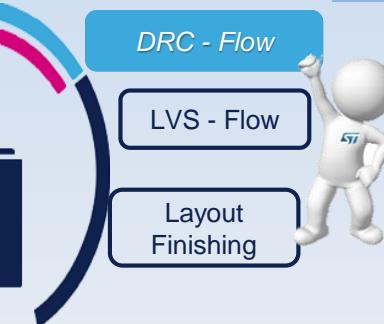
# Calibre gui Run Control Settings



- The Calibre gui run control must be set to use large resources easily. in particular:
  - Mode: all the following options must be set correctly
    - Distributed (MTFlex)
    - Hyperscale
    - Remote Data Server (RDS) with argument “1”
    - Hyperscale Remote
    - Backup Data may be left unselected by default
    - custom switch generating the command LAYOUT TURBO FLEX YES, selected by default
  - Resources
    - Master Host Selection

Min CPUs	Type	Min Memory	Resource Options	Submit Options	Matching Hosts	Wait	Port	Count	Min Count	Name
			<code>select[type==%o] rusage[mem=%m] span[hosts=1]</code>	<code>-q master -n %c</code>		300				

# Calibre gui Run Control Settings



Use	Total CPUs	Min CPUs Per Host	Type	Min Memory Per Host	Resource Options	Submit Options	Matching Hosts	Wait	Port	Count	Min Count	Name
	48	4	RH60	16384	select[ncpus>=%c && type==%o && m...	-q reg -P cmos14nmfdsoi	113					

Set the master and remote resources depending on your Chip Size.

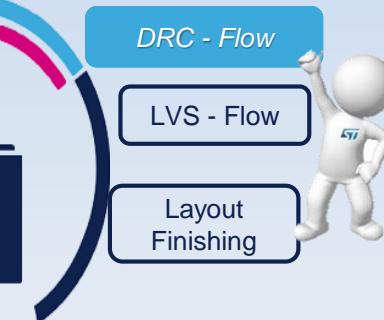
Example:

Chip Size	Master: Min CPUs Chip Size (mm <sup>2</sup> ) / 5	Master: Min Memory Per Host Master Min CPUs x 8192 (MB)	Remote: Total CPUs Master Min CPUs x 8
28 mm <sup>2</sup>	6	6 x 8192 = 49152	6 x 8 = 48

Set the remaining remote resources (independent of Chip Size, keep defaults)

4 Remote: Min CPUs per Host: 4  
Remote: Min Memory Per Host: 16384

# Calibre gui Run Control Settings



- Already default setting defined by the Design Kit
  - Good trade-off between the ideal configuration to run any full chip overnight and the resources available

Min CPUs	Type	Min Memory	Resource Options	Submit Options	Matching Hosts	Wait	Port	Count	Min Count	Name
8	RH50	32768	<code>select[type==%o] rusage[mem=%m] span[hosts=1]</code>	<code>-q master -n %c</code>		300			24	

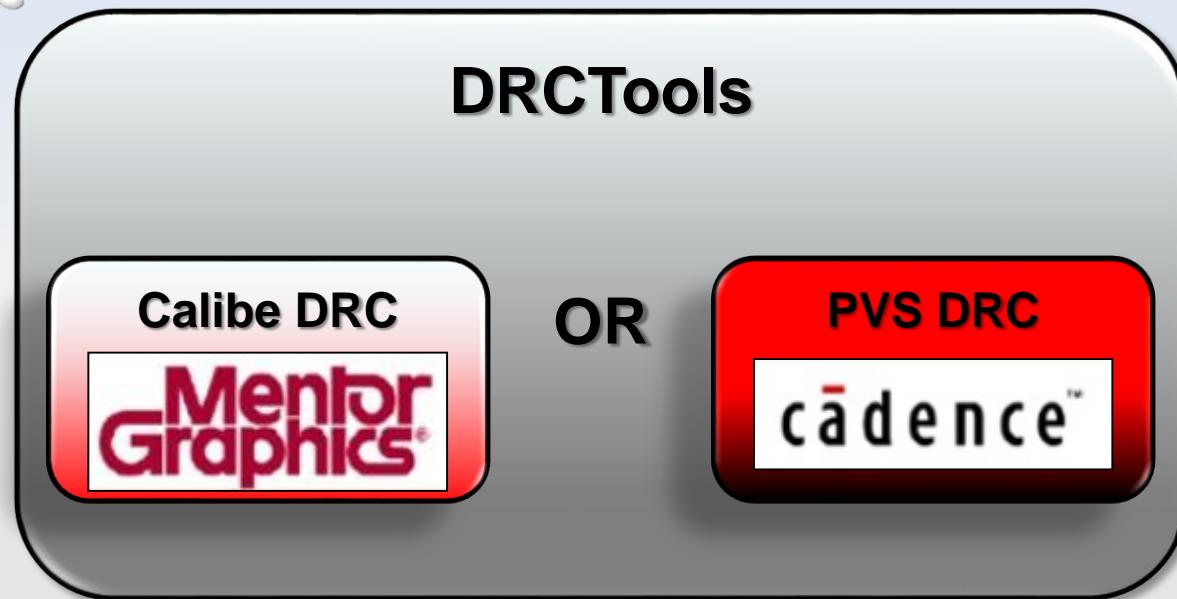
- Remote Host Selection

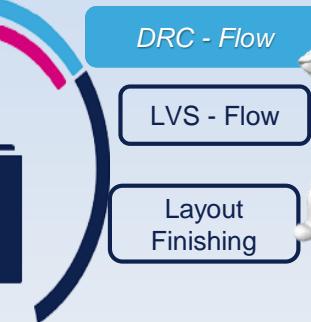
Use	Total CPUs	Min CPUs Per Host	Type	Min Memory	Resource Options	Submit Options	Matching Hosts
/	48	4	RH50	8192	<code>select[type==%o] rusage[mem=%m] span[ptile=%c]</code>	<code>-q slave</code>	

# DESIGN RULES CHECK - FLOW

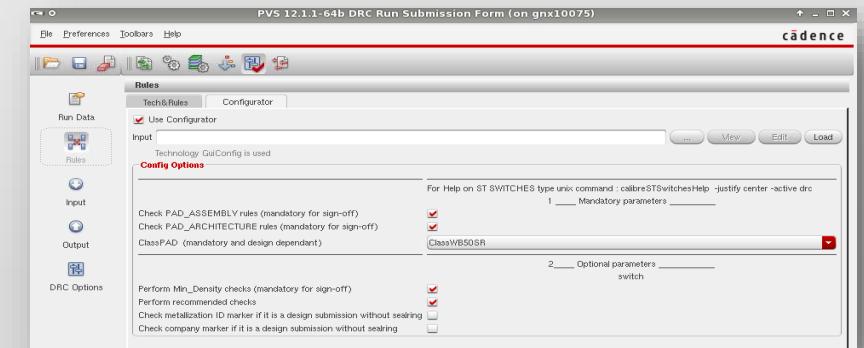
**PVS DRC**

**CMOS & Derivative PDK**





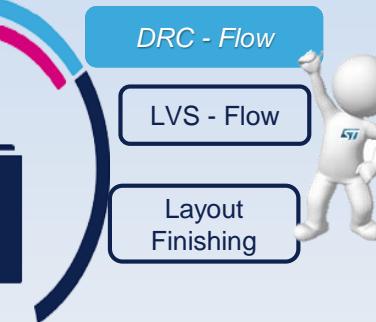
- Refer to the README documentation:
  - `$PDKITROOT/DATA/DRC/PVS/README`
  
- Set configuration files from PDK (Runset and DRC customization)
  - `setenv U2DK_PVS_PACK $PDKITROOT/DATA/DRC/PVS/pvs_pack/drc_cmos028FDSOI`
  - `setenv U2DK_PVS_DRC_TECH_FILE $PDKITROOT/DATA/DRC/PVS/pvtech.lib`
  - `setenv Layer_Map $PDKITROOT/DATA/LIB/OpenAccess/cmos32Ip_Tech/cmos32Ip_Tech.layermap`
  
- Launch PVS-DRC on a full layout
  - Click on Launch => Plugins => PVS
  - on the new menu which appeared on layout view, choose PVS => run DRC
  - on the new window do
    - “File=> load preset”
    - load `$PDKITROOT/DATA/DRC/PVS/DRC.preset`
  - -click on Submit
    - DRC errors will be displayed on a new window.



# DESIGN RULES CHECK - FLOW

***Calibre RealTime***

**CMOS & Derivative PDK**



## *What is Calibre RealTime ?*

Instantaneous DRC using Calibre



Same deck, same engine, same results as  
Calibre Interactive

Must be considered as a complement (not  
as a replacement) to help improving the  
layout verification productivity

DRC - Flow

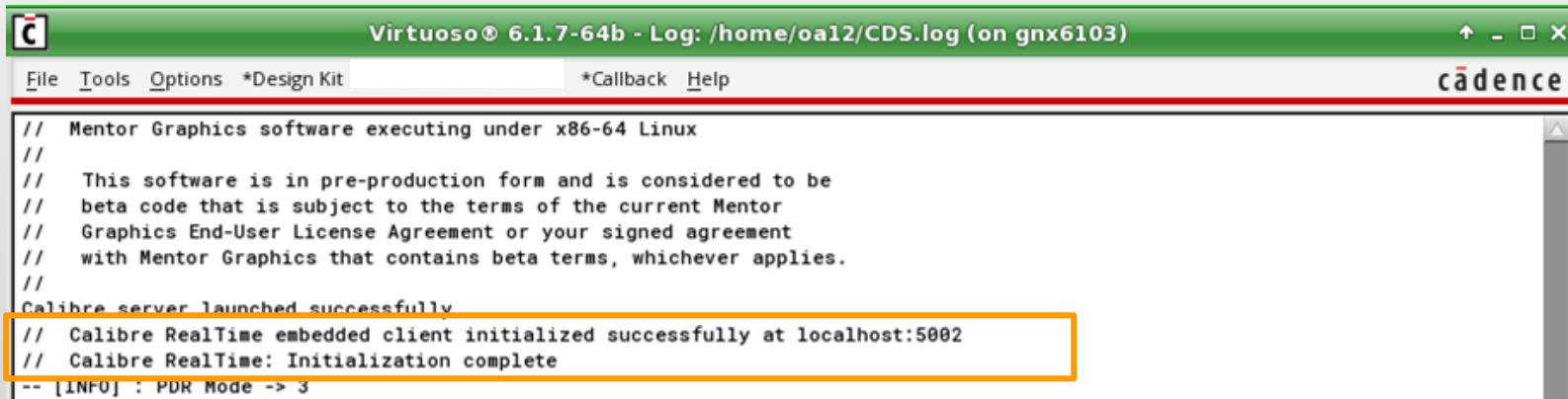
LVS - Flow

 Layout  
Finishing


- Define the following environment variables

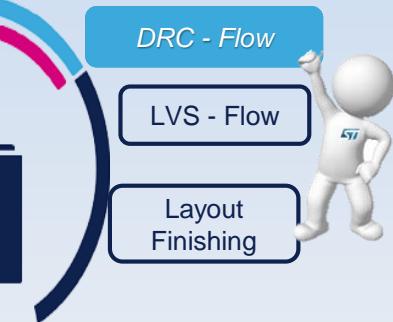
```
setenv LD_LIBRARY_PATH $MGC_HOME/shared/pkgs/icv/tools/calibre_client/lib/64:$LD_LIBRARY_PATH
setenv OA_PLUGIN_PATH $MGC_HOME/shared/pkgs/icv/tools/querieskl
setenv MGC_CALIBRE_REALTIME_VIRTUOSO_ENABLED 1
setenv MGC_CALIBRE_REALTIME_VIRTUOSO_NOT_USE_MESSENGER 1
```

- Launch Virtuoso in 64b
- Check setup is OK



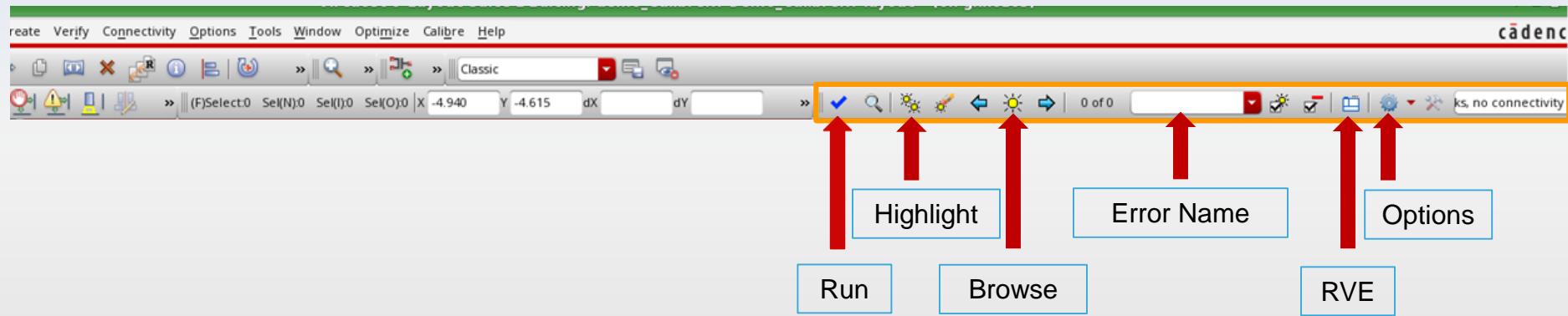
```
Virtuoso® 6.1.7-64b - Log: /home/oa12/CDS.log (on gnx6103)
File Tools Options *Design Kit      *Callback Help
cadence

// Mentor Graphics software executing under x86-64 Linux
//
// This software is in pre-production form and is considered to be
// beta code that is subject to the terms of the current Mentor
// Graphics End-User License Agreement or your signed agreement
// with Mentor Graphics that contains beta terms, whichever applies.
//
Calibre server launched successfully
// Calibre RealTime embedded client initialized successfully at localhost:5002
// Calibre RealTime: Initialization complete
-- [INFO] : PDK Mode -> 3
```

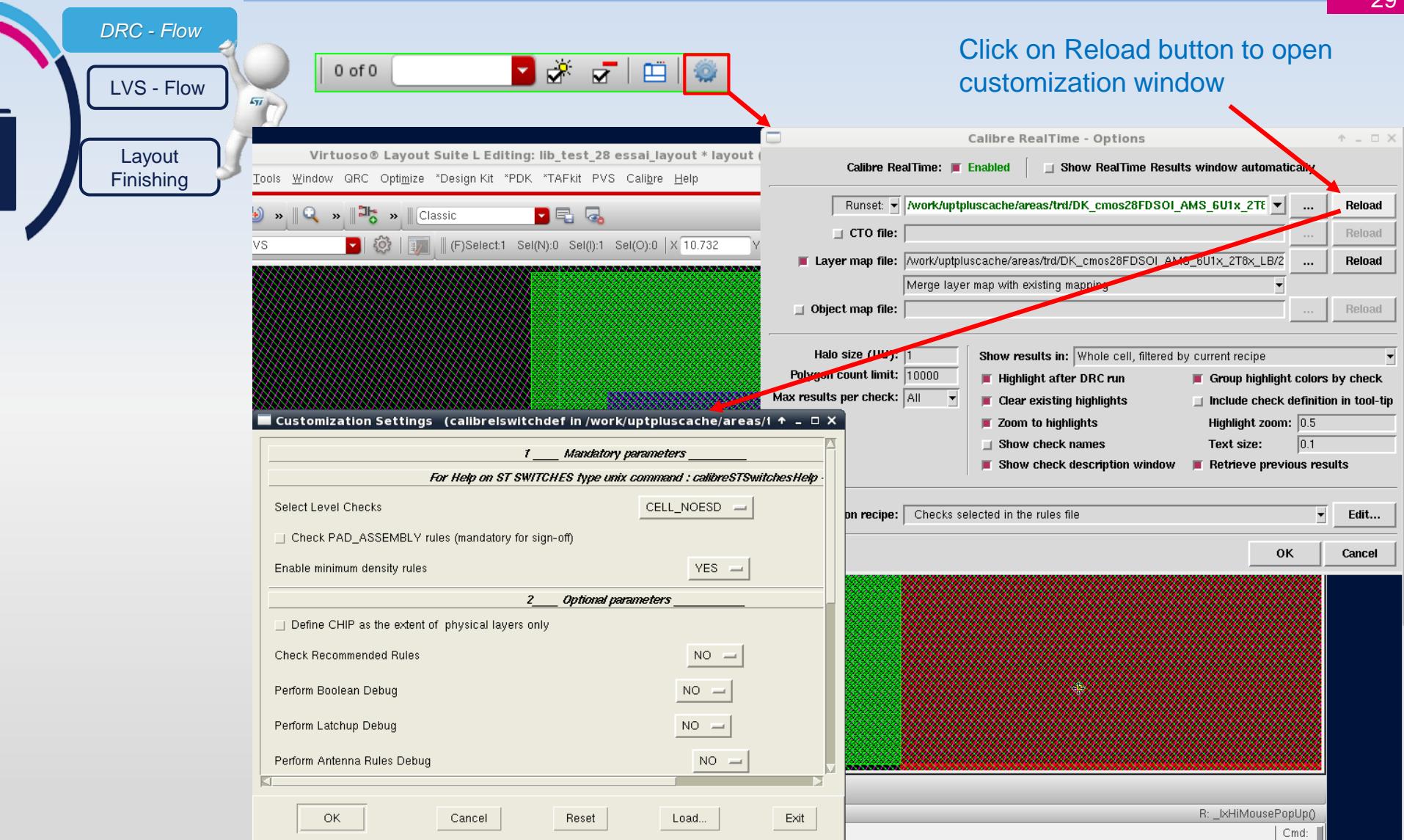


# Calibre RealTime commands

Calibre RealTime commands appear in layout window



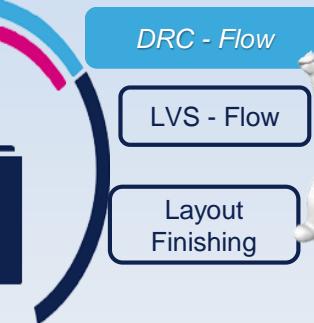
# Calibre RealTime run customization



# DESIGN RULES CHECK - FLOW

***Interactive PVS***

**CMOS & Derivative PDK**



## What is iPVS ?

In same time the layout is done, DRC errors are highlighted



Integrated DRC (like In-Design capability)

Must be considered as a complement (not as a replacement) to help improving the layout verification productivity

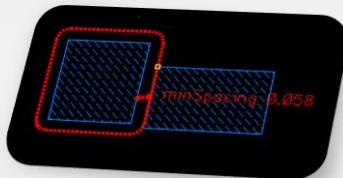


## Run PVS on In Design mode

- DRD Edit toolbar visible with 3 modes

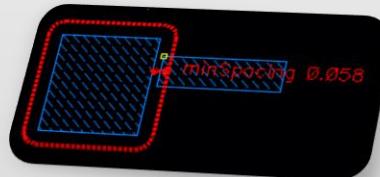
### **Enforce mode :**

- Shows surrounding regions for [techfile rules](#) violation
- Forbids shapes placement in the layout



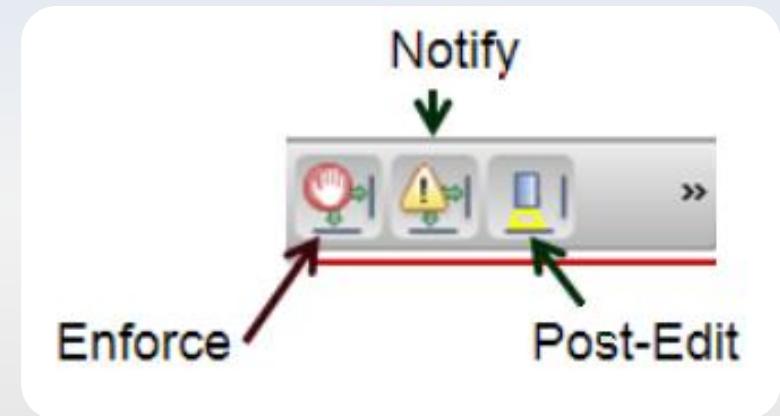
### **Notify mode :**

- Shows surrounding regions for [techfile rules](#) violation.
- Shapes placement in the layout still possible

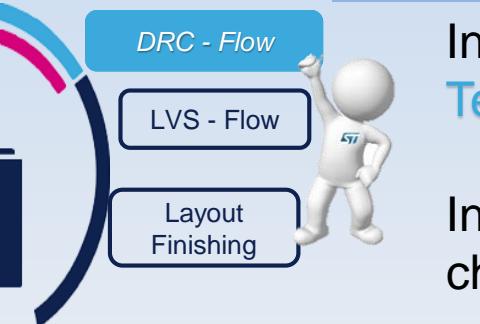


### **Post-Edit mode :**

- After whatever modification of the layout
- Complete [PVS-DRC deck](#) is run
- On current cellview or on complete layout



# In Design mode



In the **Enforced / Notified** modes, only rules defined in **Techfile** are checked ;

In the **Post-Edit** mode, all rules from the **PVS deck** are checked, need to full PVS check.

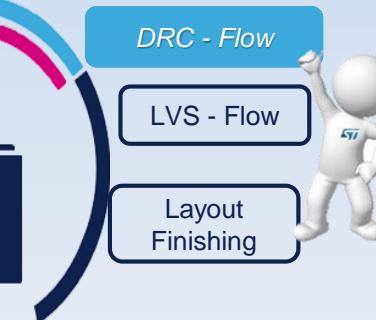
The screenshot shows the Cadence DRD Options dialog box. On the left, there are sections for DRD Mode (checkboxes for Enforce, Notify, Post-Edit), Hierarchy Depth (set to Top Level), and various rule categories like All, Spacing, Via, Width, Area, Edge Length, Extension, Length, and Misc. Under Notify Options, checkboxes are shown for Halos, Dashed, True Color, Rule Text, Arrows, and Violation Edges. Under Enforce Options, checkboxes include Enable Pushing, Push Vias, Keep Stacked, Cleanup After Push, and Targets Only. Post-Edit Options include Marker Limit (5000) and Timeout (secs) (1). At the bottom, there's a Virtuouso IPVS section with a Setup... button and a checkbox for Use IPVS for Post-Edit & Batch. A status message at the bottom says "Status Press \"Setup...\" to start up". A context menu is open over the DRD Options window, with the 'Assistants' submenu and the 'Annotation Browser' item both highlighted with orange boxes. The 'Assistants' submenu includes options like Display..., Editor..., Selection..., Magnifier..., DRD Edit..., Toolbox..., Highlight..., and Dynamic Display... The 'Annotation Browser' submenu includes Dynamic Selection, Search, Property Editor, Palette, Navigator, and World View. A 'Toggle Visibility' button with F11 keybinding is also visible in the menu.

**When errors, shapes are highlighted**  
➤ Activate annotation browser for seeing In Design errors

# DESIGN RULES CHECK - FLOW

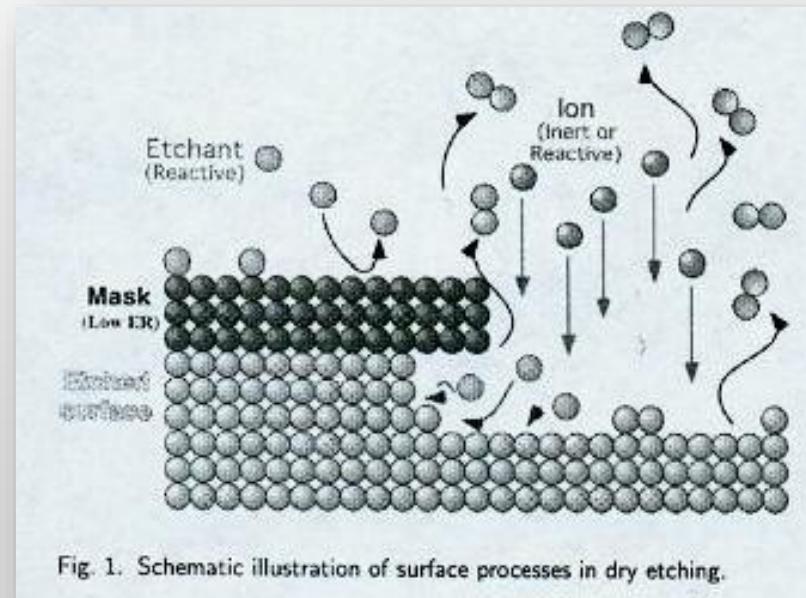
*Antenna DRC*

CMOS & Derivative PDK



During the process, intense E-field can occur.

- Accumulation of electrostatic charges
- In IC Manufacturing , the term ‘antenna’ refers to partially processed metal/poly leads, which collect charges from Plasma etching in the wafer process
- This collection of charges in the gate leads to a discharge to the S/D: damages!



➤ Plasma process damage includes

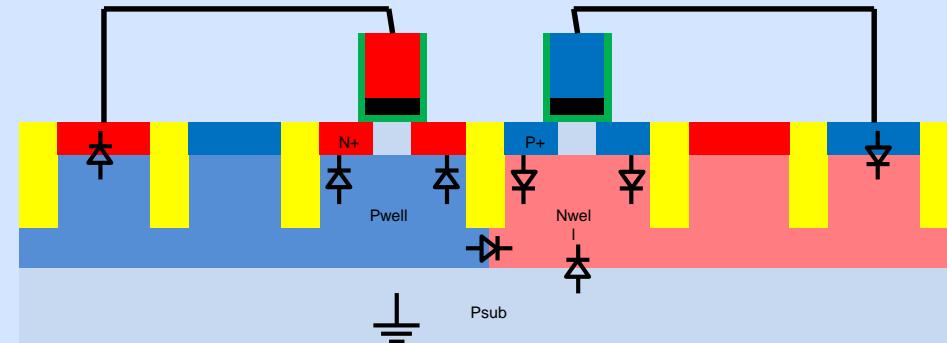
- Silicon Lattice Damage
- Metal Contamination
- Oxide damage

Oxide Damage due to the charge collection on the metal segment will degrade the transistors operation/performance.

DRC - Flow

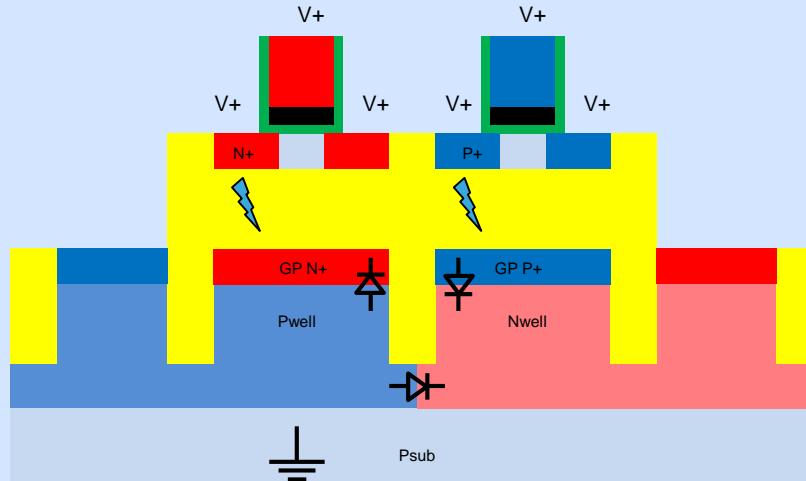
- In Bulk technology

- Gates are isolated from the substrate  
→ antenna rules to protect gates



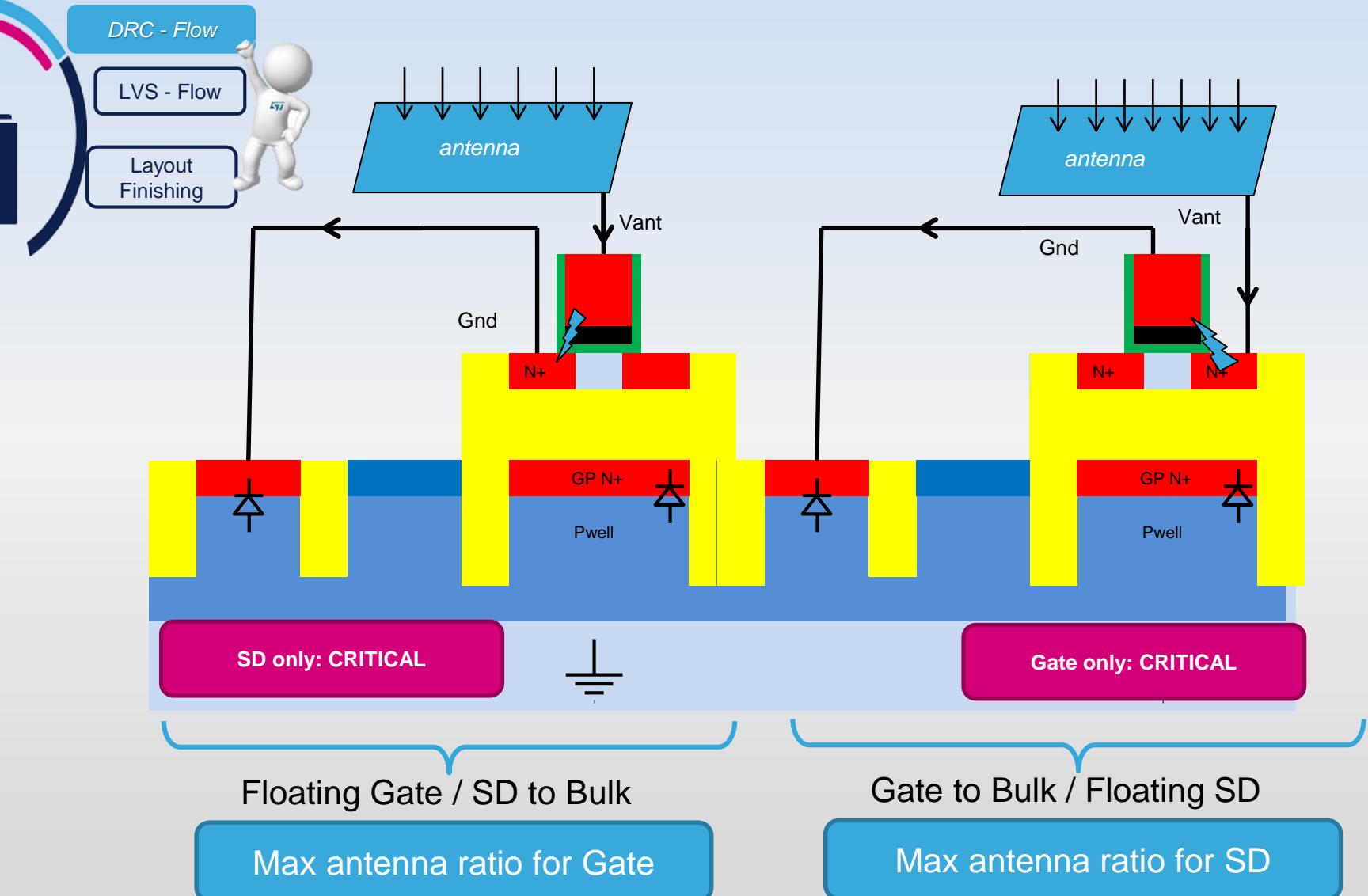
- In SOI technology

- both **Gates and Source/Drain** are isolated from the substrate  
→ antenna rules to protect gates **and SD**



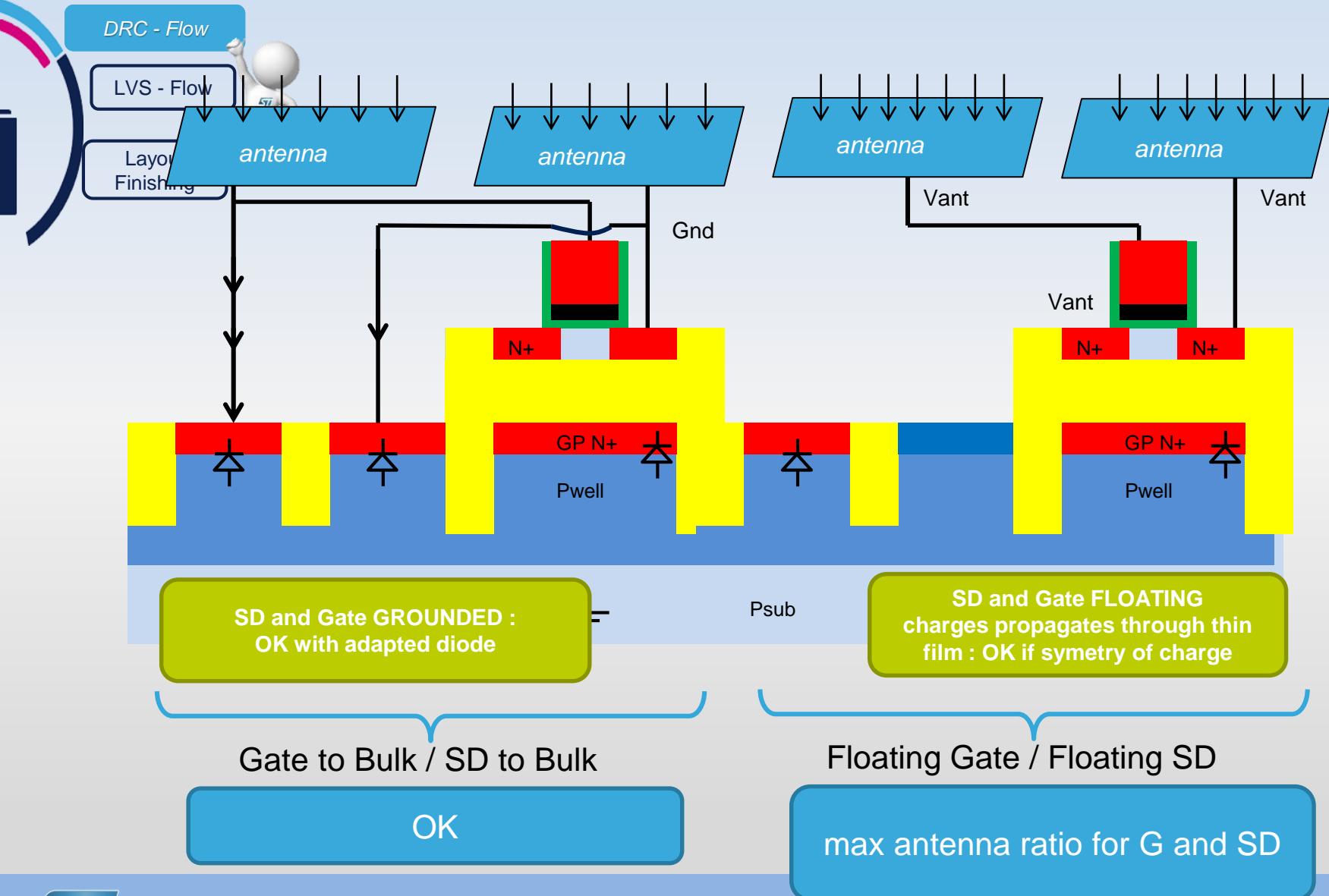
# 28UTBB-FDSOI : Antenna rules

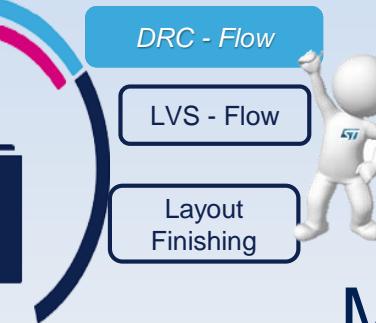
37



# DRC run for: Antenna rules

38

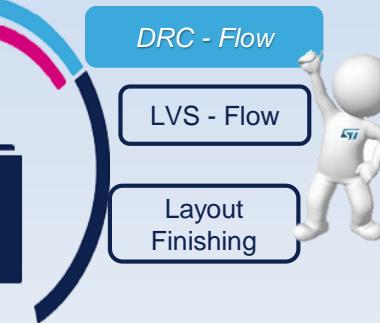




Magnitude of effect proportional to :

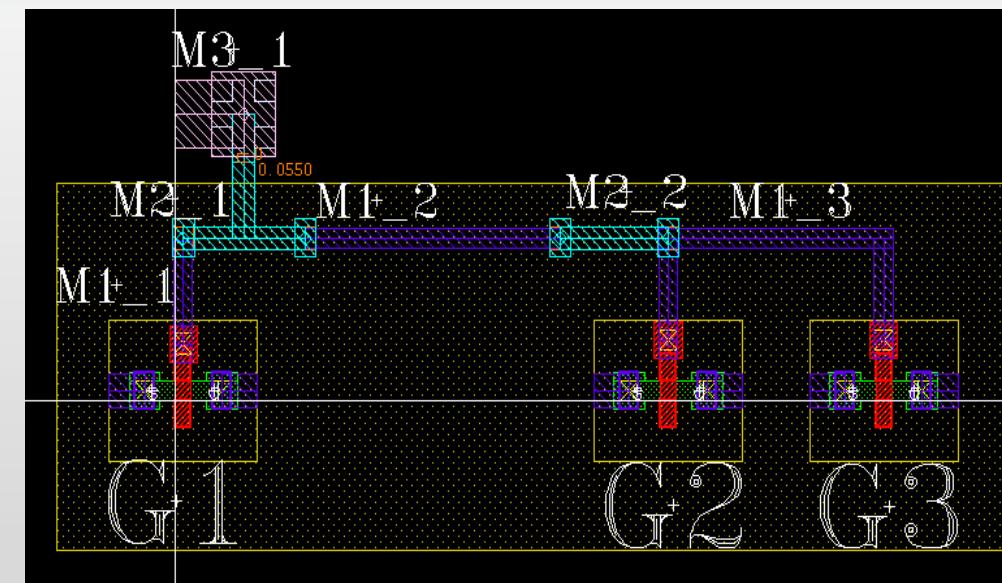
$$\frac{\text{Exposed Conductor Area}}{\text{Gate Oxide Area}}$$

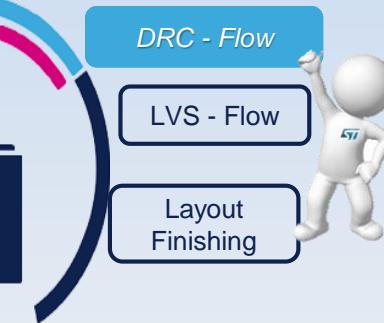
DRC rules check if this ratio is higher than the limit defined in the DRM



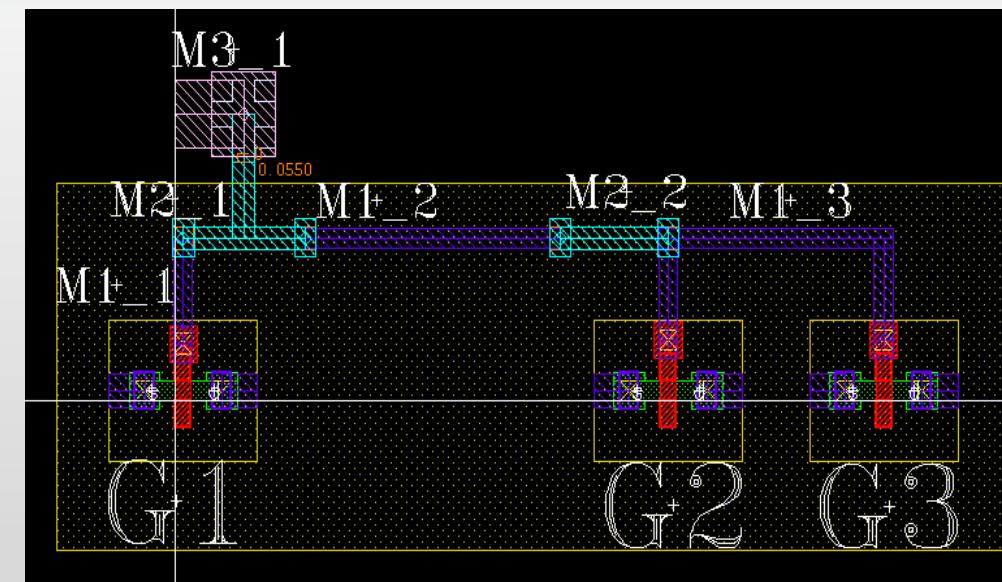
## Partial Antenna Ratio

Ratio of each poly, contact, metalx and viax of the net with regard to the gate

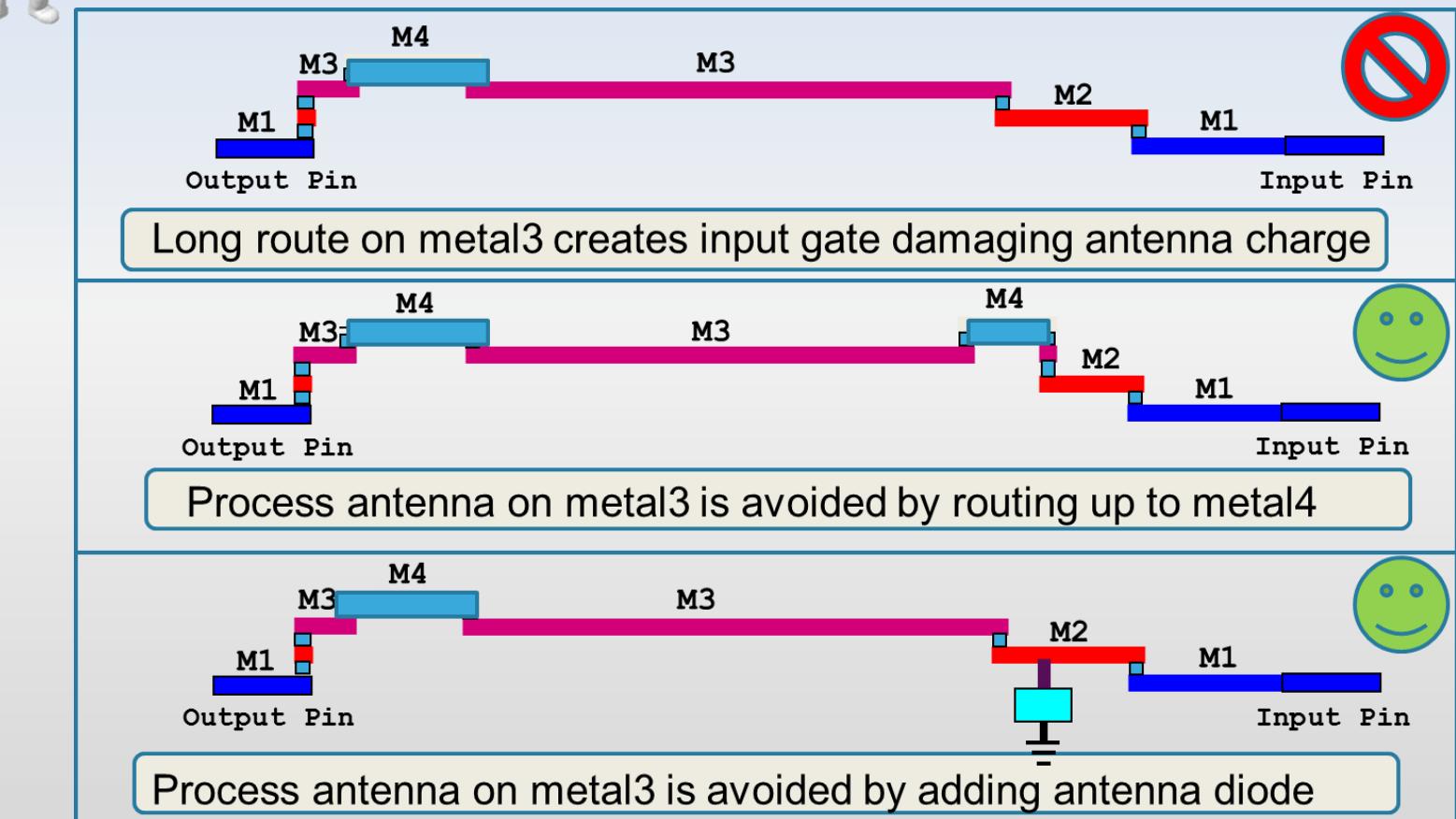


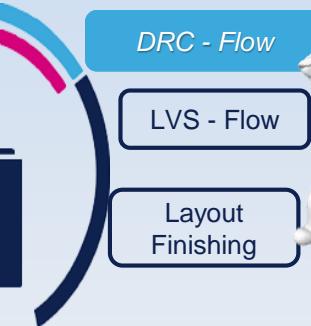


**Cumulated Antenna Ratio:**  
Cumulate poly and metal *PAR*.  
No cumulate contact and via *PAR*.



# Antenna Layout protection example

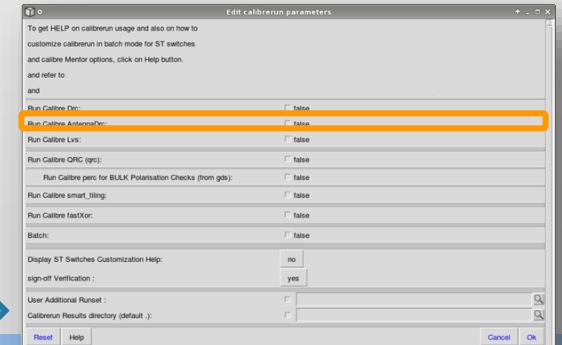


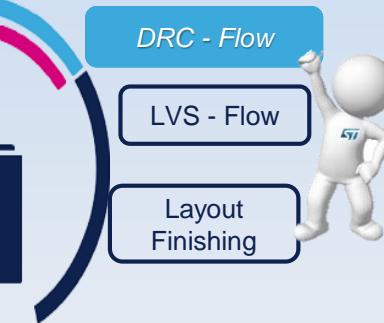


- Antenna rules can be checked with *standard DRC* (antenna checks activated by default) *or* with dedicated *AntennaDRC*
- For dedicated AntennaDRC checks, refer to the README documentation:
  - `$PDKITROOT/DATA/ANTENNA/CALIBRE/README`
- Create ANTENNA run directory
  - `mkdir -p antRunDir`
- Set configuration files from PDK (Runset and DRC customization)
  - `setenv MGC_CALIBRE_ANTENNA_RUNSET_FILE $PDKITROOT/DATA/ANTENNA/CALIBRE/calibreGuirunsetAntennadrc`
  - `setenv MGC_CALIBRE_ANTENNA_CUSTOMIZATION_FILE $PDKITROOT/DATA/ANTENNA/CALIBRE/calibreGuiswitchdefAntennadrc`
  - `setenv U2DK_CALIBRE_ANTENNA_DRC_DECK $PDKITROOT/DATA/ANTENNA/CALIBRE/calibreAntennadrc_cgi`
- Launch calibre gui:
  - `calibre -gui -drc -runset $MGC_CALIBRE_ANTENNA_RUNSET_FILE -custom $MGC_CALIBRE_ANTENNA_CUSTOMIZATION_FILE -drcLayoutPaths <full gds path> -drcLayoutPrimary <topcell> -batch > drc.log &`

or

- `calibre -gui -drc -runset $MGC_CALIBRE_ANTENNA_RUNSET_FILE -custom $MGC_CALIBRE_ANTENNA_CUSTOMIZATION_FILE &`



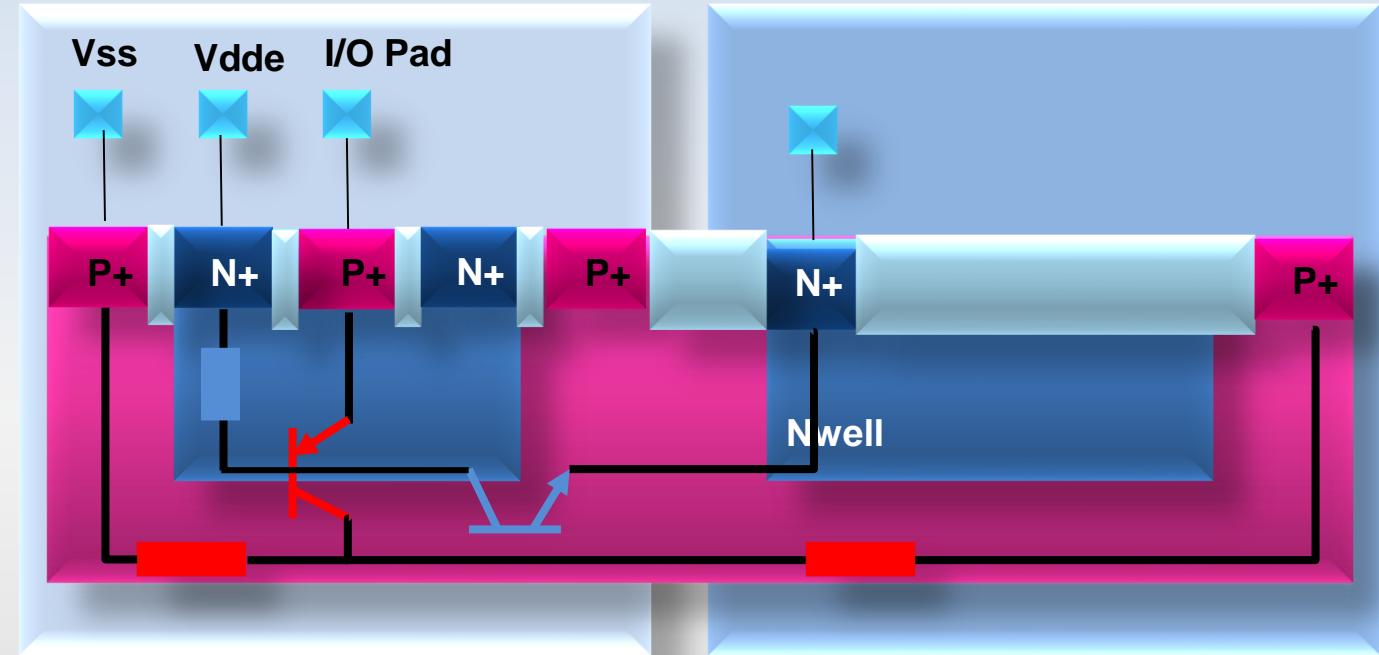
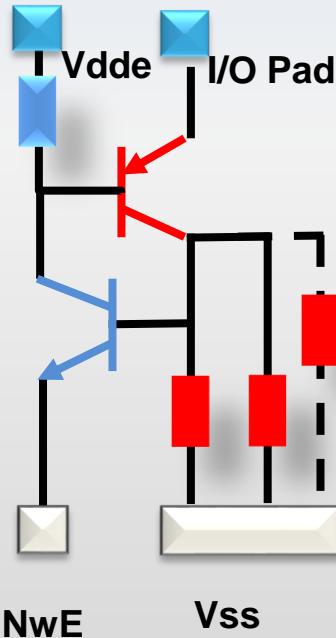
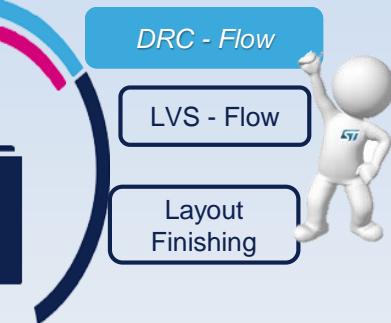


- Refer to the README documentation
  - `$PDKITROOT/DATA/ANTENNA/CALIBRE/README`
- Customization switches in  
**`$PDKITROOT/DATA/ANTENNA/CALIBRE/calibreAntennadrc.ctrl`**
  - Retrieve and copy the file in the working directory
  - Update it with the appropriate gds path and Topcell name
    - “myCell” fields to changed
- Launch calibre Antenna
  - `calibre -hier -drc calibreAntennadrc.ctrl > ant.log`

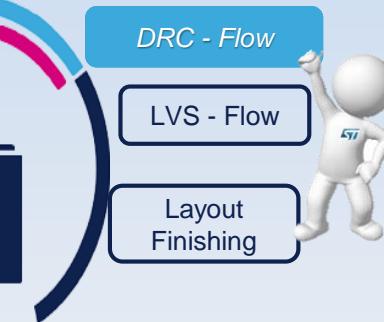
# DESIGN RULES CHECK - FLOW

***Latch-up Rules***

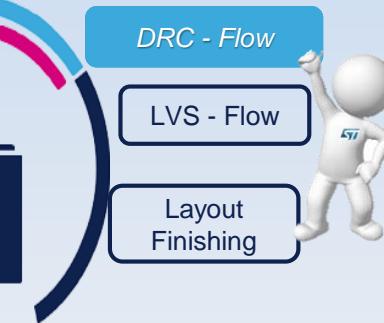
**CMOS & Derivative PDK**



- No Risk For device On SOI Film (No injection in substrate)
- Remaining Injectors are on Hybrid Part (mainly ESD Devices). Therefore risk is:
  - LU between Hybrid parts.
  - LU due to abutment between IO Pad and Wells (especially if NW@0V)



- Consequences for DRM Rules
  - Redefinition of Injectors : On hybrid part only => (“.And.HYBRYD”)
    - All substrate plugs must be surrounding by hybrid layer otherwise all LUP rules will highlighted
- Guard Rings (GRs):
  - Definition of guard rings and double guard rings updated
  - GRs Rules applies only to Injector In hybrid.
  - Suppression of Guard Rings (GRs) for all devices which are not on HYBRID
    - » No GRs for NMOS, PMOS IO Drivers which are on SOI Film.
    - » No GRs for protection of ‘victims’ on SOI Film.
- ESD Emitters to be added for
  - » Protection of “victims” on Hybrid part (LUP.R.1, LUP.R.2)
  - » Hot Well spacing Rules.
- Improvement of existing 28nm rules
  - » Readjustment of LUP.R8 => GRs for ESDS Devices (clarification vs 28 Bulk)
  - » Distance Isolated Pw to Pw: LUP.D19&20 values will be aligned to 3T18 rules (isolation rules).
  - » For protection of victim on Hybrid part (GRs) , exclusion of all ESD devices (gate & STI diode, ESDS, ESDT) as victims

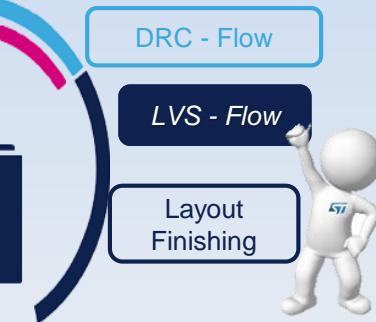


- LUP rules automatically tested during calibre drc
- Voltage assessment for latch-up rules update
  - Automatic detection of well voltage properties with labels
    - If no label is detected, the voltage property of the wells is derived from the layout following design rules (cf DRM)
    - The label information is predominant on layout derivation for the voltage property assessment

# LAYOUT VERSUS SCHEMATIC - FLOW

*PART II*

**CMOS & Derivative PDK**



## Tool Capabilities

	Calibre	PVS
DRC	○	○
Antenna	○	○
DRC In-Design	🚫	○
LVS	○	○
Tiling	○	○

# LAYOUT VERSUS SCHEMATIC - FLOW



*LVS Flow*

**CMOS & Derivative PDK**

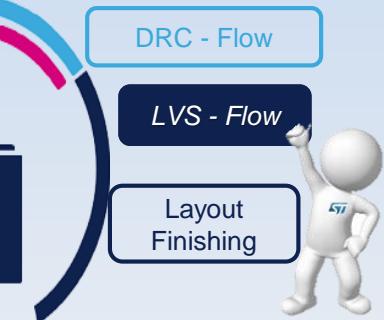
# LAYOUT VERSUS SCHEMATIC - FLOW



*LVS Flow*

***Calibre LVS***

**CMOS & Derivative PDK**

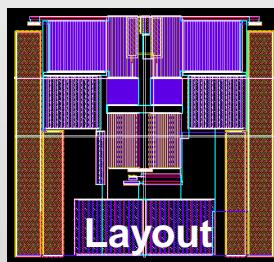
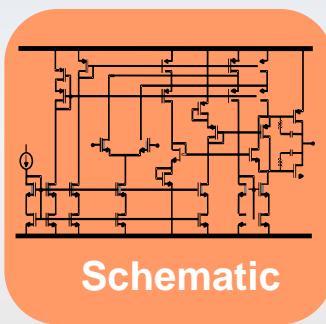


# Physical LVS Verification

53

*Is my layout equivalent to the schematic?*

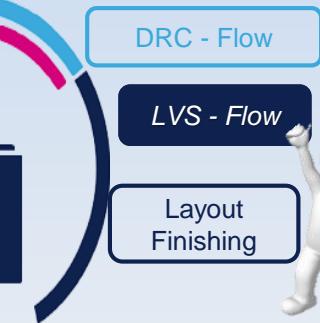
**LVS**, Layout Versus Schematic, will tell me!



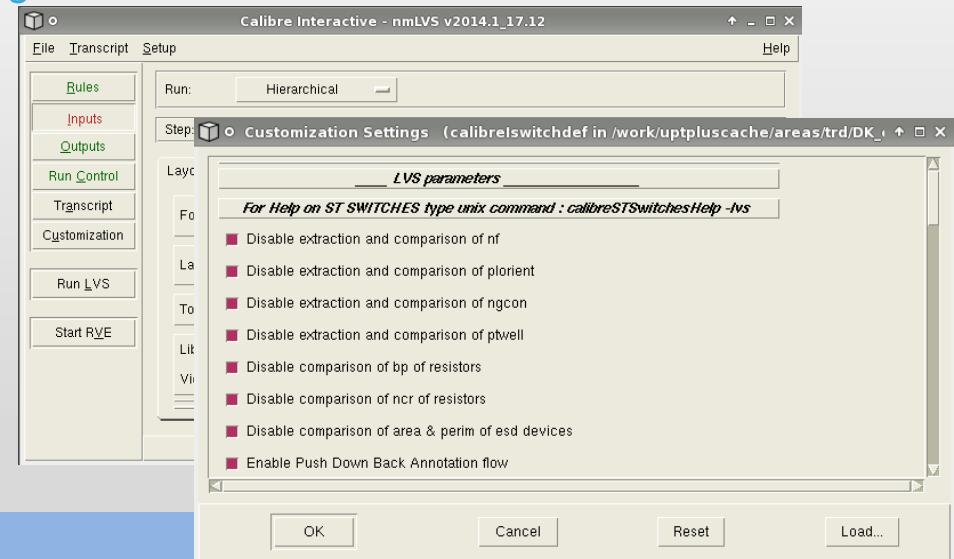
**LVS Options**



**Process Rules,  
Devices description**



- Refer to the README documentation
  - `$PDKITROOT/DATA/LVS/CALIBRE/README`
- Be sure you have loaded the file:
  - `source $PDKITROOT/PDK*.csh`
  - define all environment variable required by the flow (BEOL\_STACK, etc.)
- To use interface:
  - `setenv MGC_CALIBRE_LVS_RUNSET_FILE $PDKITROOT/DATA/LVS/CALIBRE/calibreGuirunsetlvs`
  - Launch `calibre -gui -lvs`





- Customization switches in  
**\$PDKITROOT/DATA/LVS/CALIBRE/calibreLvs.ctrl**
  - Retrieve and copy the file in the working directory
  - Update it with the appropriate gds path and Topcell name
    - “myCell” fields to changed
- Launch **calibre LVS**
  - **calibre –hier –lvs calibreLvs.ctrl > lvs.log**

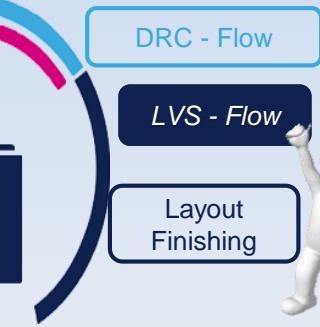
# LAYOUT VERSUS SCHEMATIC - FLOW



*LVS Flow*

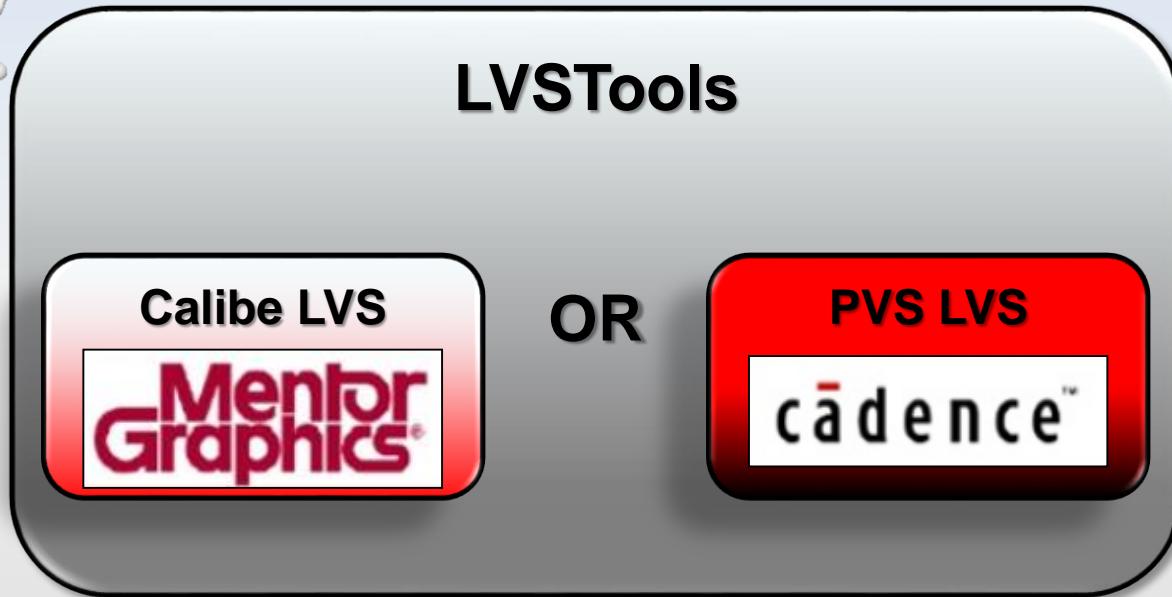
**PVS LVS**

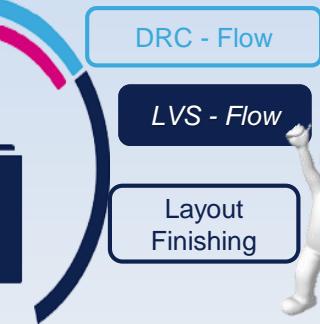
**CMOS & Derivative PDK**



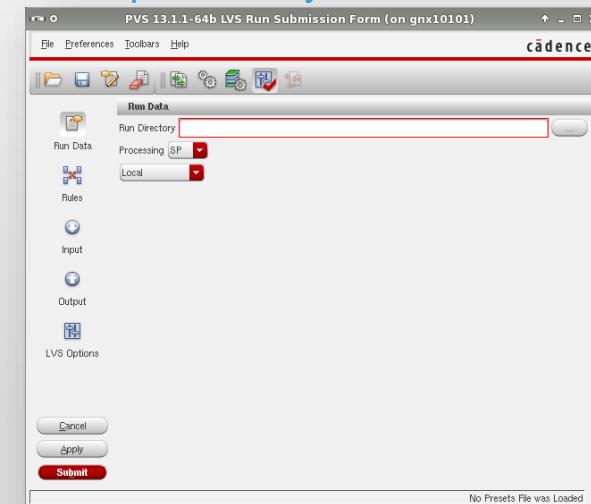
# Physical LVS Verification

57





- Refer to the README documentation:
  - **\$PDKITROOT/DATA/LVS/PVS/README**
  
- Customization switches in  
**\$PDKITROOT/DATA/LVS/PVS/pvslvs.ctrl**
  - Retrieve and copy the file in the working directory
  - Update it to customize lvs run
  
- Launch PVS-LVS
  - `pvs -lvs -top_cell "myCell" -gds "myCell.gds" -source_top_cell "myCell" -source_cdl "myCell.cdl" pvslvs.ctrl >& lvs.log`

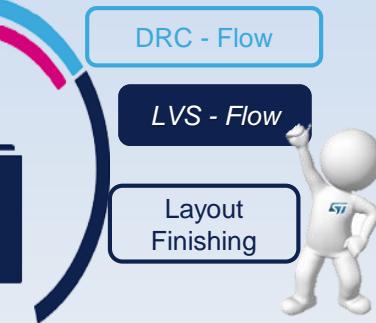


# LAYOUT VERSUS SCHEMATIC - FLOW



*Advanced technology features*

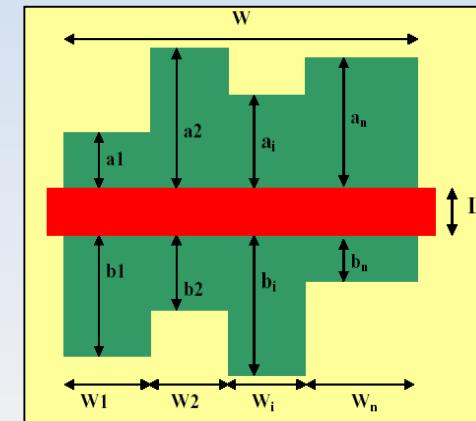
**CMOS & Derivative PDK**



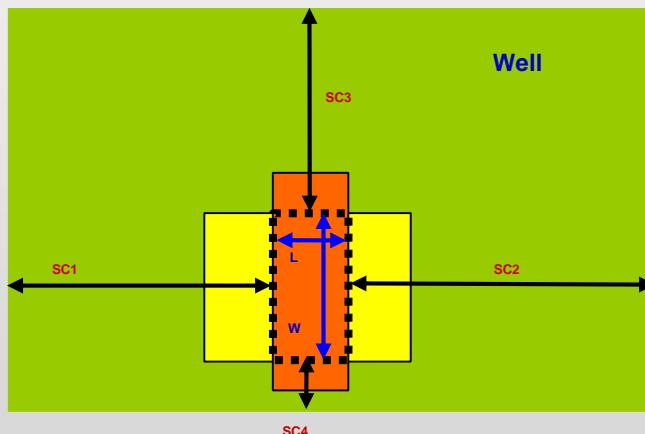
# Spice Accuracy Parameters

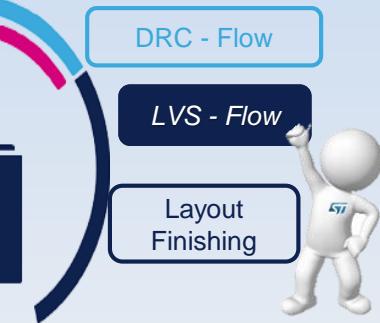
60

Stress STI:  
SA, SA1, SA2, SA3  
SB, SB1, SB2, SB3

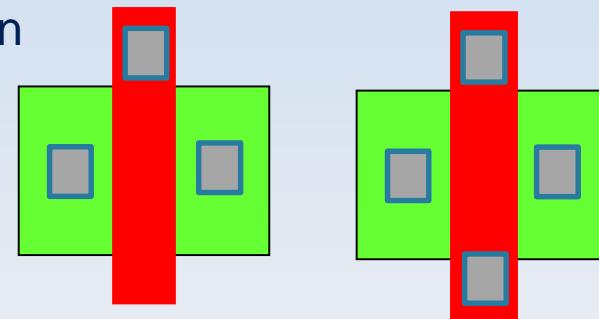


Source/Drain Area &  
perimeter: AS,AD,PS,PD



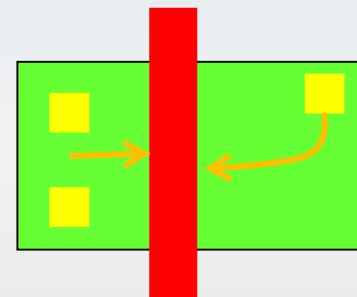


Number of gate access: ngcon

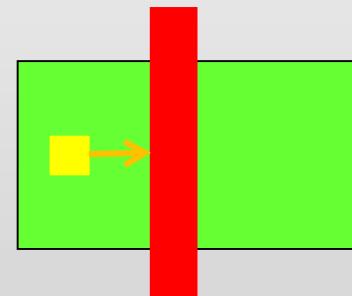


NRS/NRD is dependent with :

- OD area
- number of contacts



NRS/NRD corresponds to OD resistance effect to go from source/drain contacts to gate



# Spice Accuracy Parameters : p\_la

DRC - Flow

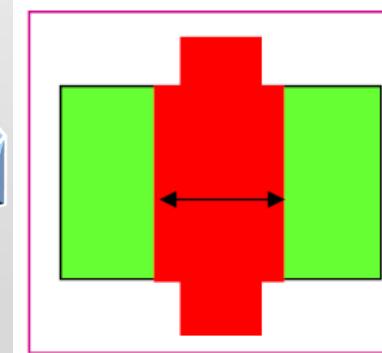
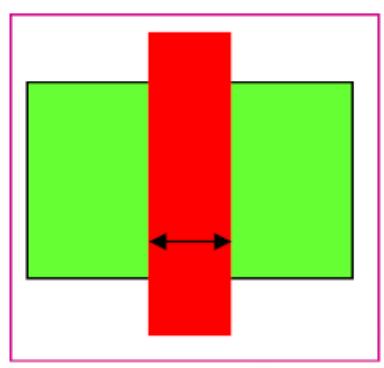
LVS - Flow

Layout  
Finishing



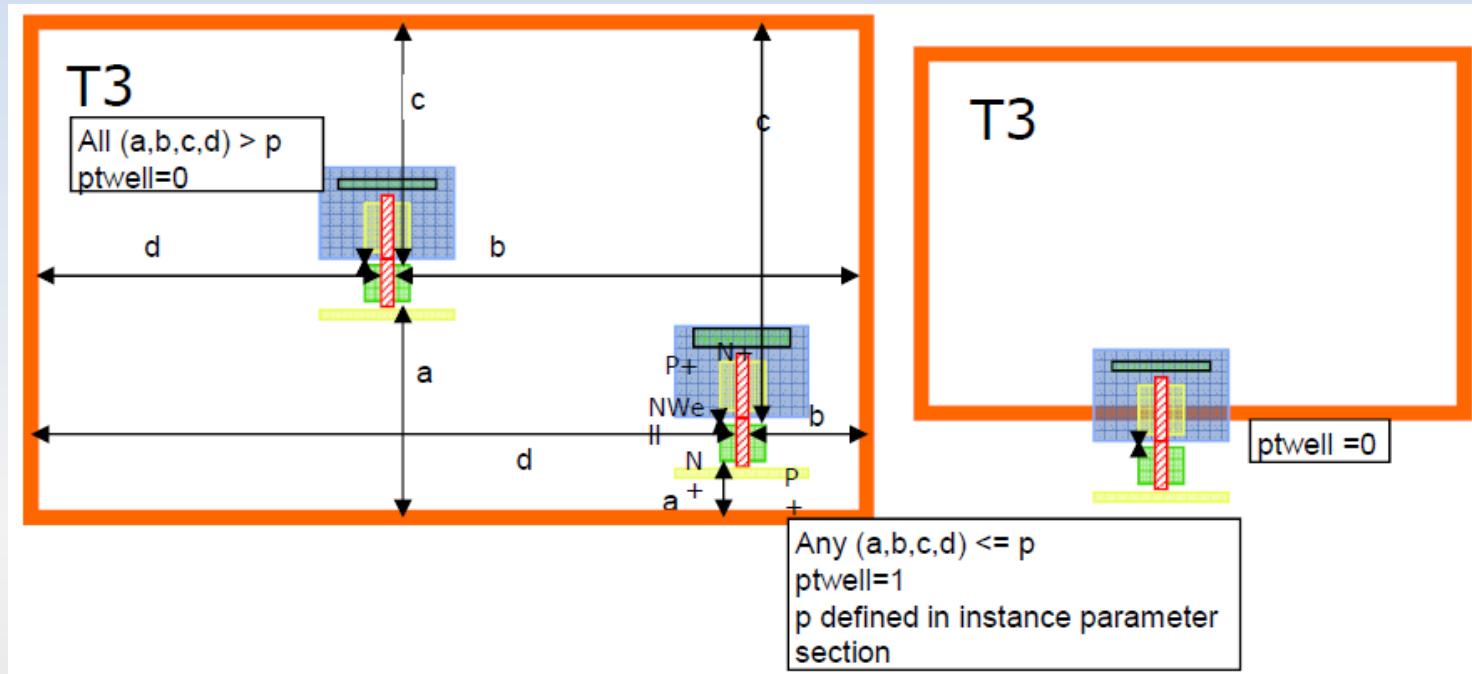
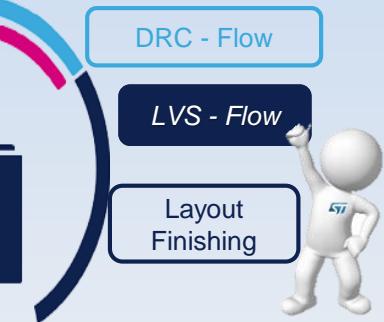
## Gate biasing :

- Gate covered by dedicated CAD layers are biased by a certain amount. This allows to tune the MOS behavior (less leakage, less parasitics,...)
- p\_la parameter is computed by LVS to reflect this biasing
- Parasitic extraction takes this effect into account
- Following layers are available :
  - POBIASP\_drawing4 : +4nm (+2nm per edge)
  - POBIASP\_drawing10 : +10nm (+5nm per edge)



- This parameter is extracted by default during LVS (switch not available in GUI)
  - LVS still uses original length for comparison

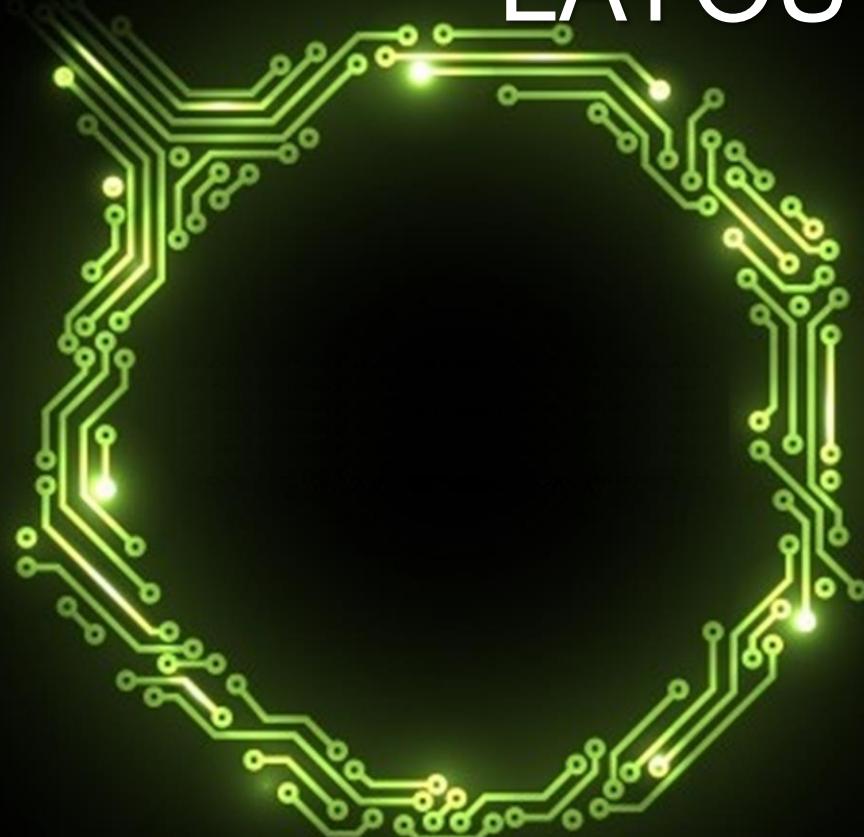
# Spice Accuracy Parameters : ptwell



This parameter depends on the distance between the MOS and the triple-well (T3).

- The ptwell instance parameter is extracted during LVS with the following conventions
- This parameter is extracted by default during LVS

# LAYOUT FINISHING



**PART IV**

**CMOS & Derivative PDK**

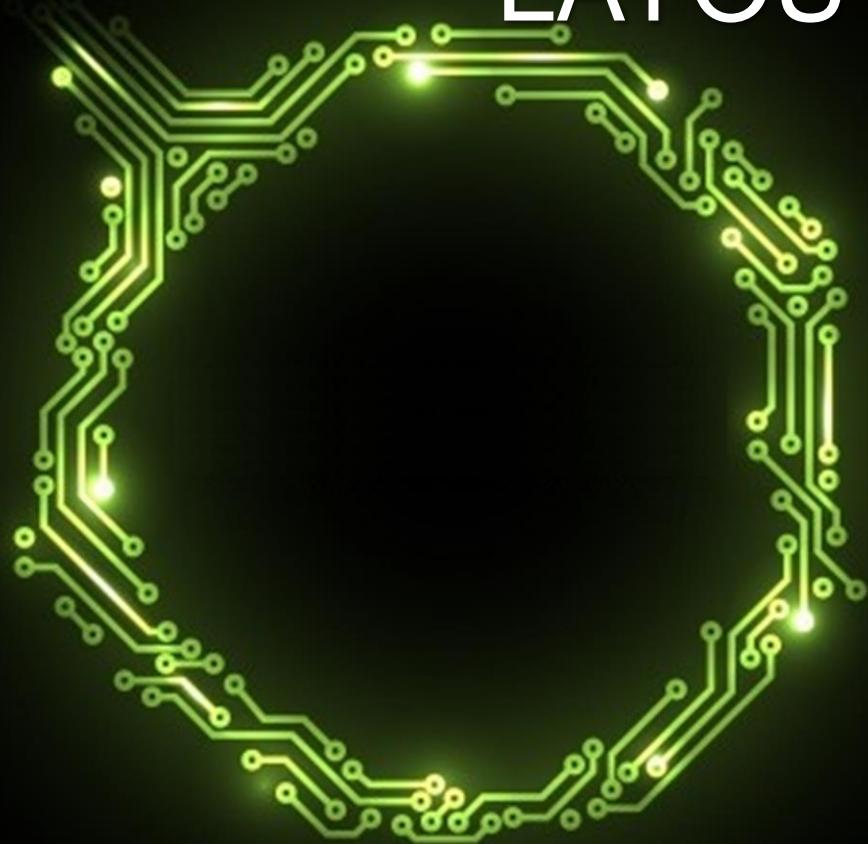


## Tool Capabilities



	Calibre	PVS
DRC	○	○
Antenna	○	○
DRC In-Desing	🚫	○
LVS	○	○
Tiling	○	○

# LAYOUT FINISHING



*Sealring  
Chip Name & ST Logo*

**CMOS & Derivative PDK**

# Layout finishing Flow

DRC - Flow

LVS - Flow

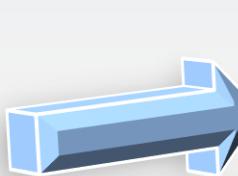
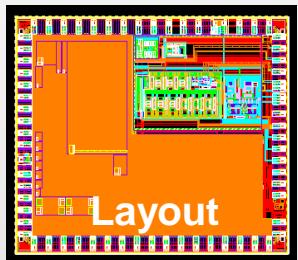
Layout  
Finishing



*How can I protect and identify my chip?*

**Sealring** = add protection for wafer cutting

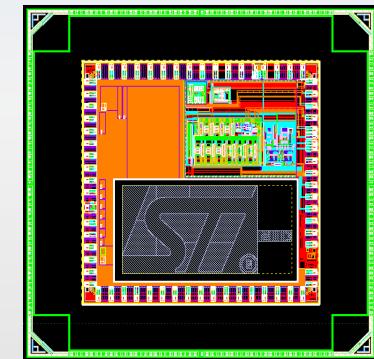
**ST Logo** = identify the chip

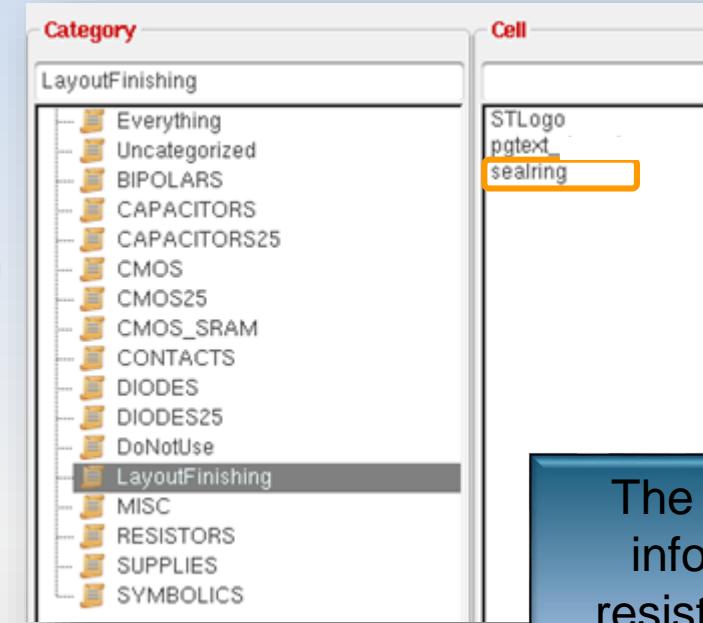
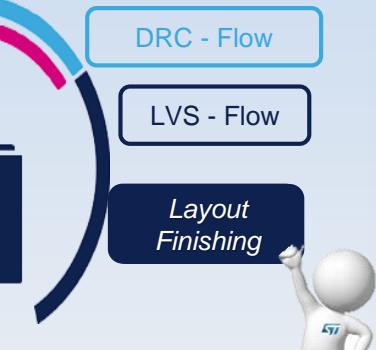


**Design Kit  
PCells**



**Process Rules**





The purpose of the seal ring is to give information for achieving a moisture resistant die, without the possibility that cracks generated in the scribe lane during the saw can propagate into the die.

Mandatory to any design.

No modification allowed.

It is mandatory to use the ST-specific sealring devices library

DRC - Flow

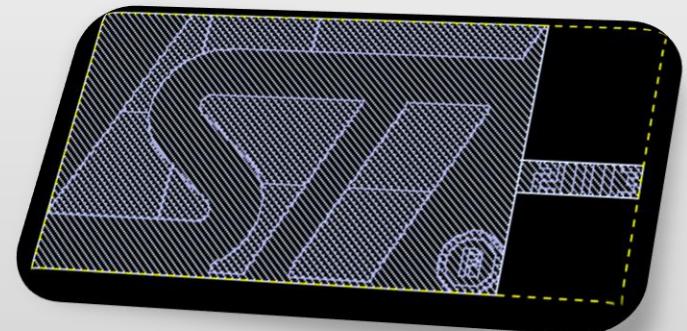
LVS - Flow

Layout  
Finishing



Mandatory in order to identify the chip:

- **Chip Name**
  - Pcell pgtext in ST\_C32\_addon\_DP library
- **ST Logo**
  - Pcell STLogo in ST\_C32\_addon\_DP library



# Layout Finishing: Sealring

70

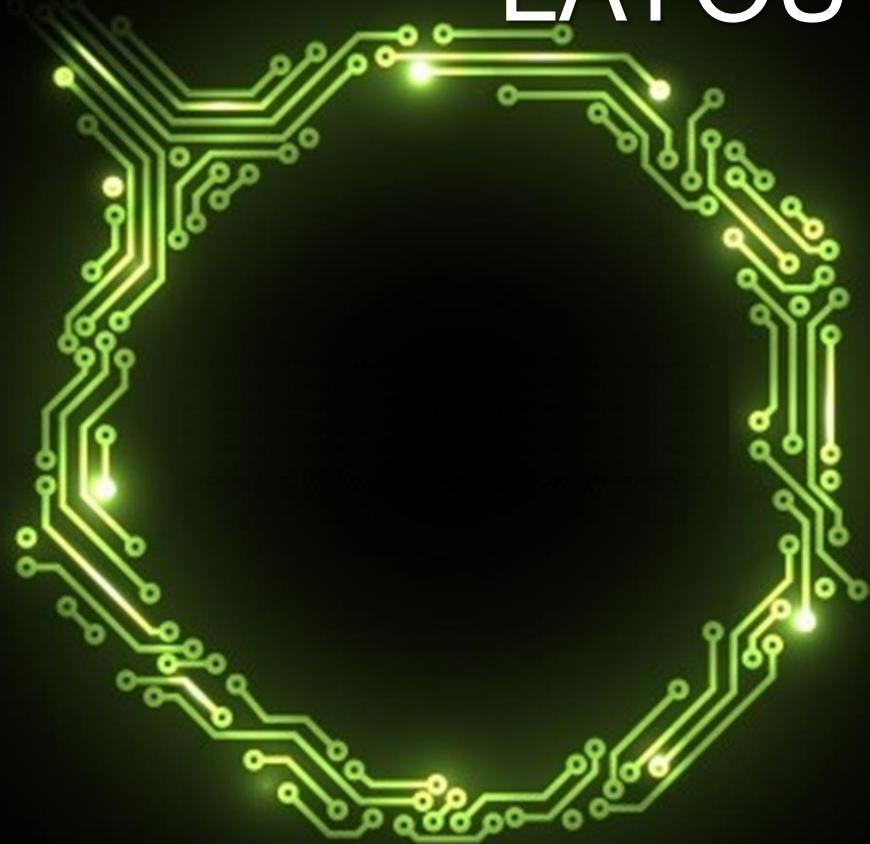
DRC - Flow

LVS - Flow

Layout  
Finishing



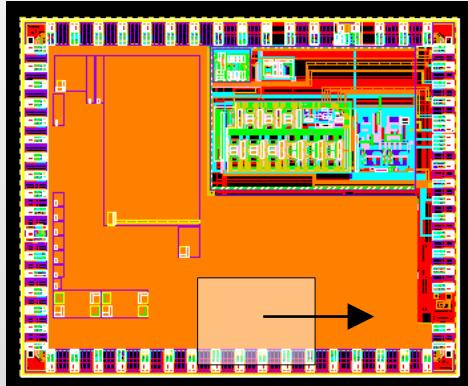
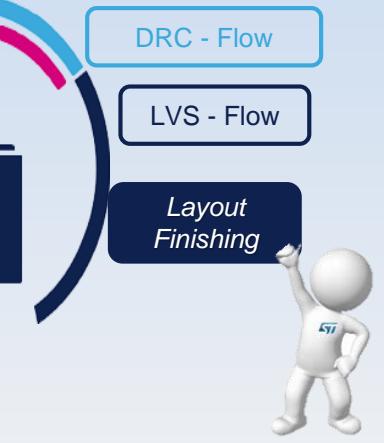
# LAYOUT FINISHING



*Tiling Processing*

**CMOS & Derivative PDK**

# Tiling procedure description



## Tiling Options



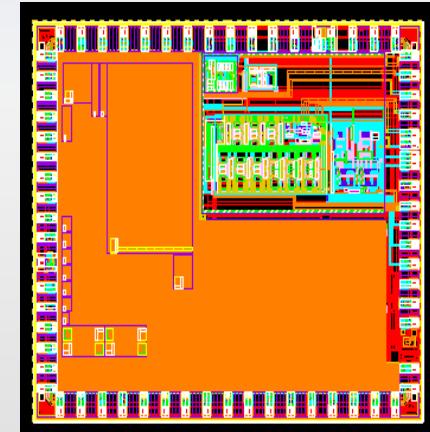
## Tiling Flow:

- FE layers (OD/PO)
- BE layers (metal)
- VIA-Tiling is performed
  - everywhere, even if no bumps present in layout

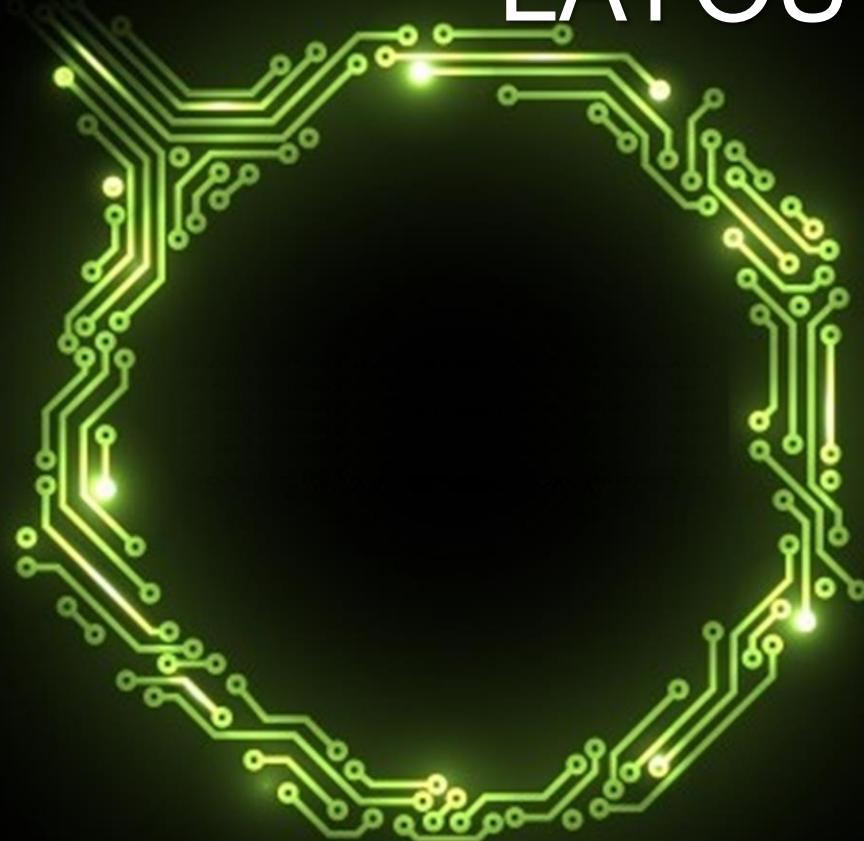


The Tiling step is done by **Designer**

- Possibility to use noTiles layer to prevent tiles generation



# LAYOUT FINISHING



*eMetro Tiling*

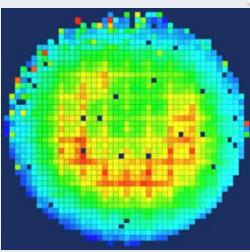
**CMOS & Derivative PDK**

# What is Embedded Metrology ?

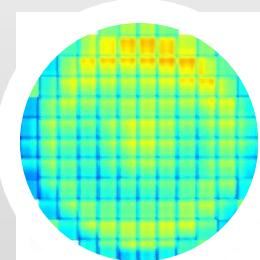
DRC - Flow

LVS - Flow

*Layout  
Finishing*



standard



with DoseMapper

- Formerly, structures for process monitoring were added in the scribe lines among chips
- Chips are bigger, lines are thinner:
  - Intrafield variability inside chip (proximity effect / circuit density etc...)
  - Intrafield variability chip to chip (Mask manufacturing uniformity, mask process on litho tools)
  - Intrawafer variability (different mask exposure, spatial uniformity of process chamber etc...)
  - wafer to wafer variability (inside one tool different process chambers can be used)
  - lot to lot process variability (drift over time / different tools used)

⇒ **Similar structures must now also be added inside chips**

eMetro allow Advance Process Control (APC) to minimize standard variation on

- Critical Dimensions (CD)
- overlay

The DoseMapper is one of the tool enabling this method

# When to place Embedded Metrology ?

DRC - Flow

LVS - Flow

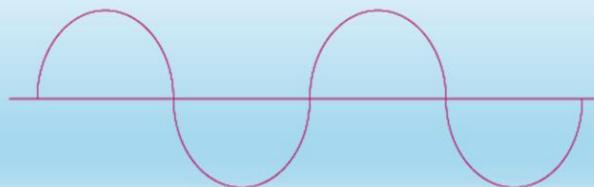
*Layout  
Finishing*



When SoC floorplanning is done, a DRC must be run to check whether emetro-insertion at Tiling-flow is required or not (to complement what has been already done at floorplanning level)

For digital IPs, there is usually no room left enough for eMetro cell insertion at SoC Layout Finishing step  
→ They must be considered during floor-planing

Analog



In pure digital SoC flow, from recent DesignPlatforms, the eMetro insertion is automatically managed by the Floorplanner-tool before place&route step (refer to suitable Digital Implementation Kits)

# When to place Embedded Metrology ?

DRC - Flow

LVS - Flow

Layout  
Finishing



eMetro can then be managed by the tiler

For full custom physical design, these cells are inserted during layout finishing in empty spaces

Digital



or they can be manually instantiated by picking them (FE or BE) from emetro library.



- **eMetro dramatically increase yield** by allowing an accurate process monitoring leading to an important CDs standard variation minimization



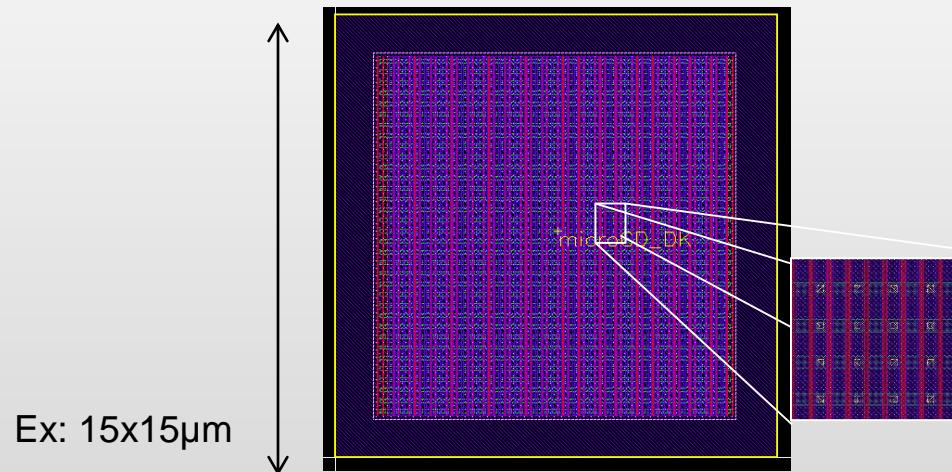
- eMetro to be placed **as early as possible** in the design-phase



# About eMetro structures

78

- their instantiation can be enabled in the tiler GUI
- they can be flipped and rotated
- they may be swapped by more efficient ones at CAD2MASK level
- they can use FE or BE layers

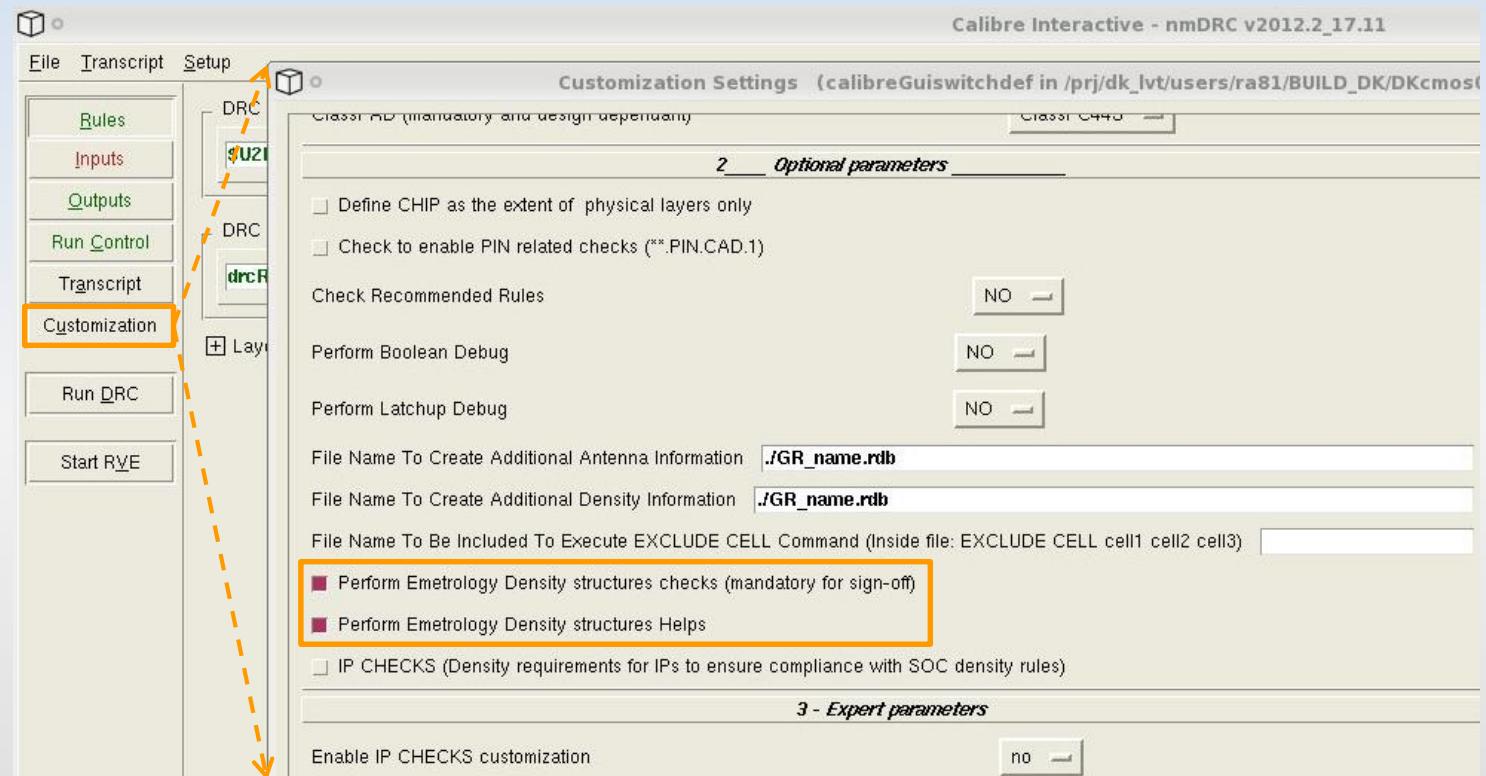


# EMET : GUI DRC Customization

DRC - Flow

LVS - Flow

Layout  
Finishing

**Calibre Interactive - nmDRC v2012.2\_17.11**

**Customization Settings (calibreGuiswitchdef in /prj/dk\_lvt/users/ra81/BUILD\_DK/DKcmos)**

**2 - Optional parameters**

- Define CHIP as the extent of physical layers only
- Check to enable PIN related checks (\*\*.PIN.CAD.1)

Check Recommended Rules

Perform Boolean Debug

Perform Latchup Debug

File Name To Create Additional Antenna Information **./JGR\_name.rdb**

File Name To Create Additional Density Information **./JGR\_name.rdb**

File Name To Be Included To Execute EXCLUDE CELL Command (Inside file: EXCLUDE CELL cell1 cell2 cell3)

Perform Emetrology Density structures checks (mandatory for sign-off)

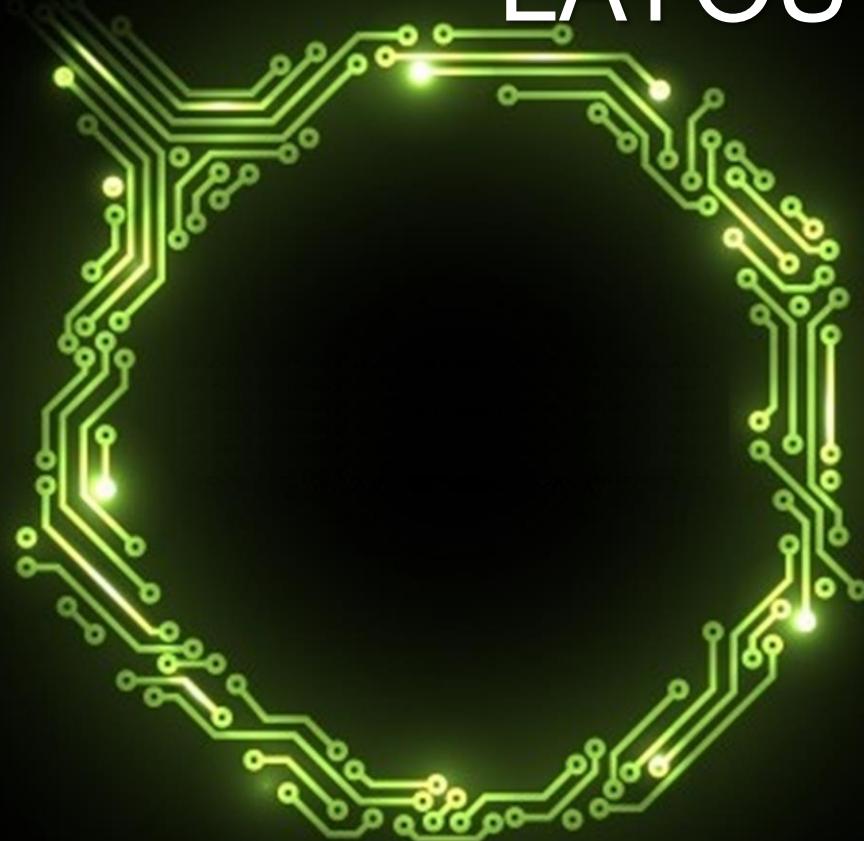
Perform Emetrology Density structures Helps

IP CHECKS (Density requirements for IPs to ensure compliance with SOC density rules)

**3 - Expert parameters**

Enable IP CHECKS customization

# LAYOUT FINISHING



*Via Tiling*

**CMOS & Derivative PDK**

DRC - Flow

LVS - Flow

 Layout  
Finishing

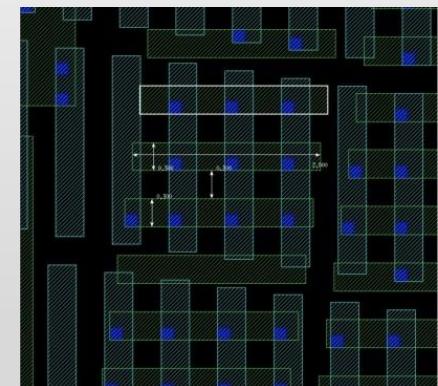

**What :** Purpose of via tiling is to increase via density in order to have uniform distribution across the circuit/SoC.

**Why :** Pads from circuit/SoC using Flip-Chip/Bumping packages must be reinforced to avoid packaging issues ( reduce mechanical stress effect )



*Warning: Density on a common layer level rises from 20% to 43%*

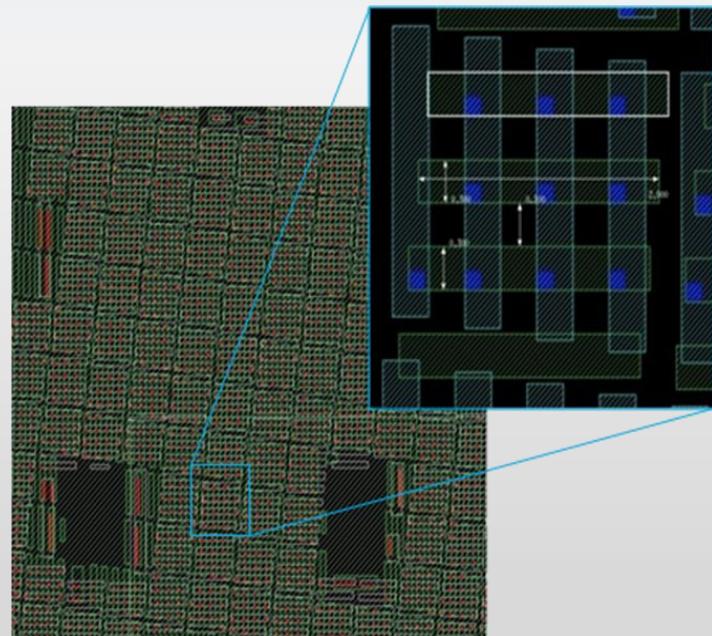
→ metal fill uniformity appears to be also important : via matrix only would have lower mechanical resistance (metal stripes are required)





Via tiling purpose is to prevent delamination effect in low k materials (so no dummy vias for thick metals). It is a must for designs containing flip chip pads.

Via tiling is done at the same time as Metal tiling step



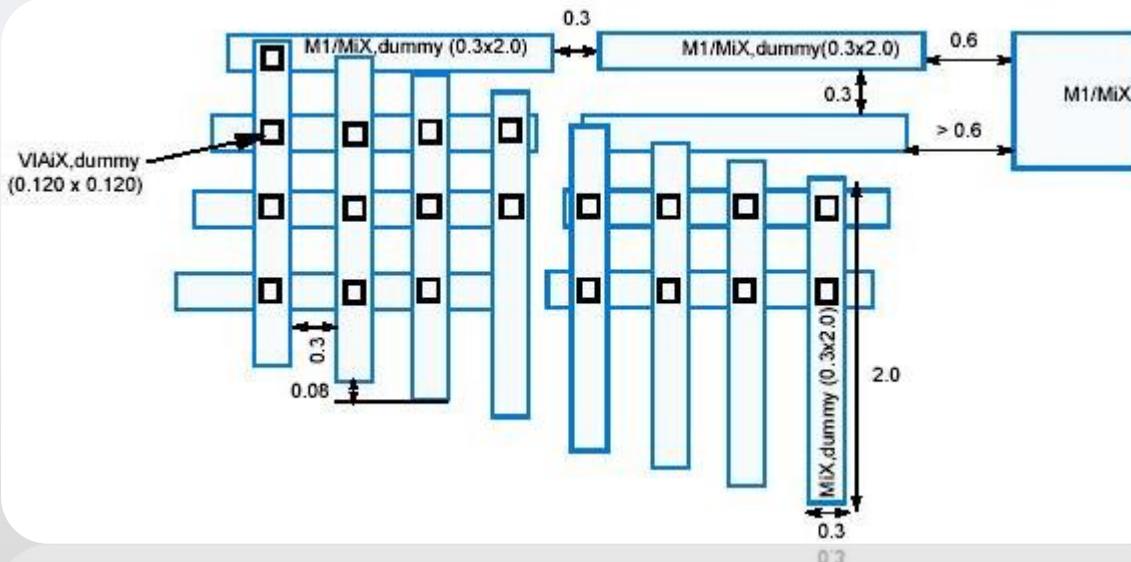
DRC - Flow

LVS - Flow

*Layout  
Finishing*



Metal Via-Tiling has been introduced in several technologies (H9A, 65nm, 45nm, 32nm) in order to reinforce the Flip Chip / Bumping pads, to avoid « white bump » effect seen on 8500 (45nm)



New dummy VIA layers

VIA-Tiling performed everywhere, even if no bumps

DRC - Flow

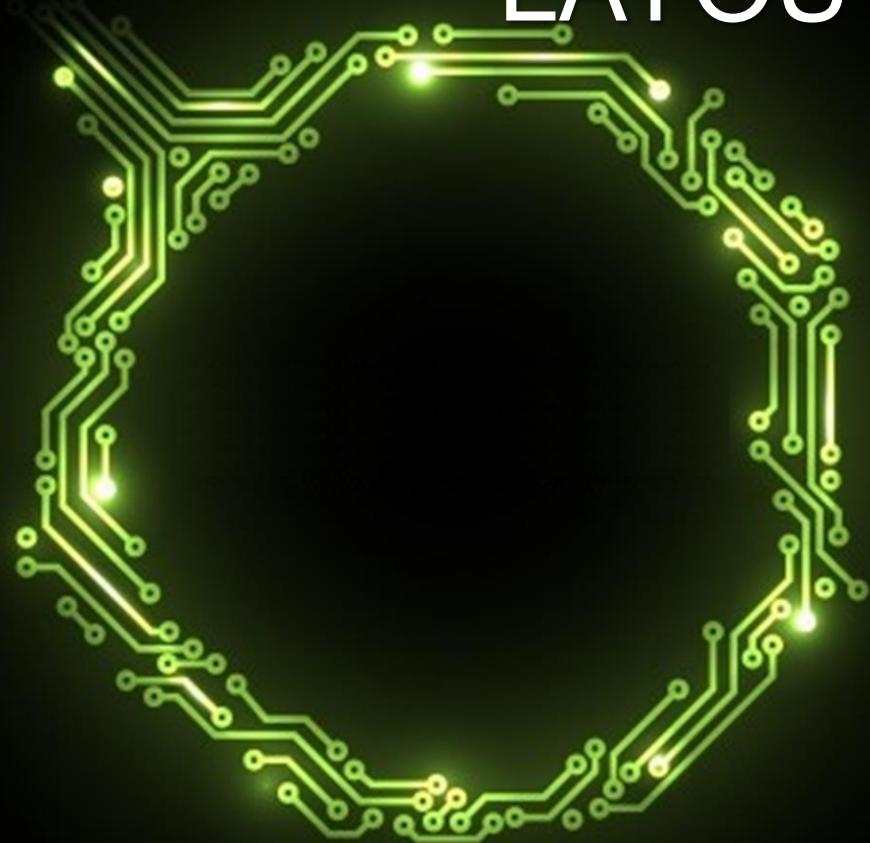
LVS - Flow

 Layout  
Finishing


Use exclusion layers to forbid automatic tiling on some levels

Exclude layer	Description
<b>RX;exclude</b>	Exclude area for active (RX)
<b>PC;exclude</b>	Exclude area for poly (PC)
<b>Mx;exclude</b>	Exclude area for metal tiling
<b>IA;exclude</b>	Exclude area for first thick metal level (IA)
<b>IB;exclude</b>	Exclude area for second thick metal level (IB)
<b>LB;exclude</b>	Exclude area for alucap (LB)

# LAYOUT FINISHING



*Run Tiling Processing*

**CMOS & Derivative PDK**



# Smart Tiling calibre gui

86

- Refer to the README documentation:  
**\$PDKITROOT/DATA/SMART\_TILING/CALIBRE/README**
- mkdir -p smart\_tilingRunDir
- Set configuration files from PDK (Runset and DRC customization)
  - setenv U2DK\_CALIBRE\_SMART\_TILING\_DECK  
**\$PDKITROOT/DATA/SMART\_TILING/CALIBRE/calibresmart\_tiling\_cgi**
  - setenv MGC\_CALIBRE\_SMART\_TILING\_RUNSET\_FILE  
**\$PDKITROOT/DATA/SMART\_TILING/CALIBRE/calibreGuirunsetsmart\_tiling**
  - setenv MGC\_CALIBRE\_SMART\_TILING\_CUSTOMIZATION\_FILE  
**\$PDKITROOT/DATA/SMART\_TILING/CALIBRE/calibreGuiswitchdefsmart\_tiling**
- launch calibre gui
  - calibre -gui -drc -runset \$MGC\_CALIBRE\_SMART\_TILING\_RUNSET\_FILE - drcLayoutPaths <full gds path> -drcLayoutPrimary <topcell> -batch > tiling.log & or
  - calibre -gui -drc -runset \$MGC\_CALIBRE\_SMART\_TILING\_RUNSET\_FILE &

# Tiling and/or EMET Generation

DRC - Flow

LVS - Flow

Layout  
Finishing



It is highly recommended to perform tiling in 2 steps:

- First generate EMETs if those instantiated at floorplan are not enough
- Then generate tiles over merged GDS (original db + EMETs)

To promote dry runs, you also have the ability to enable both features at the same time

*Smart tiling switches Help is available via unix command: calibreSTSswitchesHelp -smart\_tiling*

*Embedded Metrology structures have to be placed by Calibre BEFORE RX, PC and Metal tiling.*

*Activate EMETRO Tiling if DRC for EMETRO is not clean*

EMETRO Tiling

Enable tiling with microCD FE emetro structure  
 Enable tiling with microCD BE emetro structure  
 Enable tiling with eSCD2 emetro structure  
 Enable tiling with eSCD3 FE emetro structure  
 Enable tiling with uAIM2 FE emetro structure  
 Enable tiling with uAIM1 FE emetro structure  
 Enable tiling with uAIM4 FE emetro structure  
 Enable tiling with uAIM3 FE emetro structure  
 Enable tiling with eSCD1 FE emetro structure  
 Enable tiling with uAIM5 FE emetro structure  
 Enable tiling with uAIM6 FE emetro structure  
 Enable tiling with uAIM7 FE emetro structure

Define Calibre EFFORT for eMetro tiling(value below 4 recommended as this option increases runtime)

Define the number of tiling passes for each eMetro (value 1 recommended as it increases the runtime)

EMET  
Placement



# Tiling and/or EMET Generation

**Tiling**

Front-End/Back-End Tiling  YES

1 - Metal option: 6UTx\_2U2x\_2TRx\_LB

2 - Tiling Layers  YES

Front End layers tiling

- Enable tiling with PMOS<sub>fill</sub> and NMOS<sub>fill</sub> cells
- Enable tiling for PC
- Enable tiling for RX

Back End layers tiling  YES

- Enable tiling for M1
- Enable tiling for M2
- Enable tiling for M3
- Enable tiling for M4
- Enable tiling for M5
- Enable tiling for M6
- Enable tiling for B1
- Enable tiling for B2
- Enable tiling for IA
- Enable tiling for IB

Enable tiling with squares

Enable tiling with stretchfill shapes

Enable tiling with fillOPC shapes

Enable tiling to achieve Gradient density constraints. Increases runtime (may be disabled on smooth chips facing no density gradient DRC violation)

Enable tiling for LB

Via tiling (mandatory when flipchip pads are used)  YES

- Enable tiling for V1
- Enable tiling for V2
- Enable tiling for V3
- Enable tiling for V4
- Enable tiling for V5
- Enable tiling for W0
- Enable tiling for W1

Main routing direction (M1)  HORIZ

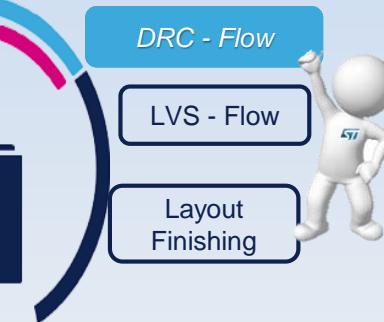
Run Via Tiling on tiles present in initial input GDS

Flexibility given on layers to be tiled:

- per group (FE/BE/via)
- per layer

Main routing direction helps the tiler to reach higher densities Ability to generate dvia between generated fills and pre-existing fills

Ability to generate dvia between generated fills and pre-existing fills



- Refer to the README documentation
  - `$PDKITROOT/DATA/SMART_TILING/CALIBRE/README`
- Customization switches in  
**`$PDKITROOT/DATA/SMART_TILING/CALIBRE/calibresmart_tiling.ctrl`**
  - Retrieve and copy the file in the working directory
  - Update it with the appropriate gds path and Topcell name
    - “myCell” fields to changed
- Launch calibre DRC
  - `calibre -hier -drc calibresmart_tiling.ctrl > smart_tiling.log &`



# Smart Tiling: merge gds

90

- Two gds files containing emetro and tiling are created in smart\_tilingRunDir
  - <name>.emetro.gds
  - <name>.tiles.gds
- Need to merge both new created gds with the original one
- To merge, launch :
  - `calibredrv -a layout filemerge -in <original_layout_name> -in <tiles_layout_name1> -in <tiles_layout_name2> -in <tiles_layout_name3> -out <merged_layout_name> -createtop TOP`

- Identify the sign-off layout verification flow
- Configure the CAD tools for DRC, LVS
- Perform the decomposition
- Identify the Layout Finishing steps





Thanks for your attention