Layout Finishing and SoC Tiling: **ECO/Metal fix**

February 14th, 2013

CMOS and derivative PDK

life.augmented

Program Management & Services / Process Design Kit

Metal fix/ECO (with eMetro/via-tiling) : the problem (1/3)

- metal fix could impact EMET structures
- metal fix could impact metal;fill supporting via;fill
- metal fix can lead to abusive re-order of via masks

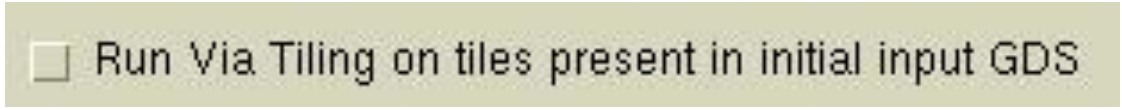
} Metal fix to be done with caution

- During metal fix on Mi
  - avoid to remove dvias, otherwise you might need to re-order also VIAi and/or VIAi-1 masks
  - avoid eMetro: they are useless if modified

Metal fix/ECO (with eMetro/via-tiling) : update in **design db** (2/3)

• If the metal fix is *small*: update *manually* local fill/dvia
(using calibreDRV: PROMOTE macro explained in next slides)

• if there is a need to re-generate back automatically V2/M3 (for ex.), user must select in the GUI-tiler M1/V2/M3, and activate the switch allowing the tiler to generate selected dvia between pre-existing fill, and Mi (also to be generated):



Nb:
• metals above/below must also be selected to generate in-between dvia
• you may disable fillOPC in Expert Parameters in such occasion
• this behavior will be improved in next DKs so that the designer can select only the needed dvia
• there is no feature to generate metal;fill on pre-existing via;fill

Metal fix/ECO (with eMetro/via-tiling) : update in **PnR db** (3/3)

- By making a fix in the PnR db, you take the risk to overlap eMetro or dvia.
- DK team works on the following specifications:
  - in CalibreDRV, select the minimum set of shapes to be taken care of by the router for Mi fix
  - run COORDINATES macro : it will export their coordinates
  - you will then be able to import them as obstruction layers (sized) in PnR db
    - may be done in an iterative way if several metals are involved in the fix
  - finally, you will need to run the PROMOTE macro on the new gds in calibreDRV to delete/notch fills involved in DRC errors due to this fix.

Tiling flow to ease metal fix/ECO (1/3)

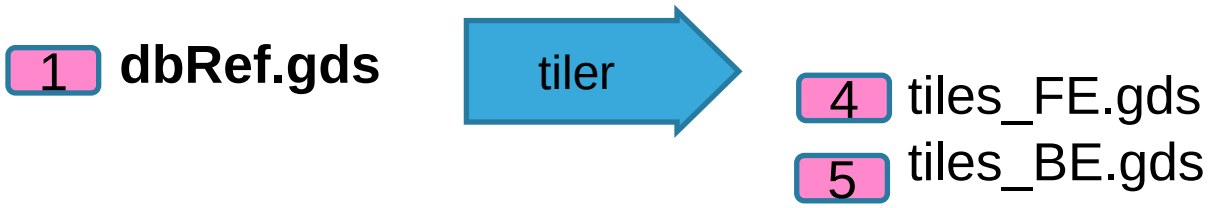
- When you tile your design before delivery, you archive the generated gds:
  - ✓ design.eMetro_FE.gds (if eMetro tiling enabled)
  - ✓ design.eMetro_BE.gds (if eMetro tiling enabled)
  - ✓ design.tiles_FE.gds
  - ✓ **design.tiles_BE.gds**

- Depending on your working model, you may want to freeze:
  - ➢ FE layers only (eMetro and tiles)
    (non dense design allowing easy BE emetro tiling)
  - ➢ FE Layers plus BE eMetro (most of the time)
    (dense design not allowing easy BE emetro tiling)

- Once your fix is done, the only gds to be re-generated by the tiler is usually **design.tiles_BE.gds**
- This will reduce tiling runtime, and ease final XOR between 1st and 2nd deliveries (no difference expected on FE layer for ex)

metal fix/ECO with floorplaned eMetro (2/3)

Basic flow



1 + 4 + 5 = **dbRefTiled.gds** : sent for PG

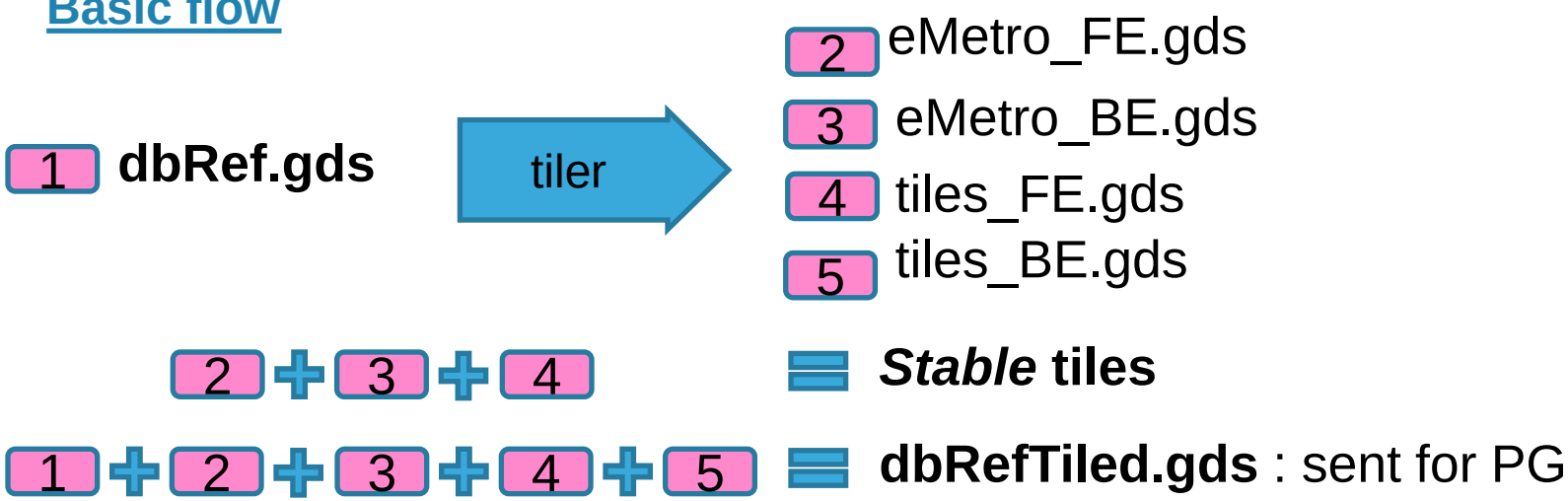
If need for a metal fix

1’ + 4 + 5’ = **dbRefFixedTiled.gds** : sent for PG

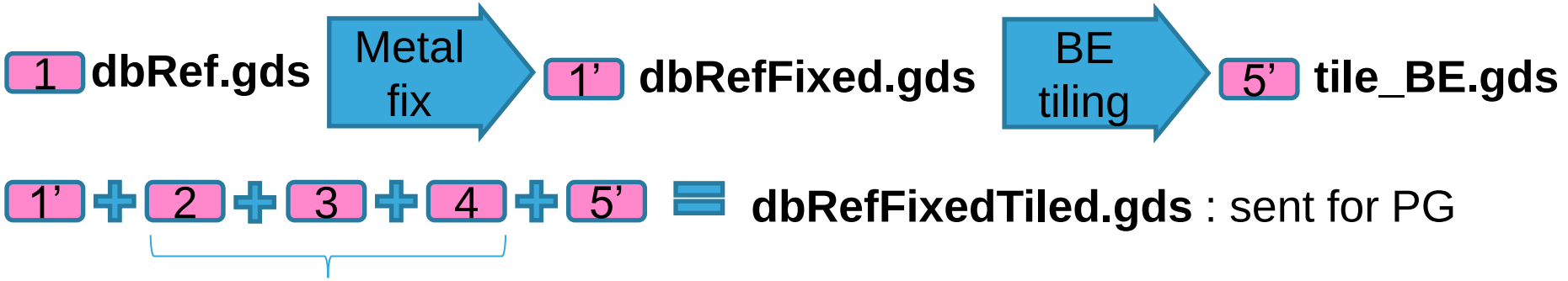
Re-used from the first delievry (reduce runtime/ease XOR)

metal fix/ECO with tiled eMetro (3/3)

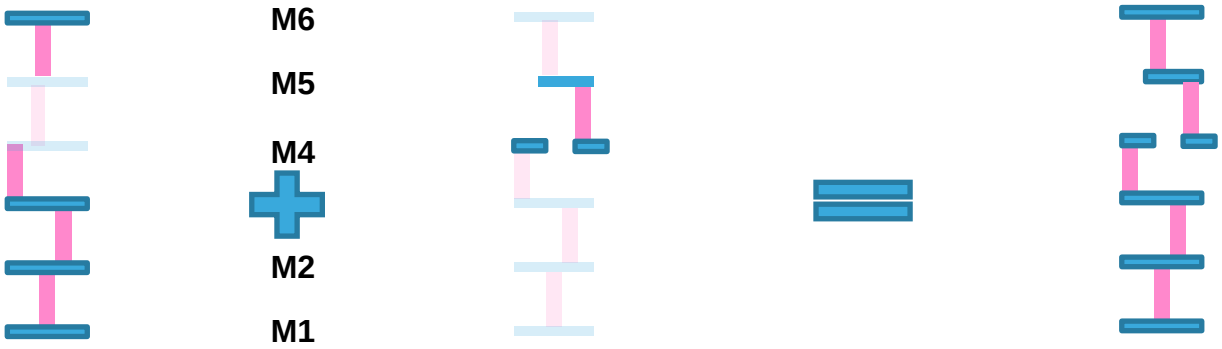
Basic flow



If need for a metal fix



Manage DRC when several cuts deliveries occur



Gds delivery for 1st cut

Gds delivery for 2nd cut
(metal fix on M4/via4/M5)

With CalibreDRV,
designers can easily emulate future silicon to run DRC,
based on masks kept from 1st delivery,
and masks re-ordered following 2nd delivery

1/ method 1: From CalibreDRV gui
- For both cuts, hide layers to be removed (here shadowed), then: File -> Export Layout …-> Layers : Visible
- Merge the gds you get then from the 2 cuts: File -> Merge
- Run the sign-off DRC on this GDS, aligned with ordered masks.

2/ method 2: With online command (under development, expected with CalibreDRV 2013.2)
- layout file merge –in cut1.gds –exclude_layer M4 V4 M5 –in cut2.gds –include_layer M4 V4 M5 –out output_file.gds
- Run the sign-off DRC on this GDS, aligned with ordered masks.

How to manage a metal-fix (shape deletion)

- Today, a metal fix consists in:
  - Shapes deletion on a GDS
  - Based on calibredrv tool
  - Usage:
    - tiling removal to correct density max
    - delete metal tiling shapes to perform metal routing fix
  - Example 1: fix for a maximum density of B1 layer
  - Example 2: metal fix without impacting via tiling

Editing Polygons in hierarchy

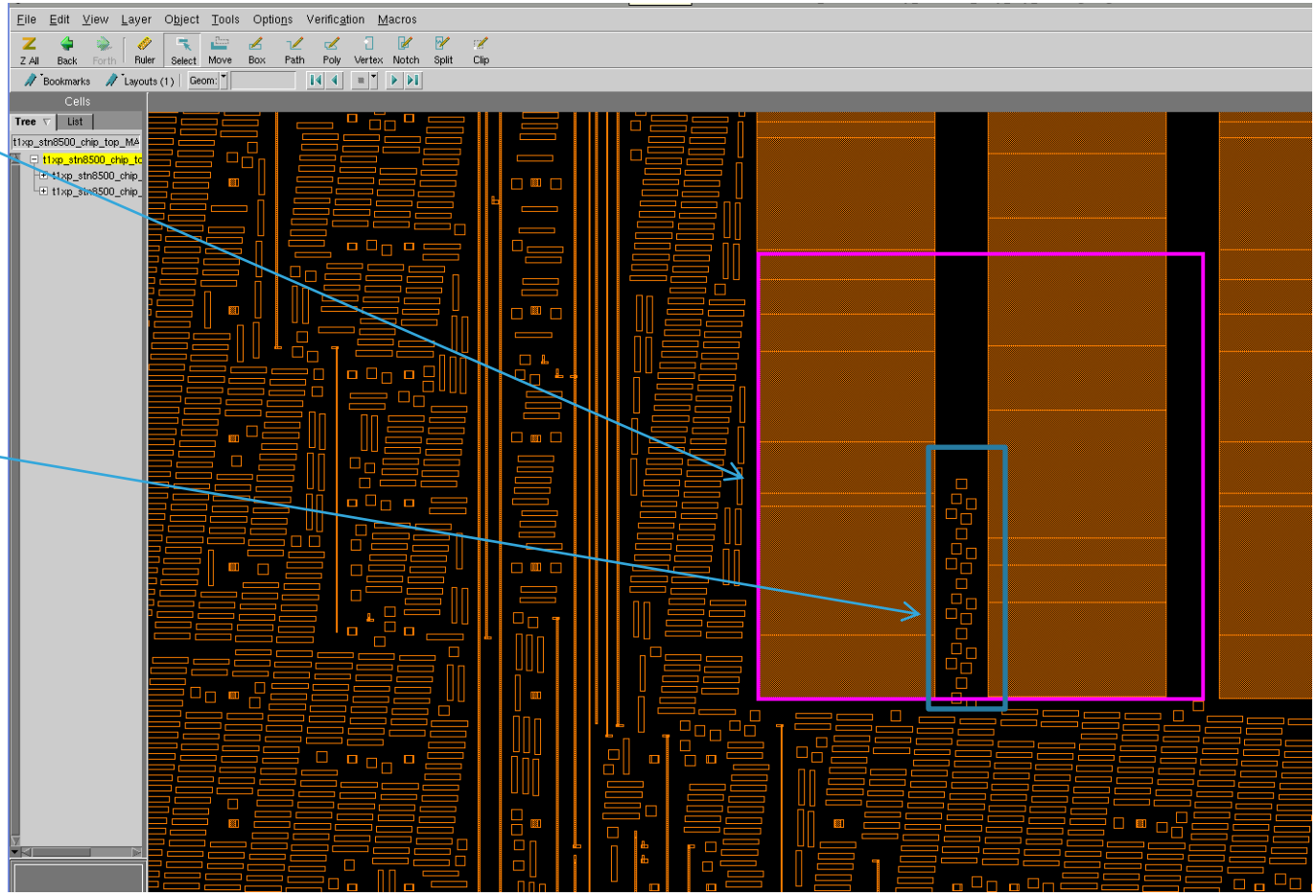
- Definitions
  - Current cell: cell currently displayed in Calibre DESIGNrev
  - Context cell: cell which directly contains the polygon to be edited
- Goal: A hierarchical GDS being provided, user wants to edit (e.g. delete or notch) some unique instances of hierarchical polygons from current cell, i.e. from higher hierarchy level (typically from topcell).
- Problem: Polygons can only be edited in their context cell, and that case any modification will apply to any instance of the cell.
- Solution: First use the Calibre DESIGNrev PROMOTE macro to promote the polygons to be edited up to the current cell, then normally edit these unique polygons directly in current cell.

Application example1: B1 max density

- On below ex, max density error is flashing due to tiling. Because of shapes of B1/fill between big B1 shield.

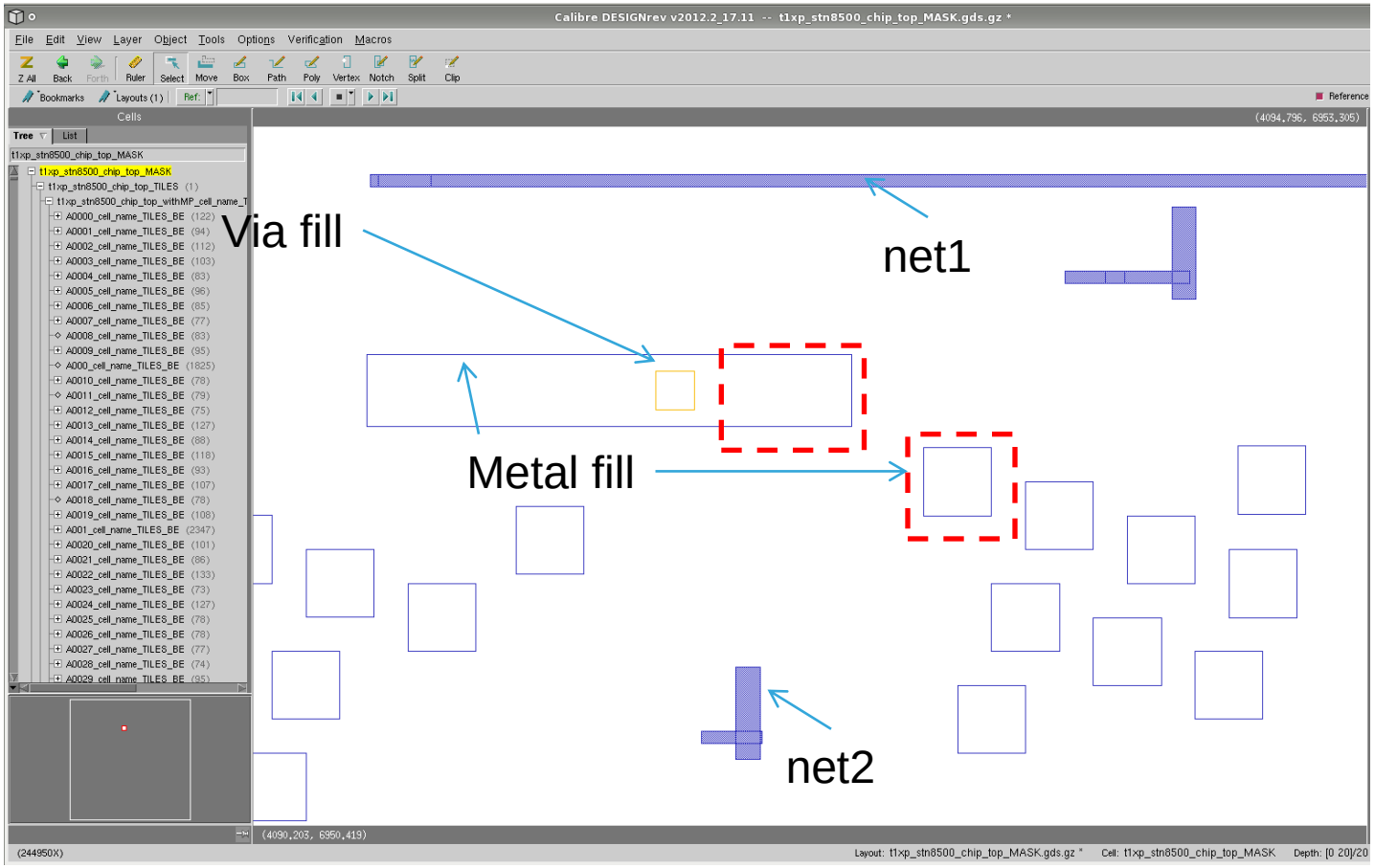
Calibre Error window

Shapes to delete

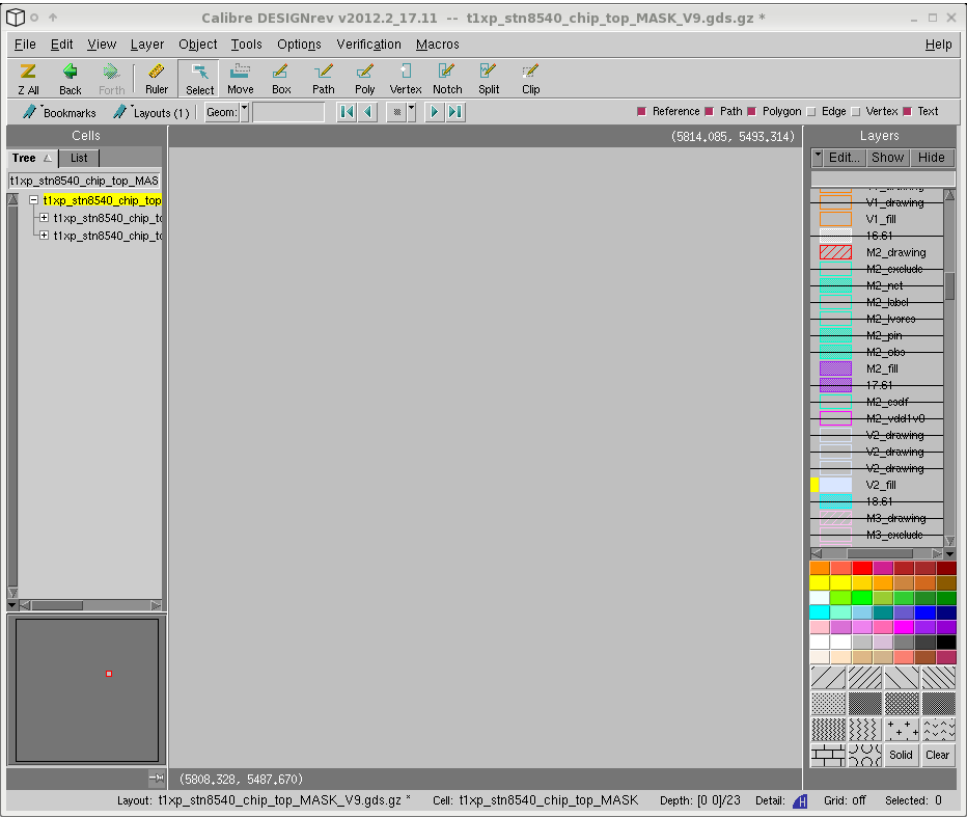


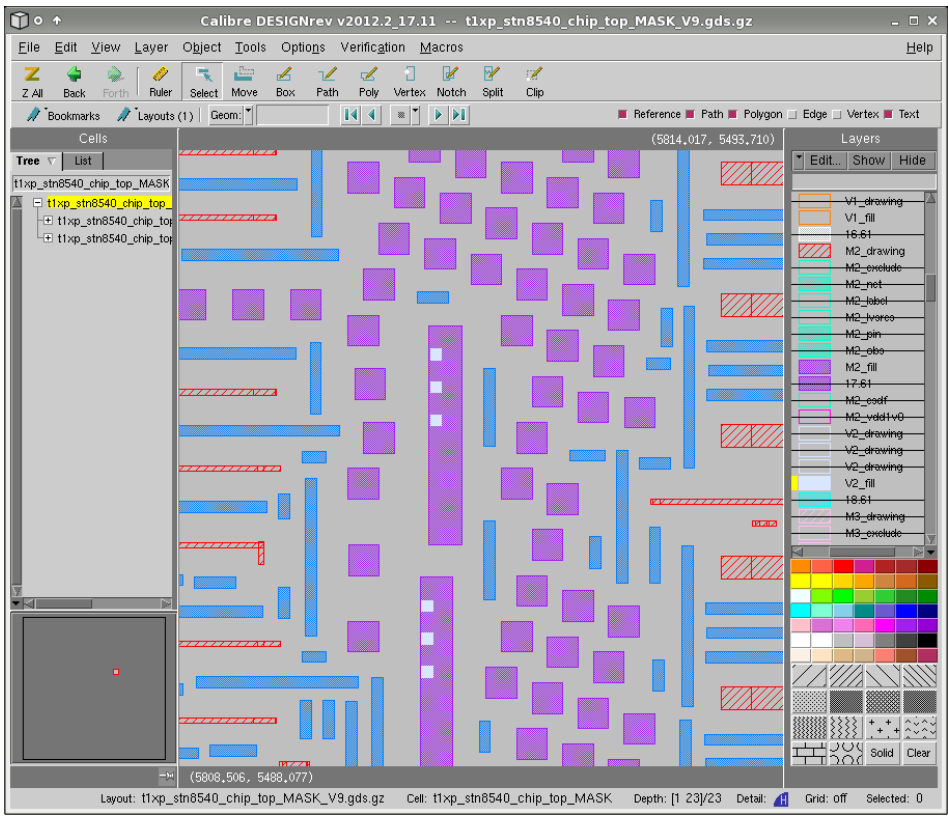
Application example2: metal fix/ECO

- Connection of net1 to net2 requires to delete one tiling shape and notch another tiling shape (in red)



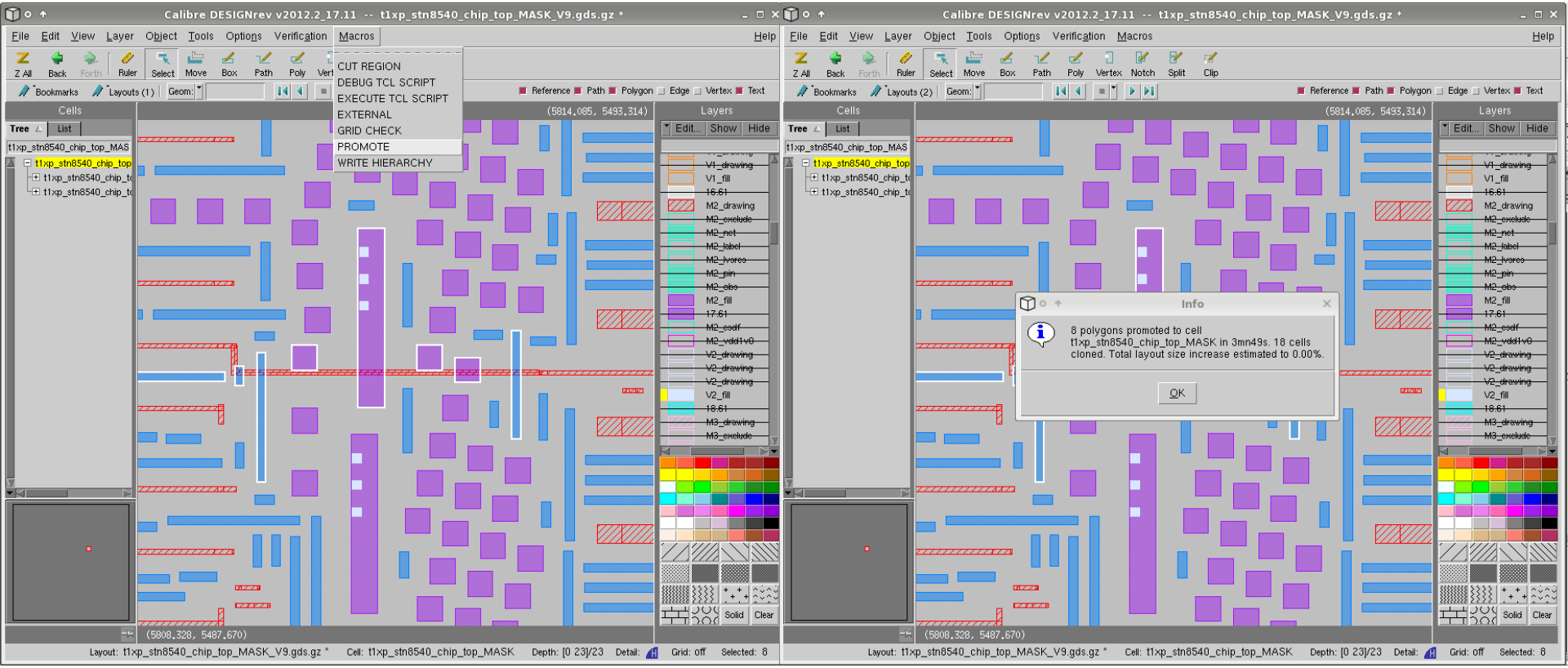
Layout before promotion





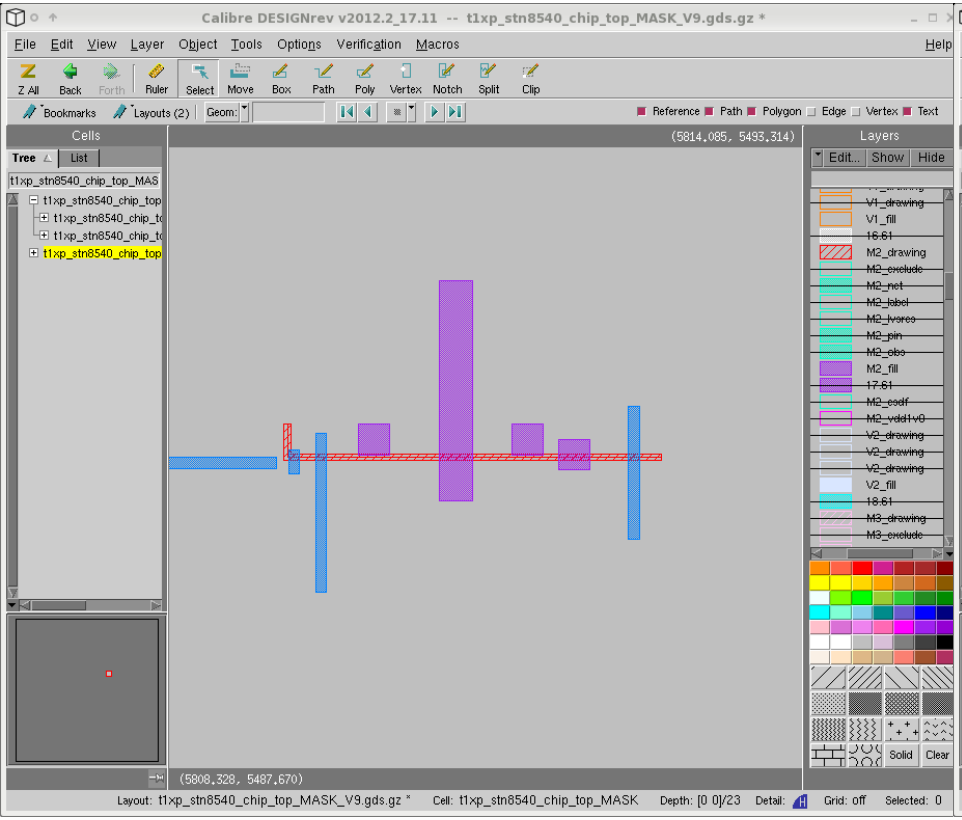
- Current cell view is topcell, zooming in a particular region, displaying only V1/M2/V2 layers
  - On the left, Depth [0 0]/23 → no polygon in these layers
  - On the right, Depth [1 23]/23 → there are polygons from multiple layers in levels #1-23

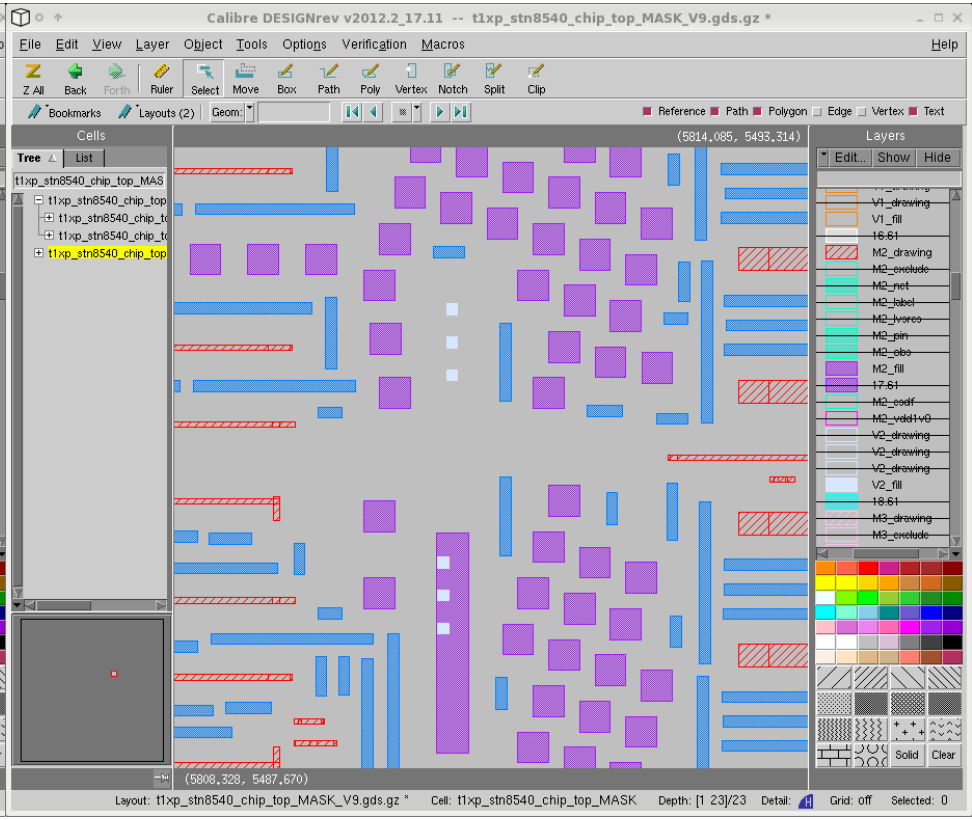
Promotion



- Left screen: The metal fix (added M2 wire) creates DRC violations with original tiles.
  For the DRC to be fast, you can run it with « select area » or Calibre RealTime.
  Select polygons to be edited (Shift+Left Mouse Button or Edit>Select Region) and Run PROMOTE (in menu MACRO)
  - 8 polygons selected (in white)
- Right screen: Result of PROMOTE
  - 8 polygons promoted to current cell (i.e. topcell in this example), 18 cells needed to be cloned (hierarchy is automatically processed and optimized)
  - Runtime was <4mn and estimated total layout size increase is negligible
- Tip: you can size selected dvias to safely draw your fix by anticipating min space violations: use CalibreDRV->Layer->Size

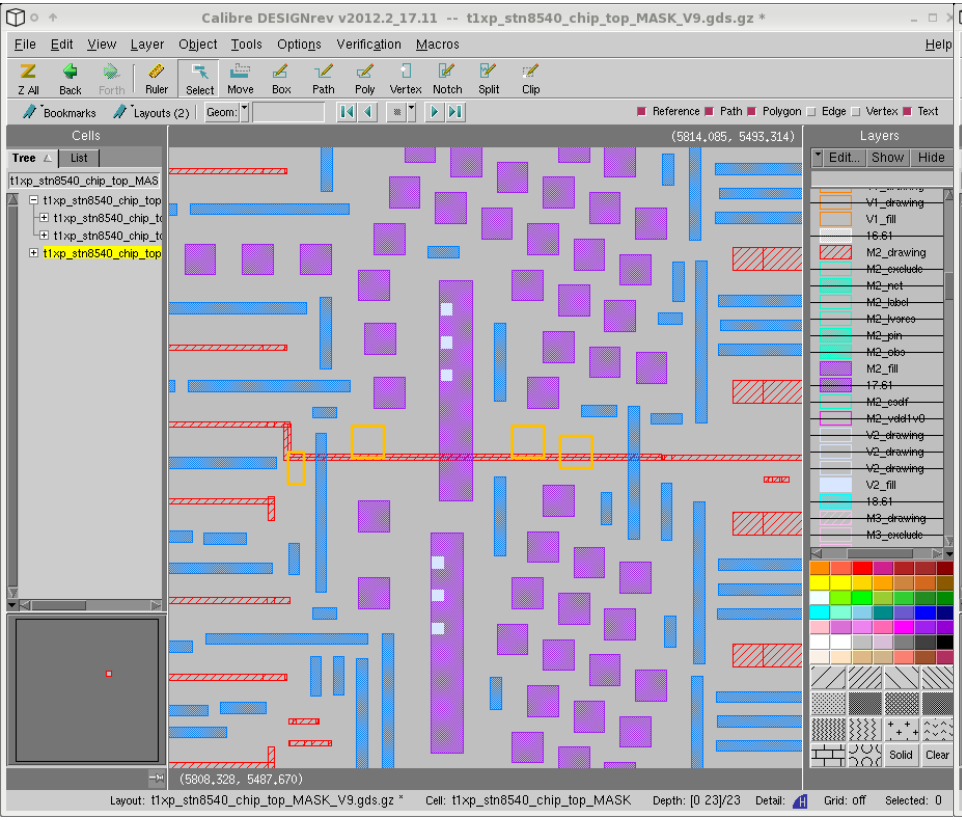
Layout after promotion

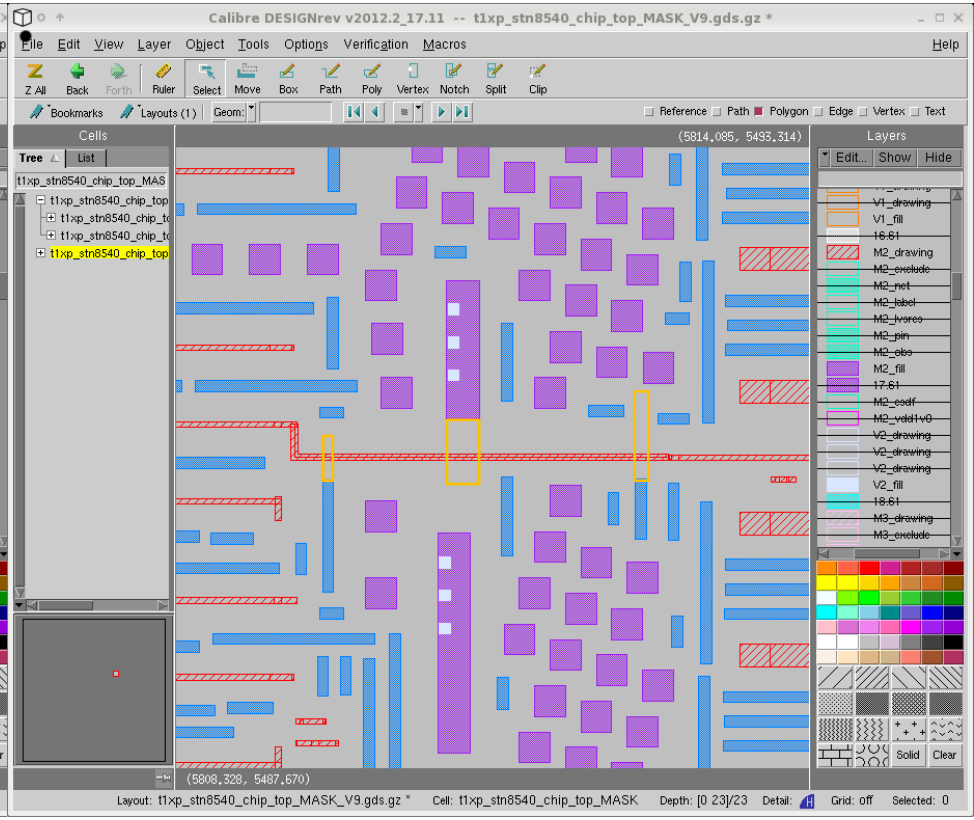




- Layout is unchanged from a flat perspective but
  - On the left, Depth [0 0]/23 → 8 polygons created (in addition to M2 wire)
  - On the right, Depth [1 23]/23 → 8 polygons deleted
  - A flat XOR would return 0 difference between pre-/post-PROMOTE layouts

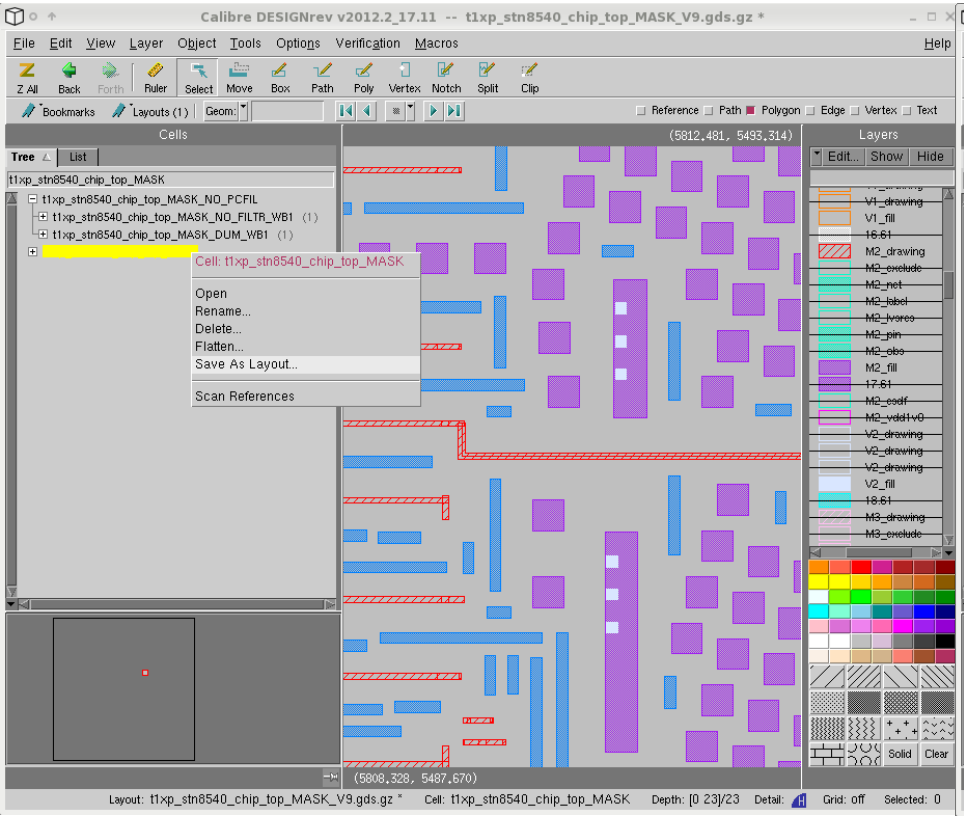
Manual ECO

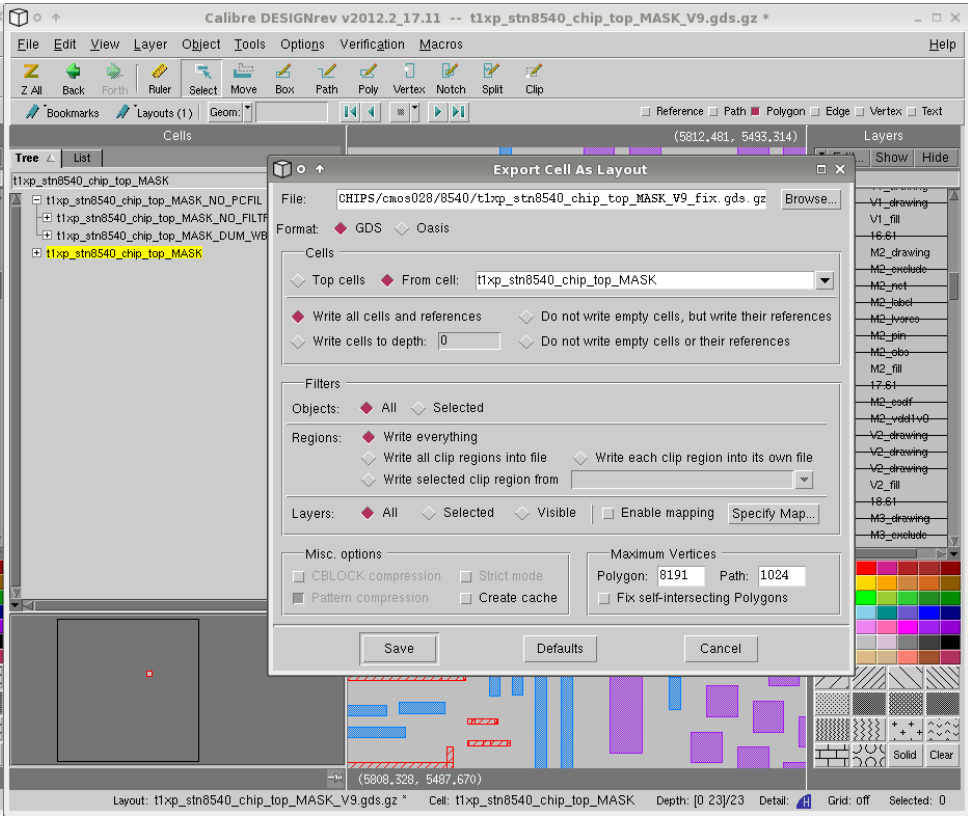




- Left screen: delete tiling shapes that need to be
- Right screen: notch tiling shapes that can be

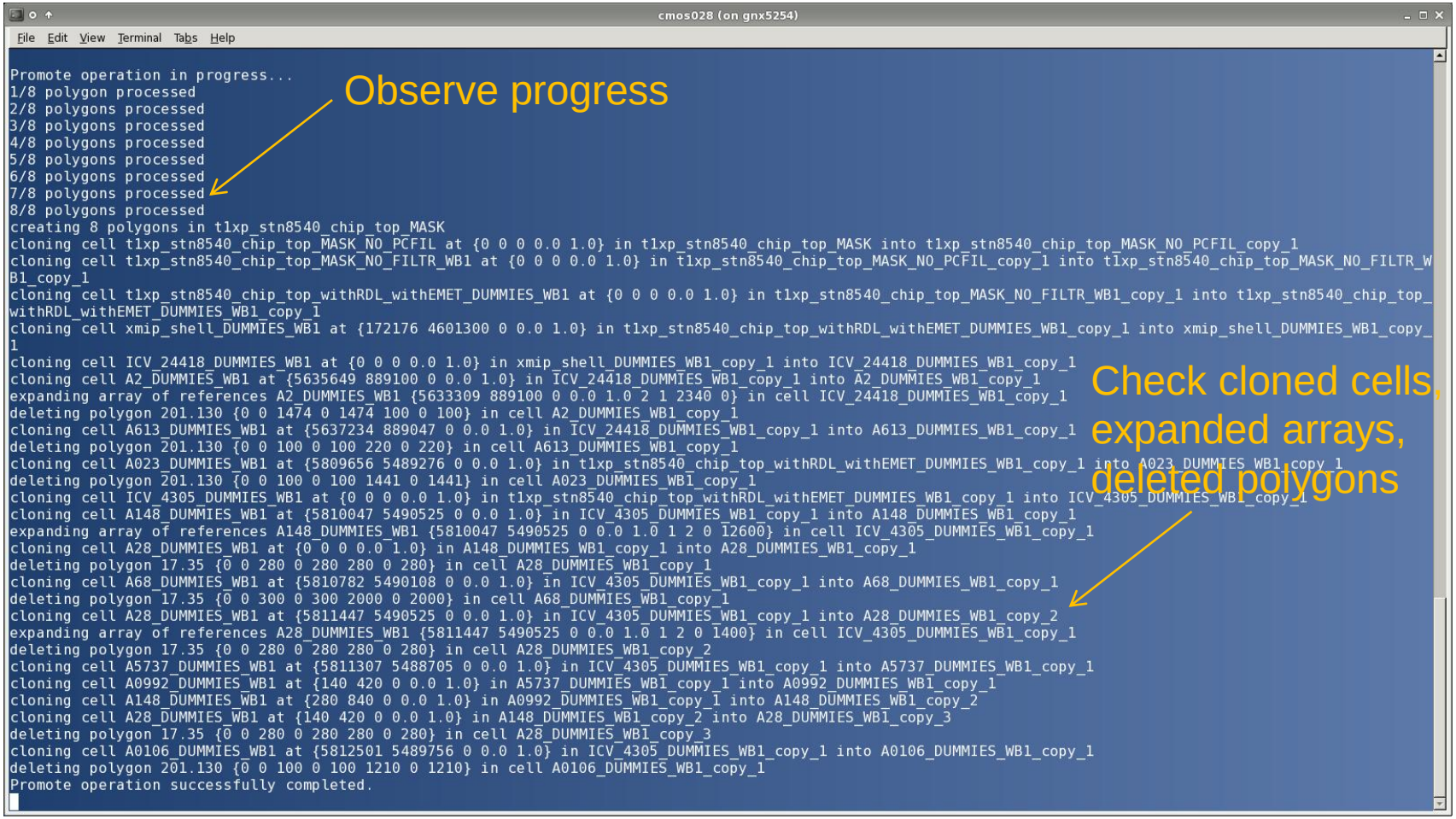
Saving updated layout





- Select the topcell then Right Mouse Button « Save As Layout… »
  - Give output filename with « .gds.gz » extension to zip at writing
- « From cell » must be ticked and filled with topcell name
  - See next slides for information on uninstanciated cells

DESIGNrev transcript




```
Promote operation in progress...
1/8 polygon processed
2/8 polygons processed
3/8 polygons processed
4/8 polygons processed
5/8 polygons processed
6/8 polygons processed
7/8 polygons processed
8/8 polygons processed
creating 8 polygons in t1xp_stn8540_chip_top_MASK
cloning cell t1xp_stn8540_chip_top_MASK_NO_PCFIL at {0 0 0 0.0 1.0} in t1xp_stn8540_chip_top_MASK into t1xp_stn8540_chip_top_MASK_NO_PCFIL_copy_1
cloning cell t1xp_stn8540_chip_top_MASK_NO_FILTR_WB1 at {0 0 0 0.0 1.0} in t1xp_stn8540_chip_top_MASK_NO_PCFIL_copy_1 into t1xp_stn8540_chip_top_MASK_NO_FILTR_W
B1_copy_1
cloning cell t1xp_stn8540_chip_top_withRDL_withEMET_DUMMIES_WB1 at {0 0 0 0.0 1.0} in t1xp_stn8540_chip_top_MASK_NO_FILTR_WB1_copy_1 into t1xp_stn8540_chip_top_
withRDL_withEMET_DUMMIES_WB1_copy_1
cloning cell xmip_shell_DUMMIES_WB1 at {172176 4601300 0 0.0 1.0} in t1xp_stn8540_chip_top_withRDL_withEMET_DUMMIES_WB1_copy_1 into xmip_shell_DUMMIES_WB1_copy_
1
cloning cell ICV_24418_DUMMIES_WB1 at {0 0 0 0.0 1.0} in xmip_shell_DUMMIES_WB1_copy_1 into ICV_24418_DUMMIES_WB1_copy_1
cloning cell A2_DUMMIES_WB1 at {5635649 889100 0 0.0 1.0} in ICV_24418_DUMMIES_WB1_copy_1 into A2_DUMMIES_WB1_copy_1
expanding array of references A2_DUMMIES_WB1 {5633309 889100 0 0.0 1.0 2 1 2340 0} in cell ICV_24418_DUMMIES_WB1_copy_1
deleting polygon 201.130 {0 0 1474 0 1474 100 0 100} in cell A2_DUMMIES_WB1_copy_1
cloning cell A613_DUMMIES_WB1 at {5637234 889047 0 0.0 1.0} in ICV_24418_DUMMIES_WB1_copy_1 into A613_DUMMIES_WB1_copy_1
deleting polygon 201.130 {0 0 100 0 100 220 0 220} in cell A613_DUMMIES_WB1_copy_1
cloning cell A023_DUMMIES_WB1 at {5809656 5489276 0 0.0 1.0} in t1xp_stn8540_chip_top_withRDL_withEMET_DUMMIES_WB1_copy_1 into A023_DUMMIES_WB1_copy_1
deleting polygon 201.130 {0 0 100 0 100 1441 0 1441} in cell A023_DUMMIES_WB1_copy_1
cloning cell ICV_4305_DUMMIES_WB1 at {0 0 0 0.0 1.0} in t1xp_stn8540_chip_top_withRDL_withEMET_DUMMIES_WB1_copy_1 into ICV_4305_DUMMIES_WB1_copy_1
cloning cell A148_DUMMIES_WB1 at {5810047 5490525 0 0.0 1.0} in ICV_4305_DUMMIES_WB1_copy_1 into A148_DUMMIES_WB1_copy_1
expanding array of references A148_DUMMIES_WB1 {5810047 5490525 0 0.0 1.0 1 2 0 12600} in cell ICV_4305_DUMMIES_WB1_copy_1
cloning cell A28_DUMMIES_WB1 at {0 0 0 0.0 1.0} in A148_DUMMIES_WB1_copy_1 into A28_DUMMIES_WB1_copy_1
deleting polygon 17.35 {0 0 280 0 280 280 0 280} in cell A28_DUMMIES_WB1_copy_1
cloning cell A68_DUMMIES_WB1 at {5810782 5490108 0 0.0 1.0} in ICV_4305_DUMMIES_WB1_copy_1 into A68_DUMMIES_WB1_copy_1
deleting polygon 17.35 {0 0 300 0 300 2000 0 2000} in cell A68_DUMMIES_WB1_copy_1
cloning cell A28_DUMMIES_WB1 at {5811447 5490525 0 0.0 1.0} in ICV_4305_DUMMIES_WB1_copy_1 into A28_DUMMIES_WB1_copy_2
expanding array of references A28_DUMMIES_WB1 {5811447 5490525 0 0.0 1.0 1 2 0 1400} in cell ICV_4305_DUMMIES_WB1_copy_1
deleting polygon 17.35 {0 0 280 0 280 280 0 280} in cell A28_DUMMIES_WB1_copy_2
cloning cell A5737_DUMMIES_WB1 at {5811307 5488705 0 0.0 1.0} in ICV_4305_DUMMIES_WB1_copy_1 into A5737_DUMMIES_WB1_copy_1
cloning cell A0992_DUMMIES_WB1 at {140 420 0 0.0 1.0} in A5737_DUMMIES_WB1_copy_1 into A0992_DUMMIES_WB1_copy_1
cloning cell A148_DUMMIES_WB1 at {280 840 0 0.0 1.0} in A0992_DUMMIES_WB1_copy_1 into A148_DUMMIES_WB1_copy_2
cloning cell A28_DUMMIES_WB1 at {140 420 0 0.0 1.0} in A148_DUMMIES_WB1_copy_2 into A28_DUMMIES_WB1_copy_3
deleting polygon 17.35 {0 0 280 0 280 280 0 280} in cell A28_DUMMIES_WB1_copy_3
cloning cell A0106_DUMMIES_WB1 at {5812501 5489756 0 0.0 1.0} in ICV_4305_DUMMIES_WB1_copy_1 into A0106_DUMMIES_WB1_copy_1
deleting polygon 201.130 {0 0 100 0 100 1210 0 1210} in cell A0106_DUMMIES_WB1_copy_1
Promote operation successfully completed.
```

- Launch « bsub –I » to keep the DESIGNrev shell

PROMOTE macro: description

- Support
  - PROMOTE macro supports polygons only
  - Paths, texts, edges, vertices and references are not supported
- Requirements
  - Usual DESIGNrev memory requirement, i.e. minimum 1.5x unzipped GDS file size
- Usage
  - PROMOTE smartly processes the hierarchy to minimize layout size increase
  - Applying PROMOTE once on a set of polygons will lead to more optimized result than applying PROMOTE multiple times on each polygon
- Runtime
  - Depends on
    - Original layout size and complexity
    - Total flat count of each polygon to be promoted
  - Linearly increases with number of polygons to be promoted
  - Can take from seconds to minutes
  - Ex: In this example, processing 8 tiling shapes in this 48GB unzipped GDS design took <4mn

PROMOTE macro: warning!

- Ambiguities
  - Ambiguities happen when selected polygons are originally duplicated, i.e. identical polygons in same cell or same hierarchy, and at same location
  - Ambiguities are resolved arbitrarily and only one polygon in hierarchy gets deleted per polygon promoted
  - Ambiguities are explicitely reported. User may want to repeat PROMOTE on remaining polygons
- Un-Instantiated cells
  - When all instances of a cell need to be cloned (or typically when a cell has a unique instance, like main blocks), the cell won't be instanciated any more in the promoted layout
  - Original uninstanciated cells are not automatically deleted by PROMOTE, they are kept for information as separate cells at same level as the layout's topcell
  - To eliminate those cells
    - Manually: select them in left pane, click Right Mouse Button, then Delete
    - Automatically: instead of « Save as », use «Export Layout » and set « From cell » to the topcell name (note: field is automatically populated with current cell if using RMB Save As Layout…)
- Empty cells
  - Recursive promotions may lead to empty cells
  - To eliminate those cells, instead of « Save as », use «Export Layout » and tick « Do not write empty cells or their references »

PROMOTE macro: Following steps

- XOR is mandatory after the modification
- Signoff LVS is mandatory after the modification
- Signoff DRC is mandatory after the modification

THANK YOU !

CMOS and derivative PDK

life.augmented