Design Closure Within ICC

With Updates For G-2012.06

Jonathan Dawes

SYNOPSYS® 25 Years

CONFIDENTIAL INFORMATION

The following material is confidential information of Synopsys and is being disclosed to you pursuant to a non-disclosure agreement between you or your employer and Synopsys. The material being disclosed may only be used as permitted under such non-disclosure agreement.

IMPORTANT NOTICE

In the event information in this presentation reflects Synopsys' future plans, such plans are as of the date of this presentation and are subject to change. Synopsys is not obligated to develop the software with the features and functionality discussed in these materials. In any event, Synopsys' products may be offered and purchased only pursuant to an authorized quote and purchase order or a mutually agreed upon written contract.

Agenda

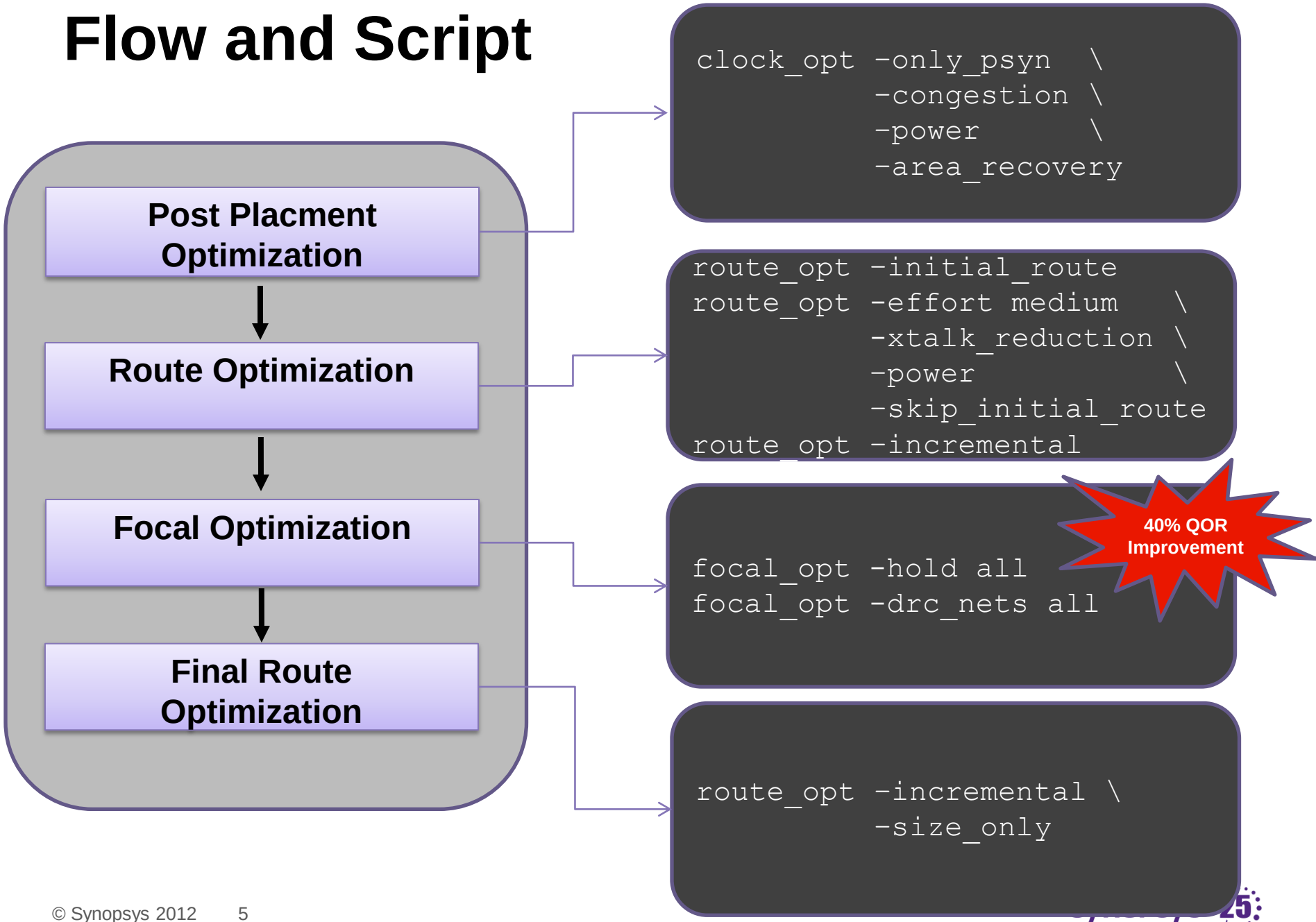
- Main ICC Closure flow
- Timing Closure
- Physical Closure

ICC Closure Flow

- Standard flow you may be familiar with
  - Route_opt
  - Focal_opt
  - Signoff_opt

  - PrimeTime ECO flow

Flow and Script



Timing Closure

- Signoff_opt
- PrimeTime Cheetah ECO
- Implementing ECOs in ICC

Timing Closure

- Signoff_opt
- PrimeTime Cheetah ECO
- Implementing ECOs in ICC

De-emphasis of signoff_opt

- **`signoff_opt`** timing optimization becomes LCA and requires a LCA license
- **`run_signoff`** and **`check_signoff_correlation`** remain GA features
- PrimeTime ECO is recommended for ECO

Feature	F-2011.09	G-2012.06
`signoff_opt` (timing and DRC)	GA	LCA

Distributed Processing Enhancement

- The **`add_distributed_hosts`** command is retired in G-2012.06 and replaced by the **`set_host_options command`**

- The **`remove_host_options`** and **`report_host_options`** commands are enhanced to support host setup for PrimeTime and StarRC

Distributed Processing Enhancement

- UI

```
set_host_options
    -target ICC | PrimeTime | StarRC

remove_host_options
    -target ICC | PrimeTime | StarRC

report_host_options
    -target ICC | PrimeTime | StarRC
```

Distributed Processing Enhancement

- Flow recommendations
  - The **`set_host_options`** command is compatible with older releases of PrimeTime
    - When you use older releases of PrimeTime, IC Compiler converts the **`set_host_options`** settings to **`add_distributed_host`** settings
  - The **`-max_cores`** option is supported only with **`-target PrimeTime | ICC`**
  - For StarRC, you need to set the number of cores by using **`set_starrcxt_options -num_parts`**
    - If you set **`-max_cores`** with **`-target StarRC`**, **`set_host_options`** errors out

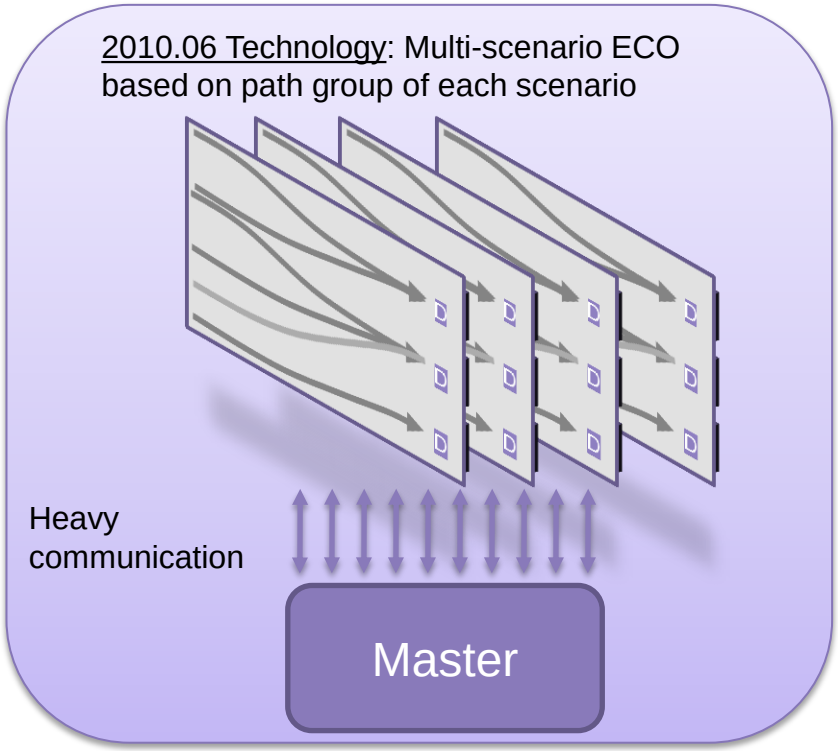
Timing Closure

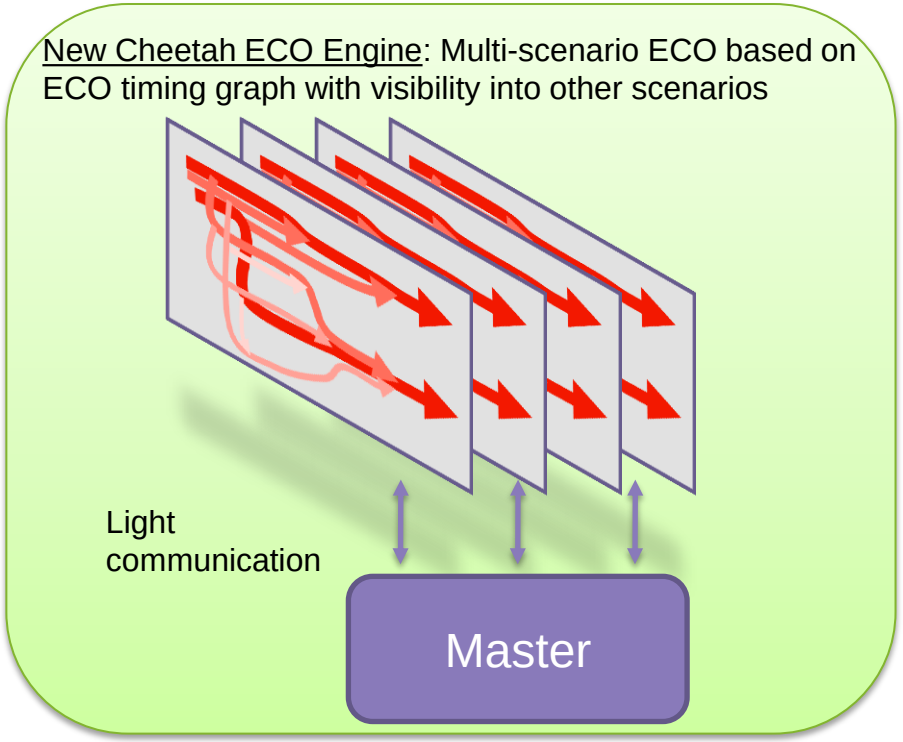
- Signoff_opt
- PrimeTime Cheetah ECO
- Implementing ECOs in ICC

Next Generation ECO Technology

Code name: Cheetah

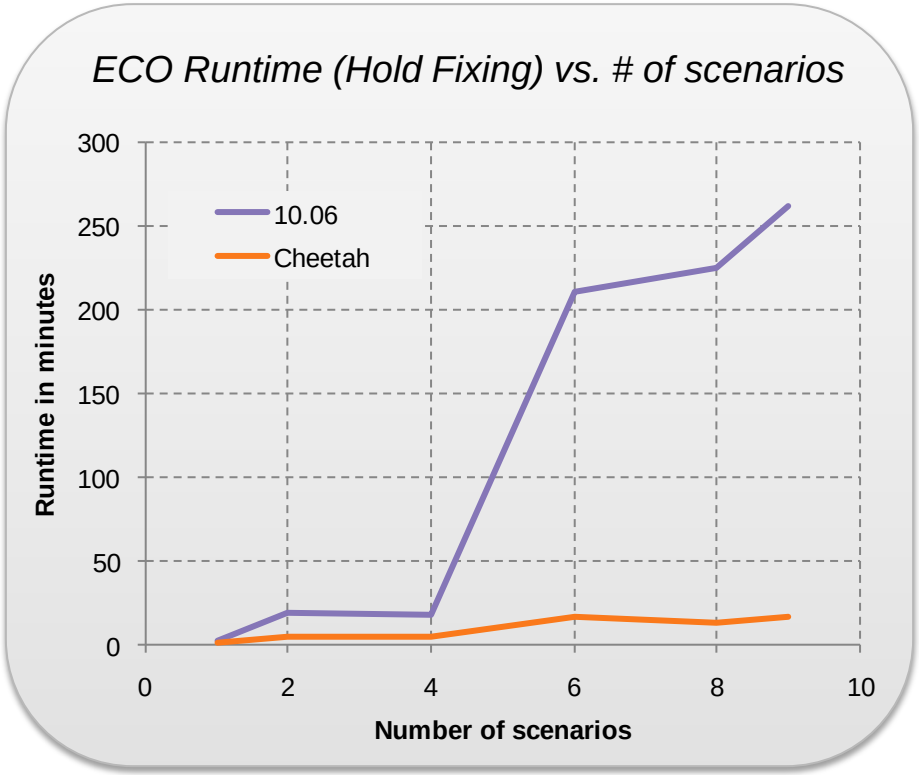
- New infrastructure and algorithms
- Maximizes parallelism for data processing
- Utilizes global timing view for optimizations
- General Availability in PrimeTime SI since 2011.06

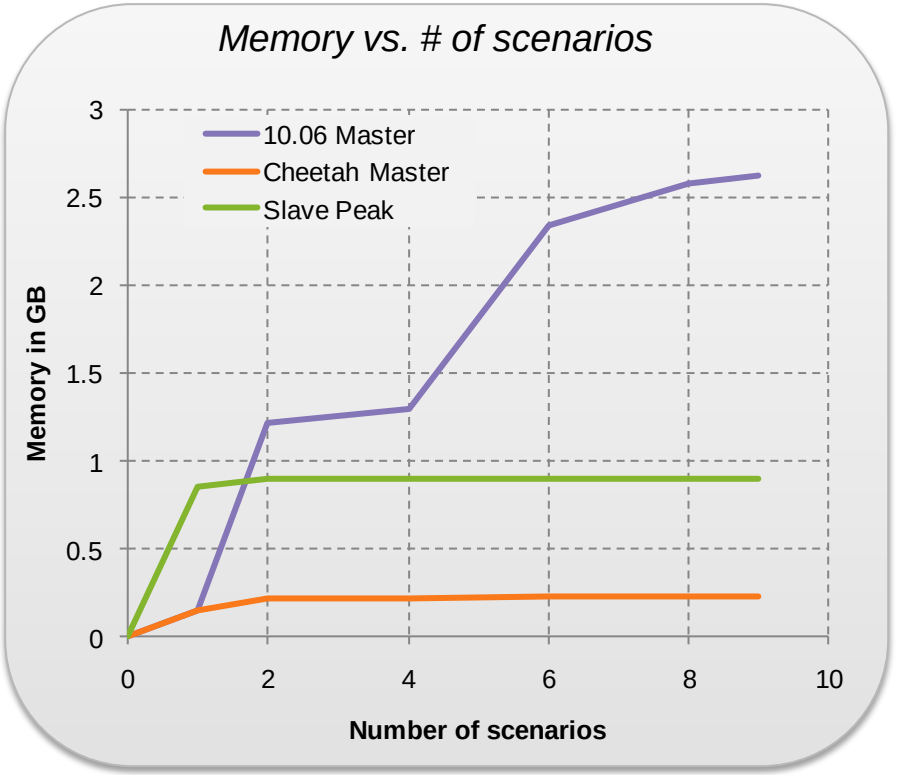




‘Cheetah’ ECO Technology

Runtime and Memory Scalability With Increasing Scenarios





With Cheetah, ECO runtime/memory
does not increase significantly with growing number of scenarios

Source: Customer 340K instance design, 9 Scenarios, same # of cores used as scenarios

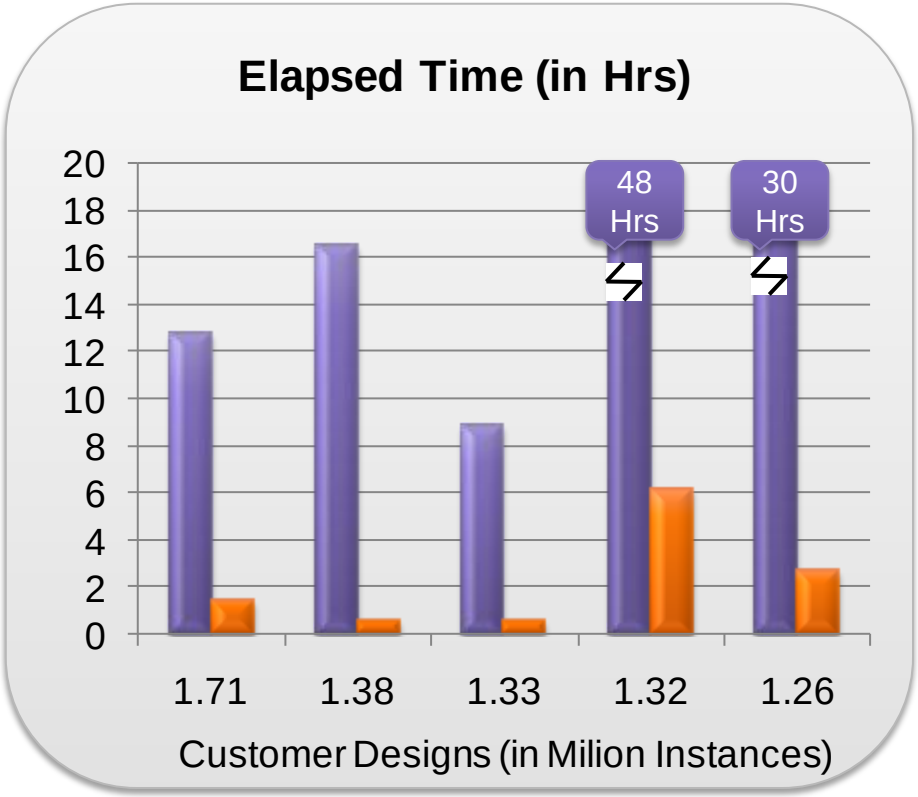
‘Cheetah’ ECO Technology

Additional controls for fixing

- Cheetah ECO fixing supports library don’t use attributes
- Cheetah Supports PBA based fixing for both setup and hold
  - Benefits regular PBA and advanced OCV PBA flows
- Cheetah supports distributed multicore (DSTA) with both setup and hold fixing
- Footprint swapping supported during sizing with hold fixing in Cheetah
  - Enables Cheetah to perform setup and hold fixing with vt swaps

Cheetah ECO – Setup Fixing Runtime

PrimeTime ECO



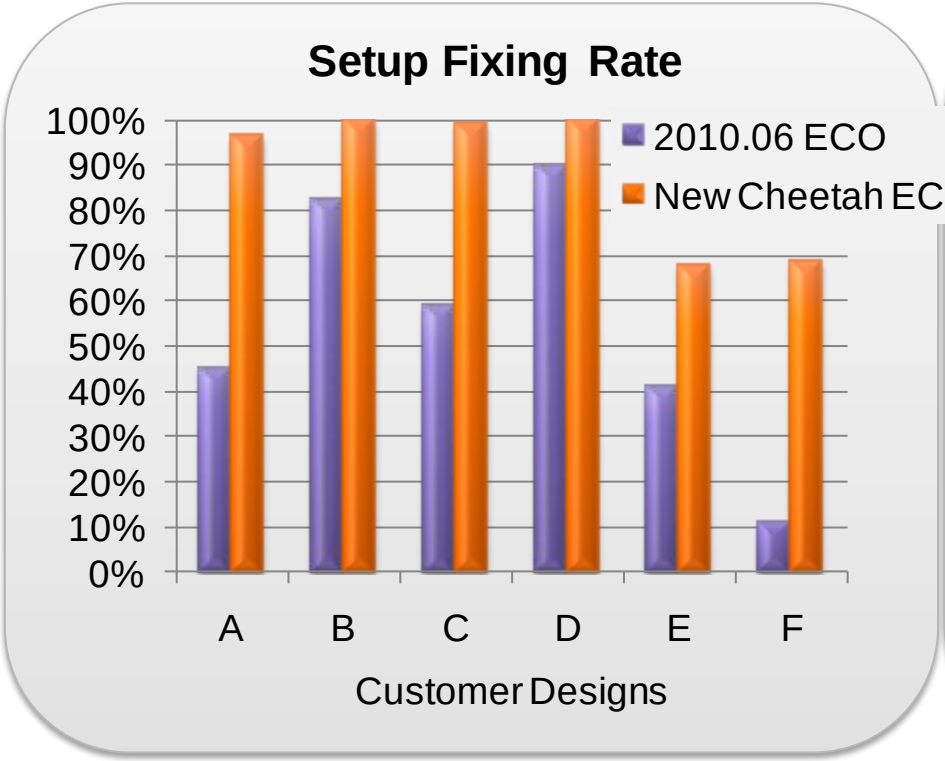
Full ECO Loop

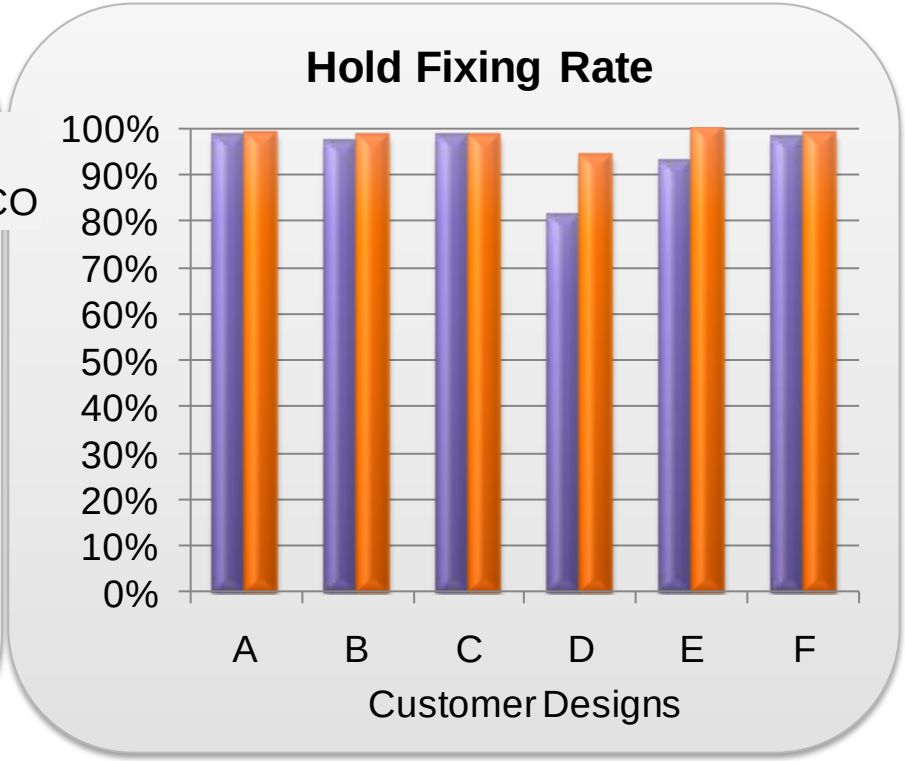
PrimeTime Setup ECO flow is **7-10X faster**

Full ECO flow 'IC Compiler + StarRC + PrimeTime' is **~1.5X faster**

Cheetah ECO – Better Fixing Rates

Effectively Fixes 1000's of Violations





Setup fixing rates improved to ~**70%**

Hold fixing rates remain ~**95%**

Timing Closure

- Signoff_opt
- PrimeTime Cheetah ECO
- Implementing ECOs in ICC

Implementing ECOs in ICC

- eco_netlist command
- Eco Placement
- add_buffer_on_route

Implementing ECOs in ICC

- eco_netlist command
- Eco Placement
- add_buffer_on_route

IC Compiler->PrimeTime Flow Support

- In the ECO flow, you often need to get a collection of ECO cells for analysis and ECO place and route
  - Important for PrimeTime ECO scripts

- `eco_netlist -by_tcl_file`
  is already enhanced to read a PrimeTime ECO script with faster runtime than sourcing

Cells Identified by `eco_netlist -by_tcl_file`

- Four cell-related operations
  - **`create_cell`**
    - Need to perform ECO placement and legalize
  - **`insert_buffer`**
    - If buffer is for fixing hold time
      - Keep original location (close to load pin), only legalize
    - If buffer is for fixing setup time
      - Need to perform ECO placement and legalize
  - **`change_link`**
    - Keep original location, only legalize
  - **`size_cell`**
    - Keep original location, only legalize

ECO Cell Collection After `eco_netlist`

- Collections:
  - `create_cell`: `eco_netlist_created_cells`
  - `insert_buffer`: `eco_netlist_inserted_buffers`
  - `change_link`: `eco_netlist_changed_link`
  - `size_cell`: `eco_netlist_sized_cells`
- Enable:
  - `set_app_var eco_record_cell_change true`
  - Default is false
  - Each run of `eco_netlist -by_tcl_file` overwrites these four collections with new cells
- Disable
  - `set_app_var eco_record_cell_change false`

Implementing ECOs in ICC

- eco_netlist command
- Eco Placement
- add_buffer_on_route

ECO Cell Placement Enhancement

Overview and Benefits

- Fast convergence is required during the ECO placement phase
- `legalize_placement -incremental -eco` was designed around cell legalization technology
  - No consideration of many aspects of ECO placement
- New command: `place_eco_cells`
  - Fast runtime, no need to link the design
  - Connectivity-based ECO cell location
  - Legalize only on free sites
    - Does not move non-ECO cells
  - Allows you to check or adjust ECO cell locations before legalizing the ECO cells

ECO Cell Placement Enhancement
UI

`place_eco_cells`	
`-cells` *`cell_list`*	(cells to place)
`-unplaced_cells`	(place all unplaced cells)
`[-no_legalize]`	(do not legalize)
`[-legalize_only]`	(only perform legalization on ECO cells that already have locations)

`-cells` and `-unplaced_cells` are mutually exclusive

ECO Cell Placement Enhancement
Usage

- Placement for a small number of ECO cells
  - Option not to legalize if `place_eco_cells` needs to be run many times
  - Only legalize the cells once
- Place ECO buffers based on connectivity improves:
  - Postroute timing (DRC) fixing with buffers
  - Timing degradation issues after reading in a new ECO netlist
  - Buffer placement from PrimeTime DRC fixing ECO script

Enhancements in 2012.06

- Channel-aware ECO placement to utilize thin channel areas
- Handles ECO placement for fully filled designs
  - Supports 1x spacing rules
- New options to ignore highly connected nets

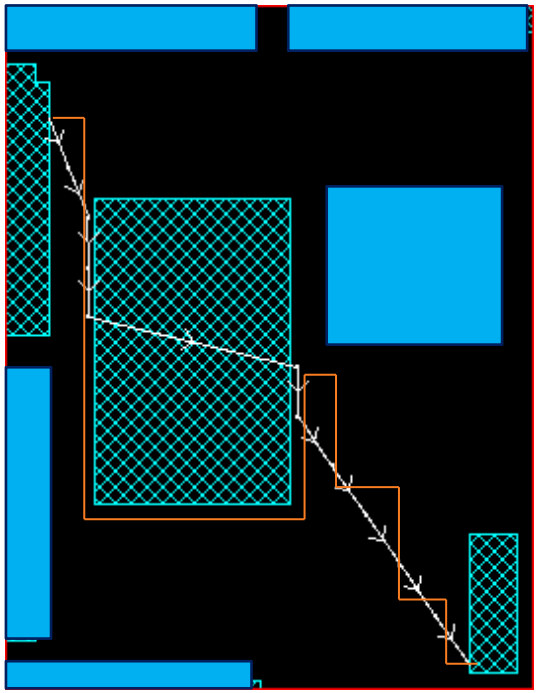
UI

```
place_eco_cells
      -channel_aware
      -remove_filler_references filler_list
      -max_fanout max_fanout
      -ignore_pin_connection pin_name_list
```

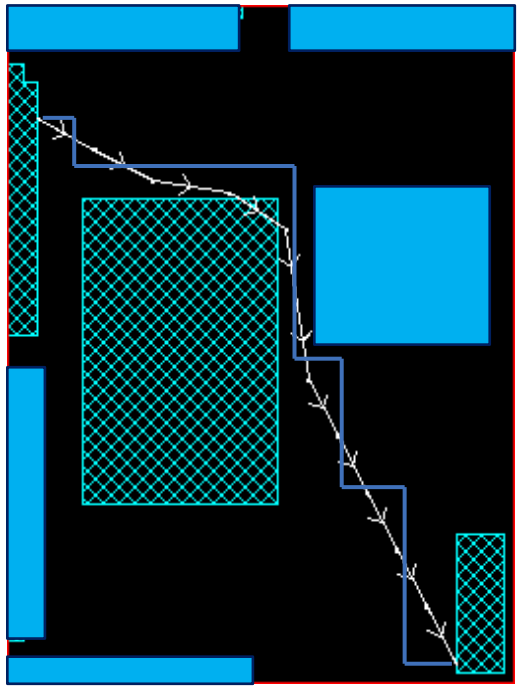
UI

-channel_aware

- Enables ECO placement to utilize thin channel areas with connectivity-aware consideration



Non channel aware



Channel aware

UI

`-remove_filler_references` *`filler_list`*

- Specified filler reference cells are ignored during placement. Any cells overlapping with ECO cells after placement are removed
- User may need to incrementally re-fill
- All unspecified filler reference cells are treated as standard cells during placement
- If this option is not specified, all filler cells are treated as standard cells during placement

- 1x filler spacing rule support when you set the hidden `epl_honor_spacing_rule` variable to `true`
  - **`set epl_honor_spacing_rule true`**

UI

`-max_fanout` *`max_fanout`*

- Ignores a net connection if the number of fanouts exceeds the specified limit
- There is no impact on legalization - the command fails if used with **`-legalize_only`**
- The default fanout value is 100

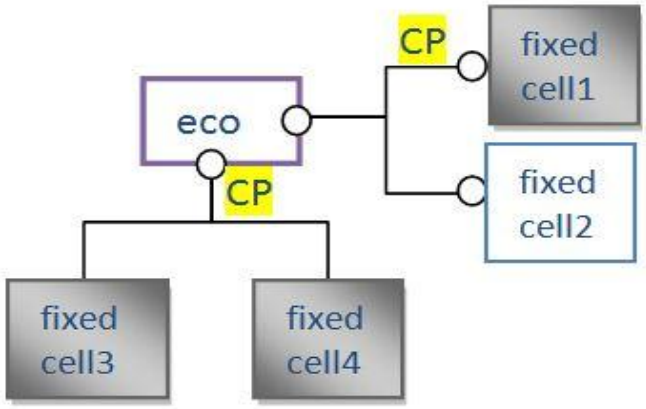
UI

-ignore_pin_connection *pin_name_list*

Use this option to ignore the connections of the specified pins. The pin name must be the reference cell pin name.

```
-ignore_pin_connection {CP}
```

Implies that cells 1, 3, and 4 are not considered when evaluating ECO cell placement



ECO Cell Placement Enhancement
Summary

- Fast runtime, no need to link the design
- Connectivity-based ECO cell location
- Can work on fully filled design
- Better design convergence
- Plan to phase out

  `legalize_placement -eco`

  in G-2012.06 release and subsequent reference methodology release

Implementing ECOs in ICC

- eco_netlist command
- Eco Placement
- add_buffer_on_route

Introduction

- A new command **`add_buffer_on_route`** has been created to provide better usability over the current **`insert_buffer -on_route`** flow

- Provides the following functionality:
  - Distance-based insertion on nets along their route
  - Place and legalize the new buffers
  - Split the existing routes, and route the new and modified nets
  - One command replaces multiple existing commands
  - Faster runtime with new route tracing functionality (no extraction)

Command Syntax

add_buffer_on_route

[-net_prefix <string>]	prefix for new nets
[-cell_prefix <string>]	prefix for new cells
[[-locations <list_of_locations>]	locations to place buffers
|| [-repeater_distance <length> [-first_distance <length>]]]	distance between new cells distance from driver to first buffer (only specified with repeater_distance)
[-max_distance_for_legalized_location <length>]	specifies the maximum distance the new buffers can move during legalization
[-inverter_pair]	library cell is inverter
[-no_legalize]	do not legalize new cells & do not route new nets
<buffer_lib_cell>	buffer library cell name
<nets>	list of routed/prerouted nets to be buffered

Command Usage

- Using the `-no_legalize` option
  - The buffers added are placed but not legalized
  - Existing route is split and associated with new nets
  - New nets are not routed
    - The new cells must be legalized and the new nets must be routed separately
  - Provides a batch mode operation with faster runtime:
    - Many nets can be buffered at the same time
      - No legalization bounds
    - All added buffers are legalized at the same time (`place_eco_cells -legalize_only`)
    - All new nets are routed at the same time (`route_zrt_eco`)

Flow Examples

- Interactive

```
add_buffer_on_route -location {100 200} net1 BUFX1
```

- Batch

```
add_buffer_on_route -no_legalize net1 BUFX1
add_buffer_on_route -no_legalize net2 BUFX1
add_buffer_on_route -no_legalize net3 BUFX1
place_eco_cells -legalize_only -cells $buffcells
route_zrt_eco
```

2012.06 Enhancements

- Scales the repeater distance based on the user-specified routing layer and route width scaling factors
- Ensures the addition of an even number of inverters from a driver to each load
- Trims buffers that are placed too close to load pins

UI

- **add_buffer_on_route**

  -scaled_by_layer {{*layer1 scale1*}{*layer2 scale2*}…}
  - Scales distance on each layer
  - Must be used with **-repeater_distance**

  -scaled_by_width
  - Scales width on each layer
  - Must be used with **-repeater_distance**

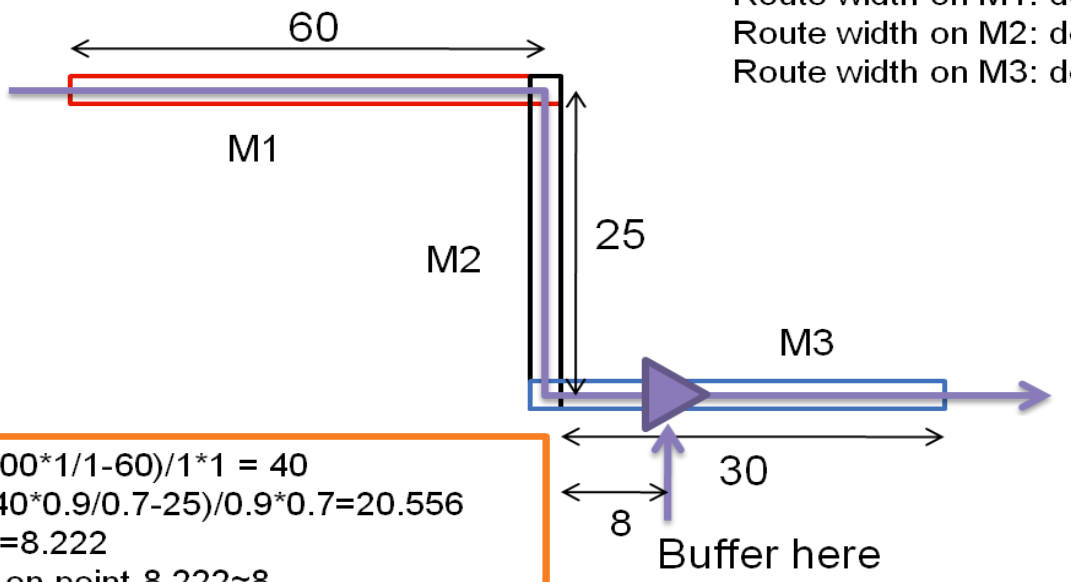
  -inverter_pair
  - Ensures the inverter numbers inserted are even from a driver to any load by adding extra inverter where needed
    - Previously would have failed with a `UIED-178` message

UI

Example

```
add_buffer_on_route  \
        -repeater_distance 100 \
        -scaled_by_layer { {M2 0.9} {M3 0.8} } \
        -scaled_by_width
```

Route width on M1: default width
Route width on M2: default width x0.7
Route width on M3: default width x 2



After M1: remain =(100*1/1-60)/1*1 = 40
After M2: remain = (40*0.9/0.7-25)/0.9*0.7=20.556
On M3: 20.556*0.8/2=8.222
So add buffer on M3 on point 8.222≈8

Add Buffer On Route

- Trims buffers that are too close to load pins
  - When the route distance to a load pin is less than the value specified with **`-repeater_distance`**, the buffer is not inserted
  - Controlled by the hidden **`add_buffer_on_route_min_distance_to_load`** variable

Benefits and Limitations

- Benefits
  - Provides greater control over repeater distance for on-route buffer insertion
  - Ensures correct inverter-pair insertion

- Limitations
  - Inverter-pair trimming is not supported

Physical Closure

- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

Physical Closure

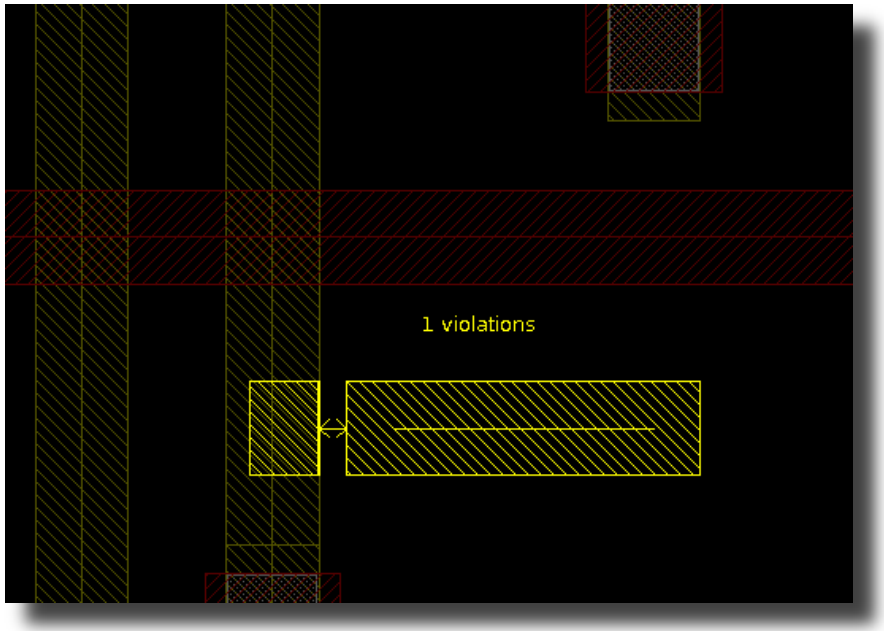
- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

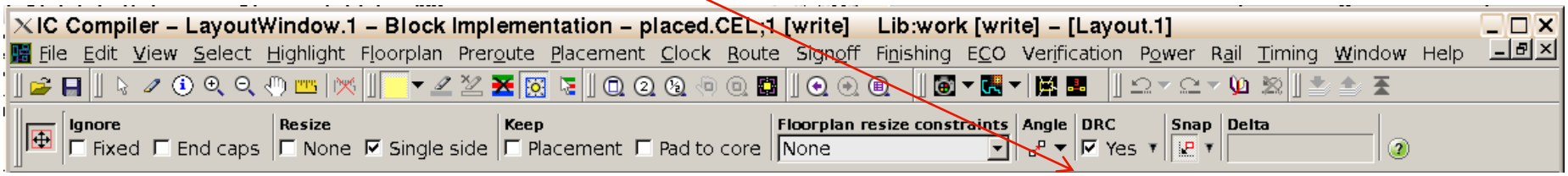
Interactive Editing – DRC Preview

- GUI tools support editing DRC preview
- User able to edit DRC aware
- Edits able to be done confidently

Command and UI
Create Shape

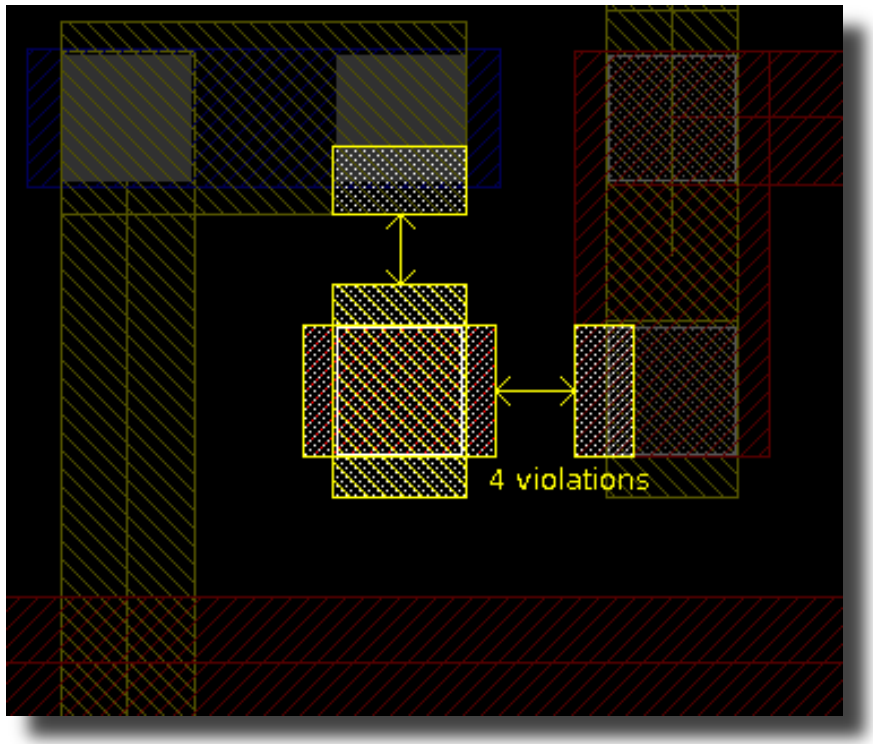
- Editing DRC violations will be previewed for new shapes created by this command
- Enables editor DRC checking





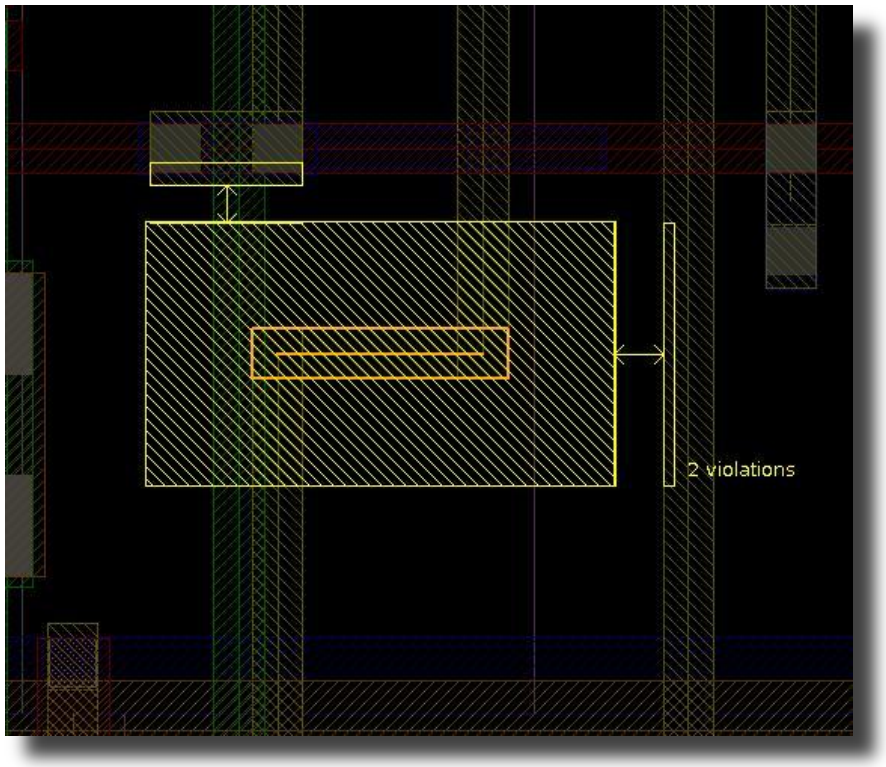
Command and UI
Create Via

- Editing DRC violations will be previewed for new vias created by this command
- Enables editor DRC checking



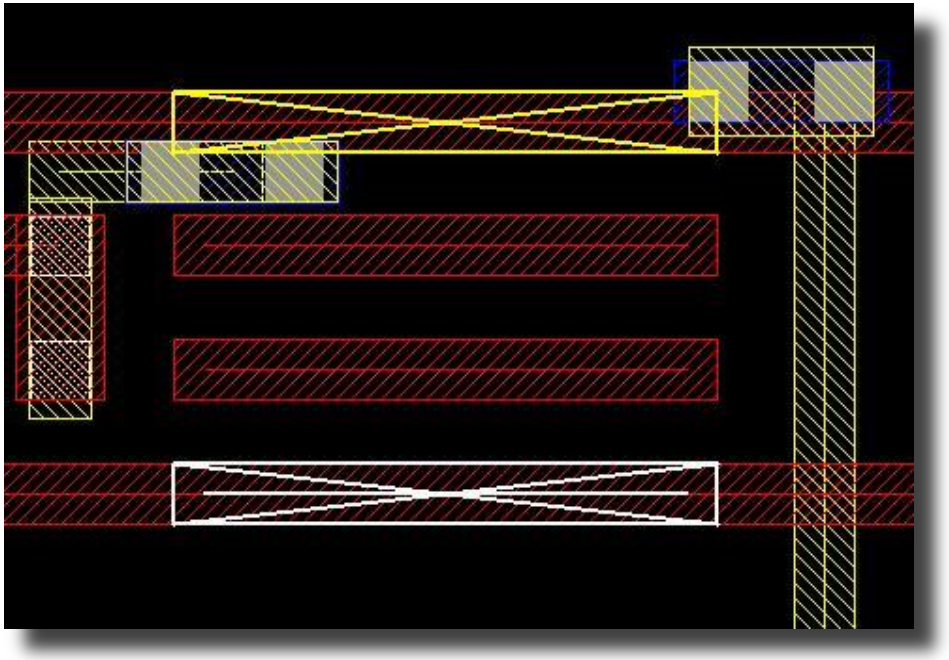
Command and UI
Move/Resize Tool

- Editing DRC violations will be previewed for Editing DRC related objects moved or resized using this command
- Enables editor DRC checking



Command and UI
Copy Tool

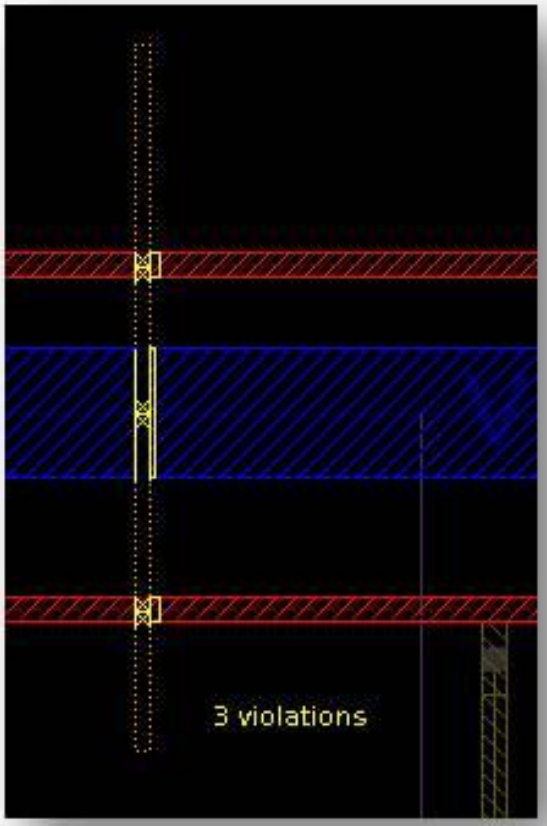
- Editing DRC violations will be previewed for new Editing DRC related objects created by this command
- Enables editor DRC checking



Command and UI
Split Tool

- Editing DRC violations will be previewed for Editing DRC related objects modified by this command

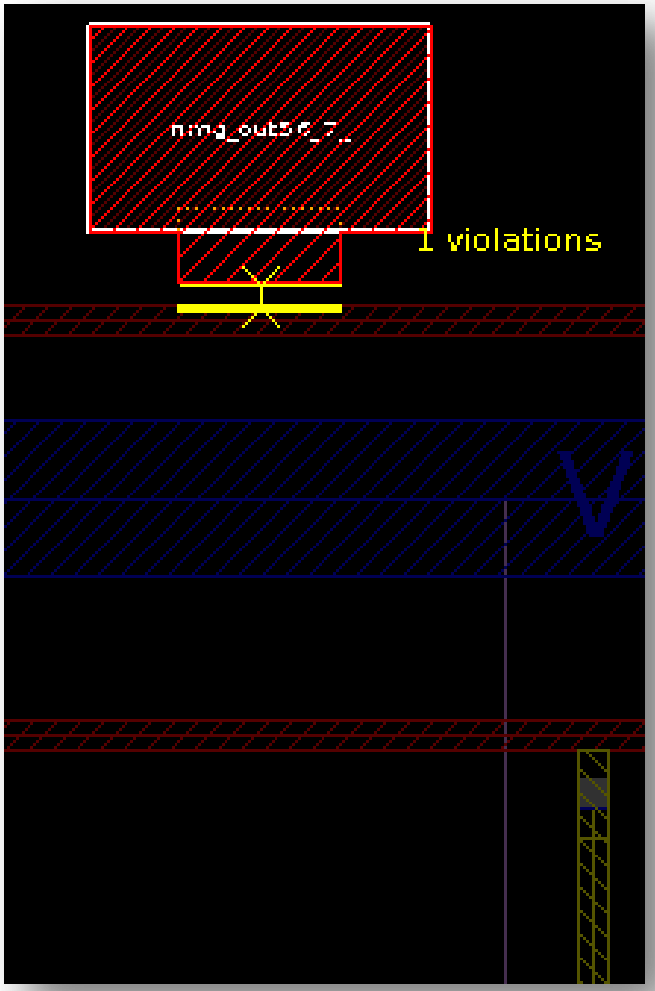
- Enables editor DRC checking



Command and UI
Reshape Tool

- Editing DRC violations will be previewed for Editing DRC related objects modified by this command

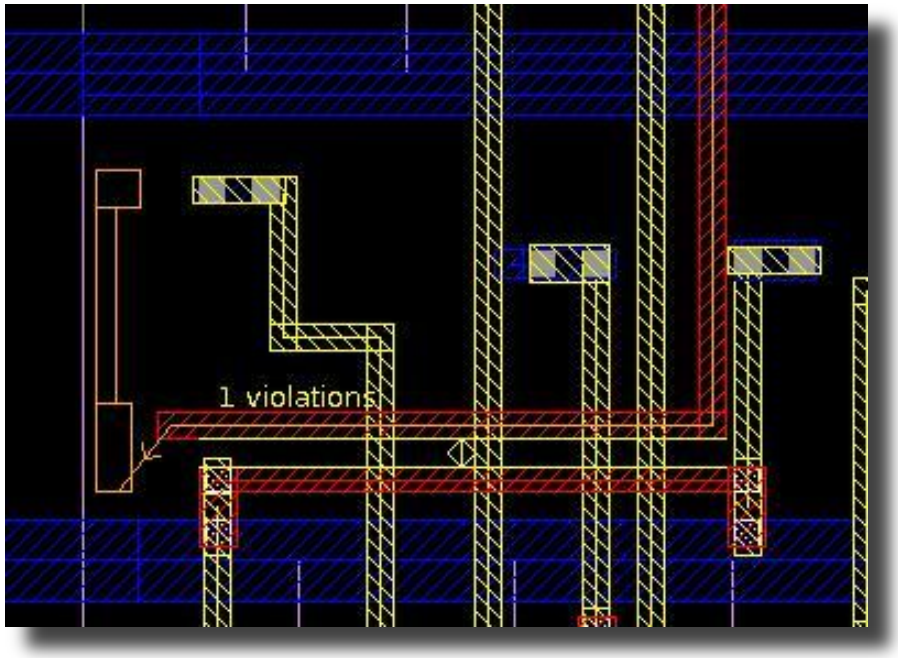
- Enables editor DRC checking



Command and UI
Create Route Tool

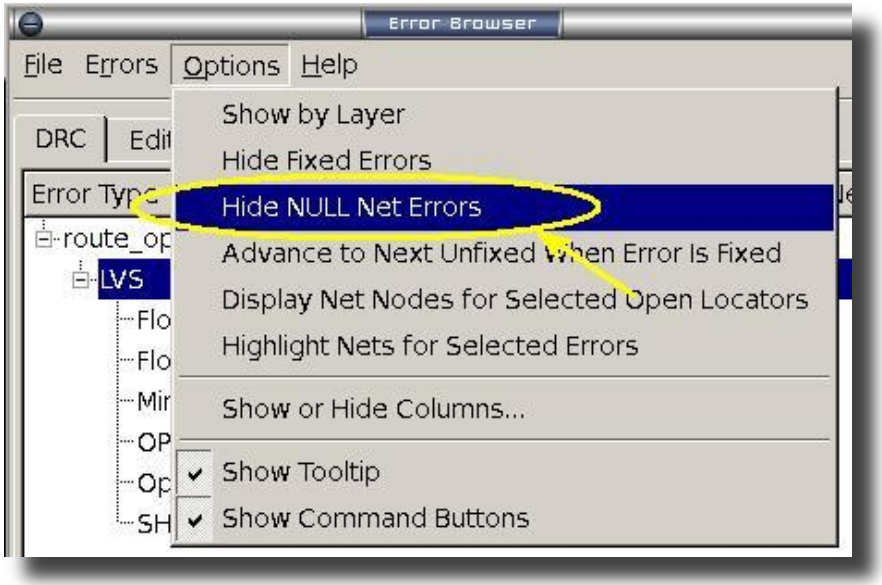
- Editing DRC violations will be previewed for wires and vias created by the Create Route Tool

- Enables editor DRC checking



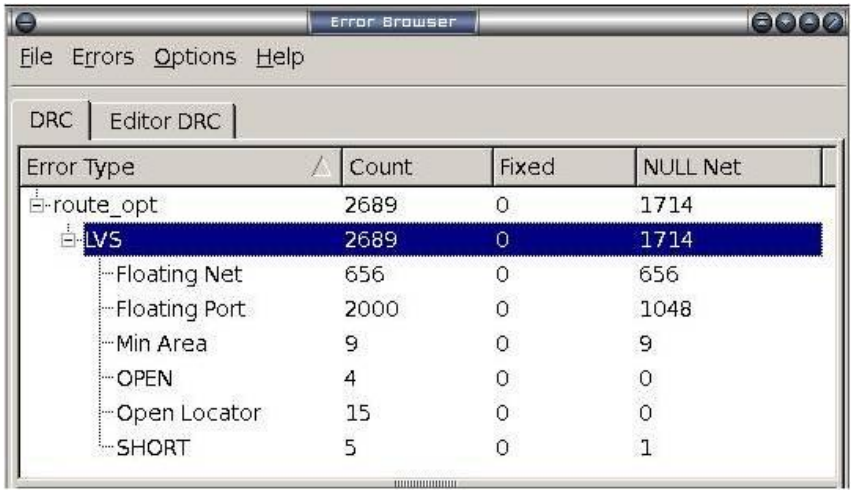
Command and UI
Error Browser GUI

- Error browser has option to hide all errors associated with NULL nets
- Toggle with Options->Hide NULL Net Errors command
- The setting will be saved in the user preferences
- Use of this option is appropriate for hiding errors associated with NULL nets for an extended period or for an entire use session



Command and UI
Error Browser GUI (2)

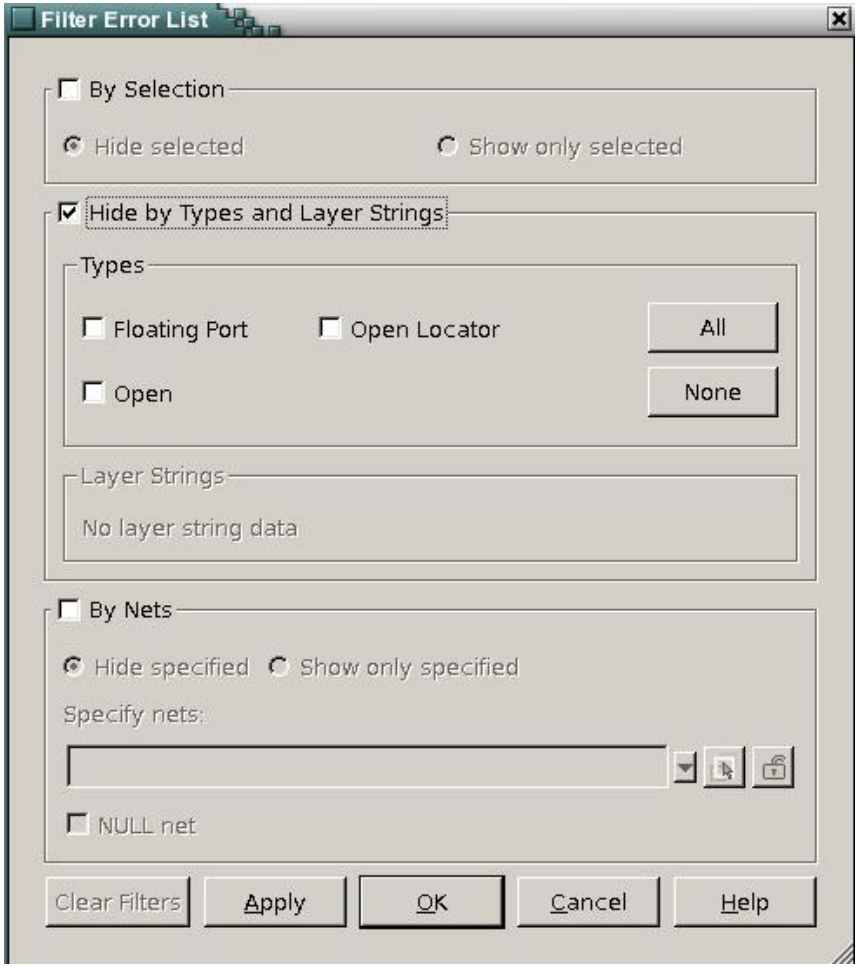
- Summary provides a column for errors against NULL nets



Command and UI
Error Browser GUI (3)

- Filtering feature is enhanced so that violations can be filtered by error types or layers regardless of how errors have been grouped

- In addition filtering by associated nets is added. Errors may be filtered:
  - By selected errors
  - By error types and layers
  - By associated nets



Command and UI
Tcl Commands

- The command `gui_filter_errors` is enhanced with the following new options:

  `-types`

  `-error_layer_strings`

  `-nets`

  `-no_apply`

- In addition filtering by associated nets is added. Errors may be filtered:
  - By selected errors
  - By error types and layers
  - By associated nets

Command and UI

Tcl Commands(2)

- The new command `gui_apply_error_filters` applies all outstanding filters that have been set with gui_filter_errors -no_apply

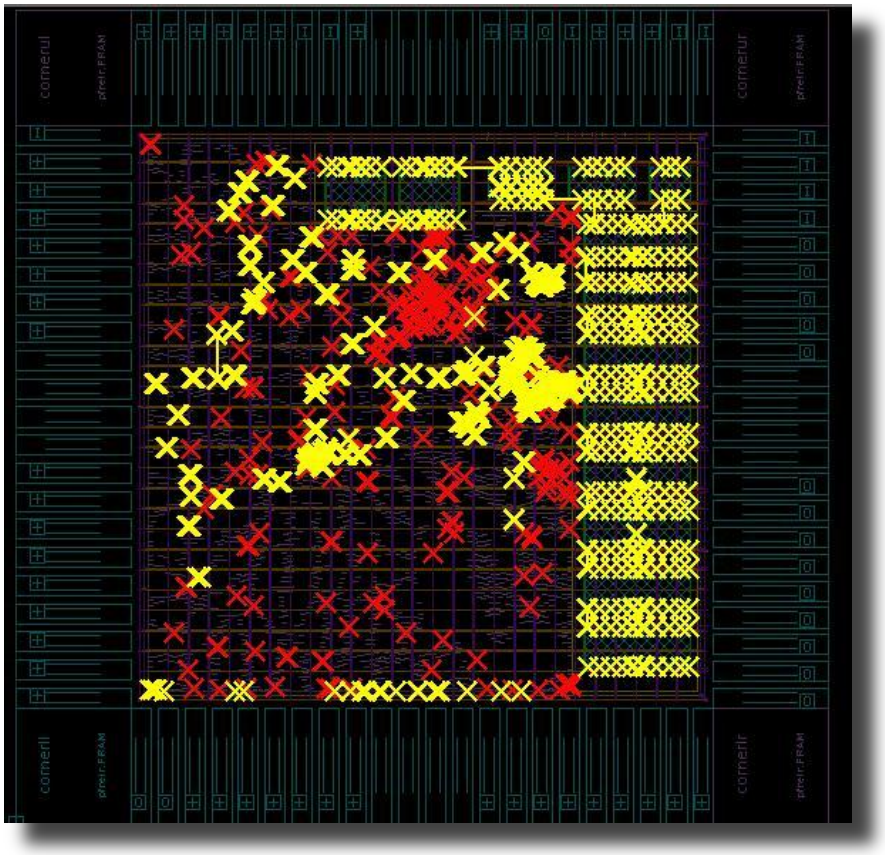
- status `gui_filter_errors`

```
[-hide | -show]
[-types <names>]
[-error_layer_strings <names>]
[-nets <nets>]
[-no_apply]
[-groups <names>]
```

Command and UI

Error Highlight in the Layout View

- Error groups can be assigned highlight colours
- Highlight colours are also displayed in the Error Browser error list
- Highlight colours are not saved in the database

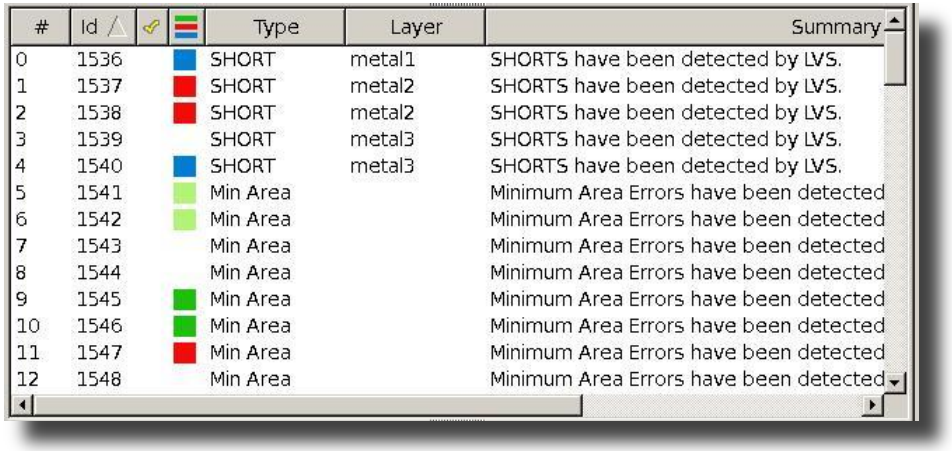


Layout view with errors drawn in multiple colors

Command and UI

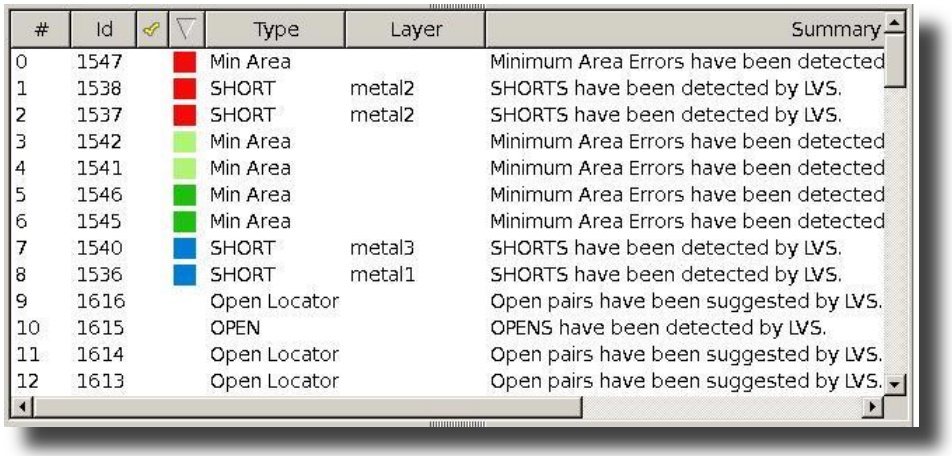
Colour Column in the Error List

- A new column in the error list displays the highlight color icons for highlighted errors



#	Id	Type	Layer	Summary
0	1536	SHORT	metal1	SHORTS have been detected by LVS.
1	1537	SHORT	metal2	SHORTS have been detected by LVS.
2	1538	SHORT	metal2	SHORTS have been detected by LVS.
3	1539	SHORT	metal3	SHORTS have been detected by LVS.
4	1540	SHORT	metal3	SHORTS have been detected by LVS.
5	1541	Min Area		Minimum Area Errors have been detected
6	1542	Min Area		Minimum Area Errors have been detected
7	1543	Min Area		Minimum Area Errors have been detected
8	1544	Min Area		Minimum Area Errors have been detected
9	1545	Min Area		Minimum Area Errors have been detected
10	1546	Min Area		Minimum Area Errors have been detected
11	1547	Min Area		Minimum Area Errors have been detected
12	1548	Min Area		Minimum Area Errors have been detected

The color column displays highlight colors

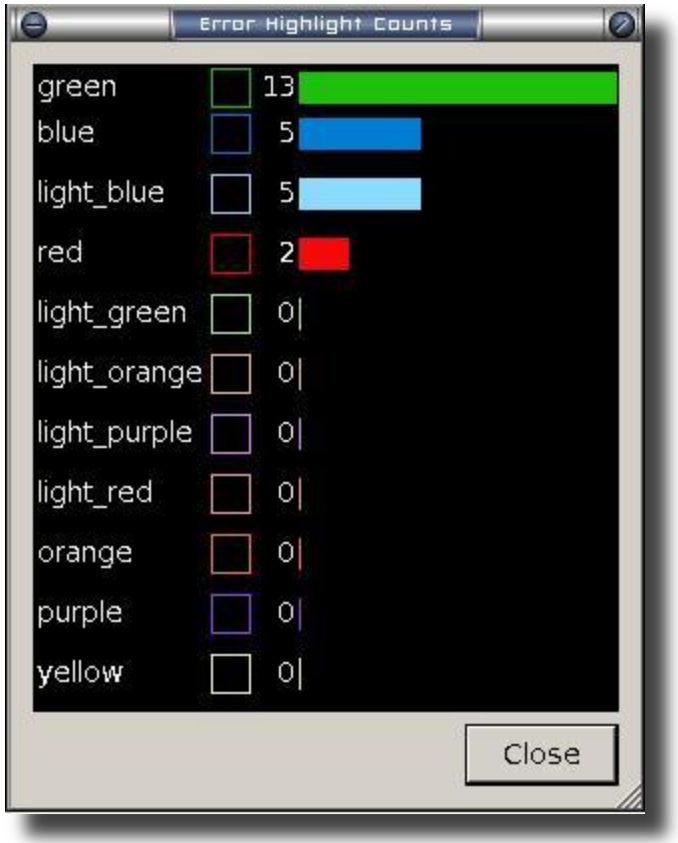


#	Id	Type	Layer	Summary
0	1547	Min Area		Minimum Area Errors have been detected
1	1538	SHORT	metal2	SHORTS have been detected by LVS.
2	1537	SHORT	metal2	SHORTS have been detected by LVS.
3	1542	Min Area		Minimum Area Errors have been detected
4	1541	Min Area		Minimum Area Errors have been detected
5	1546	Min Area		Minimum Area Errors have been detected
6	1545	Min Area		Minimum Area Errors have been detected
7	1540	SHORT	metal3	SHORTS have been detected by LVS.
8	1536	SHORT	metal1	SHORTS have been detected by LVS.
9	1616	Open Locator		Open pairs have been suggested by LVS.
10	1615	OPEN		OPENS have been detected by LVS.
11	1614	Open Locator		Open pairs have been suggested by LVS.
12	1613	Open Locator		Open pairs have been suggested by LVS.

Error list sorted by color

Command and UI
Error Highlight Counts

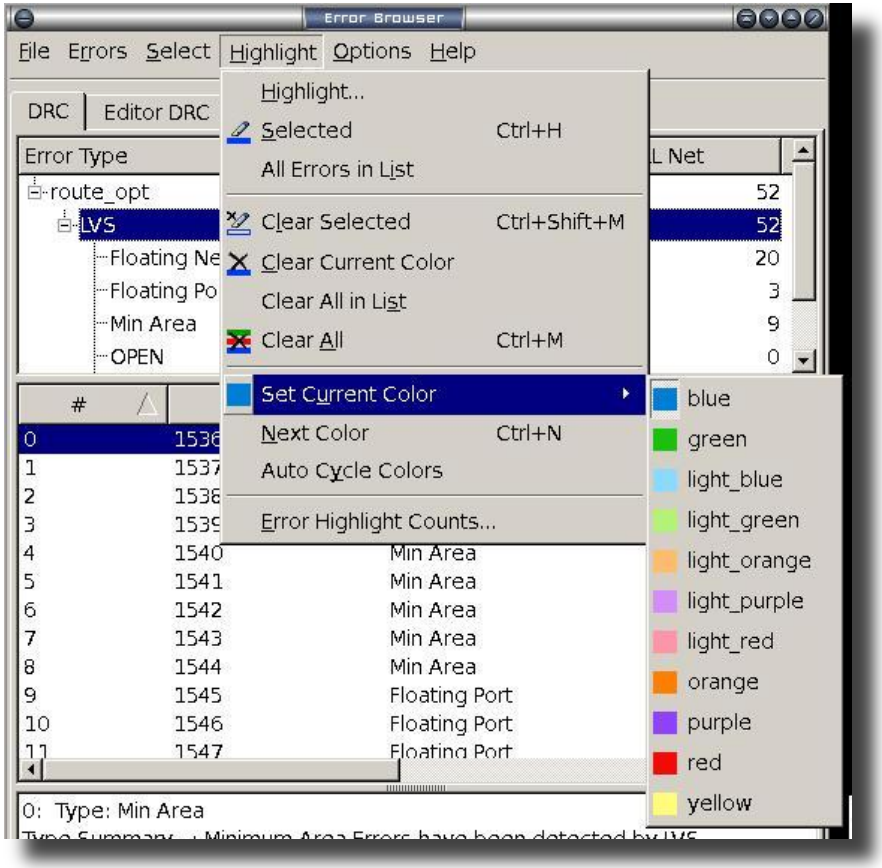
- Displays a histogram of the number of errors that have been colored in each color with the largest count at the top
- Histogram updates as counts change
- Error Browser GUI
  - Highlight>Error Highlight Counts



Error Highlight Counts Dialog

Command and UI
Error Highlight Commands

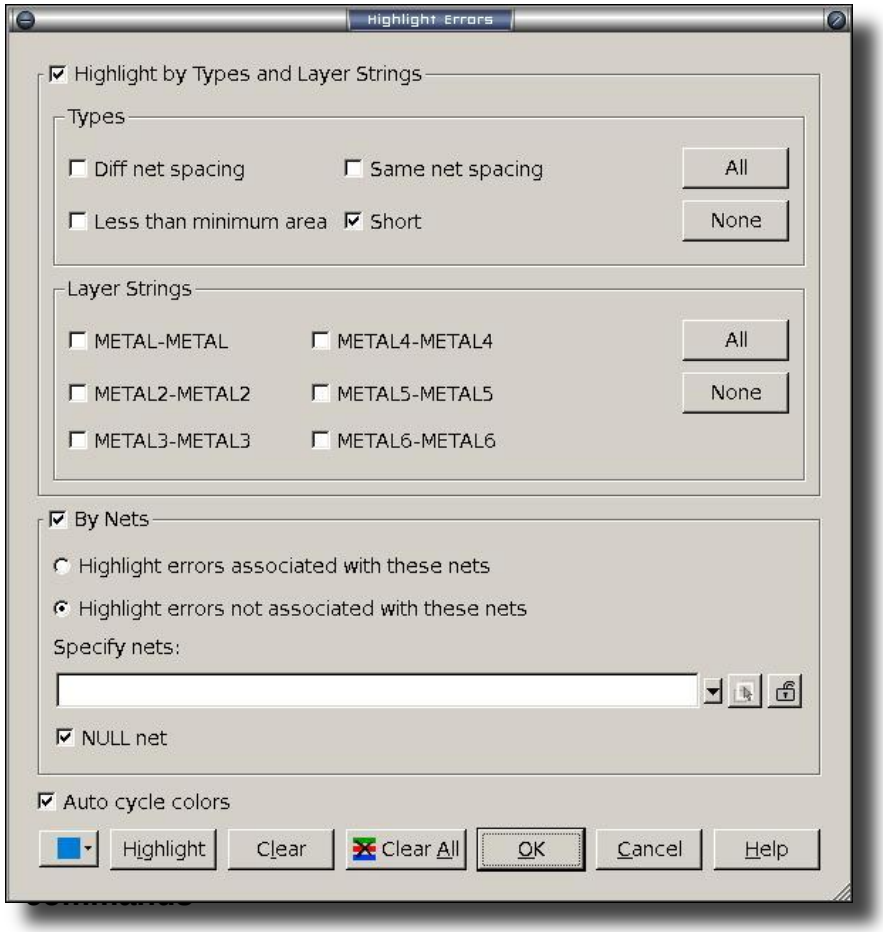
- Provided in the **Highlight** menu
- Highlight colors are not saved in the database and will not be retained when the error view is closed



The Error Browser Highlight menu with the coloring commands

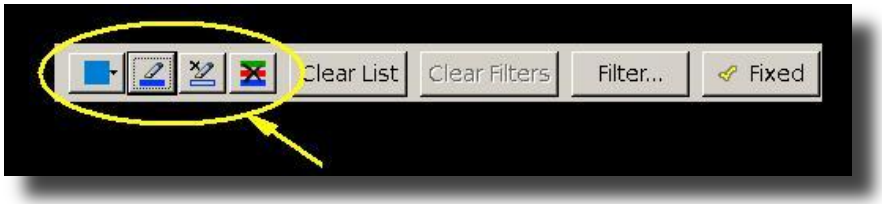
Command and UI
Highlight Errors Dialog

- New dialog for highlighting errors using type, layer and net association criteria
- Resembles the filtering dialog but allows the user to set error highlight rather than to filter visible errors
- Error Browser GUI
  - Highlight->Highlight



Command and UI
Command Buttons

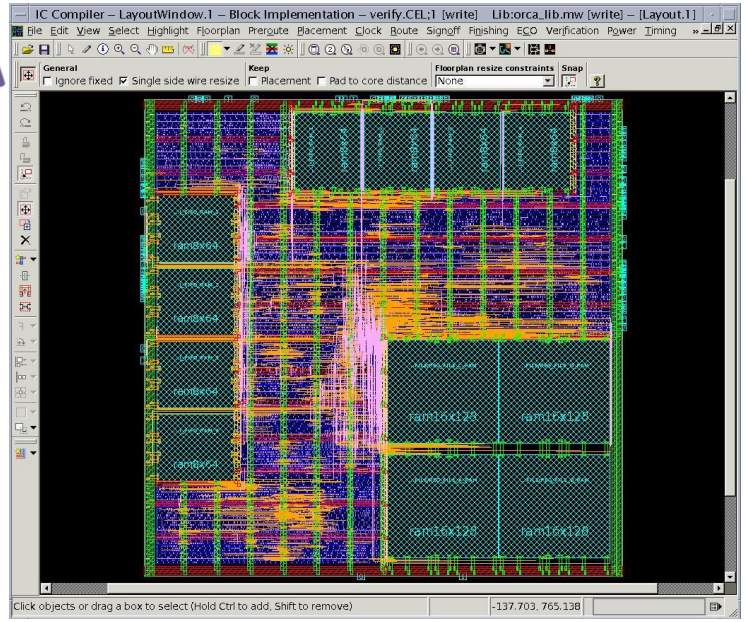
- Command buttons for **Set Current Colour**, **Highlight Selected**, **Clear Selected**, and **Clear All** commands described above are provided under the error list for easy access



Error highlight command buttons under the error list

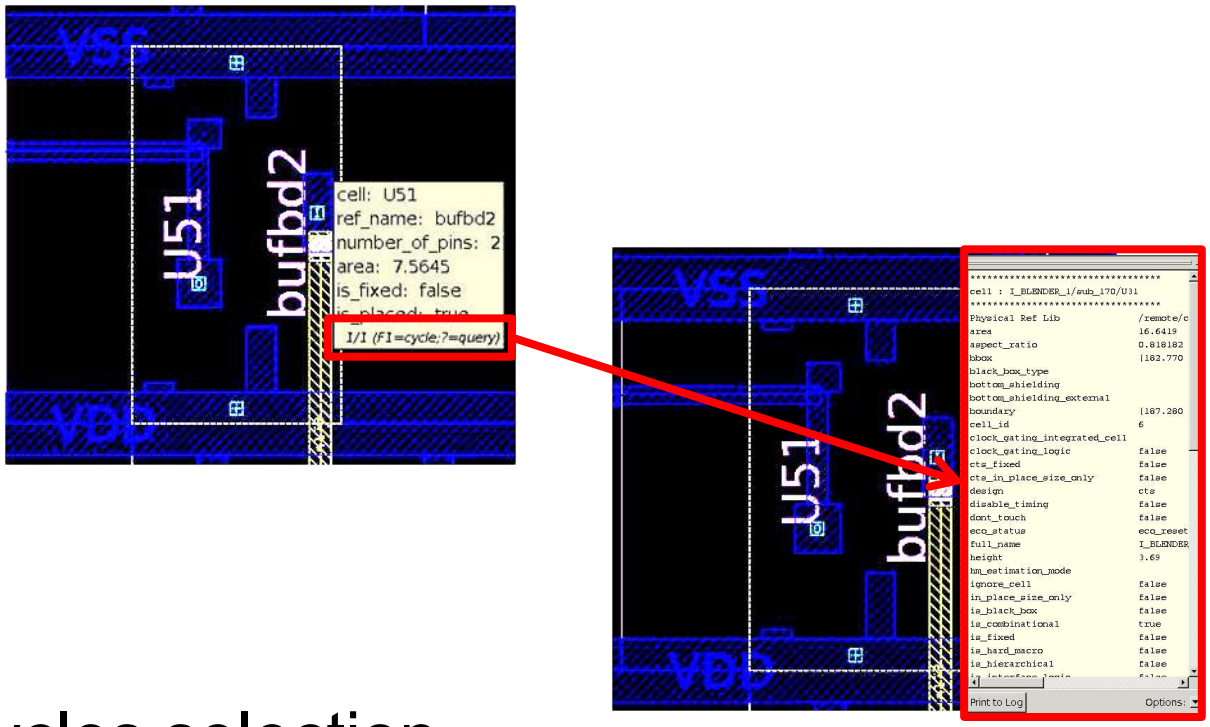
Selection Tool

- Object selection modes
  - Smart
    - Supports point or rectangle with add/remove/replace
    - Selection preview and cycling is supported
  - Rectangle
    - Supports only rectangle input
    - Allows nested zoom/pan operations
  - Line
    - Supports select for intersection



Selection Preview

Info Tips

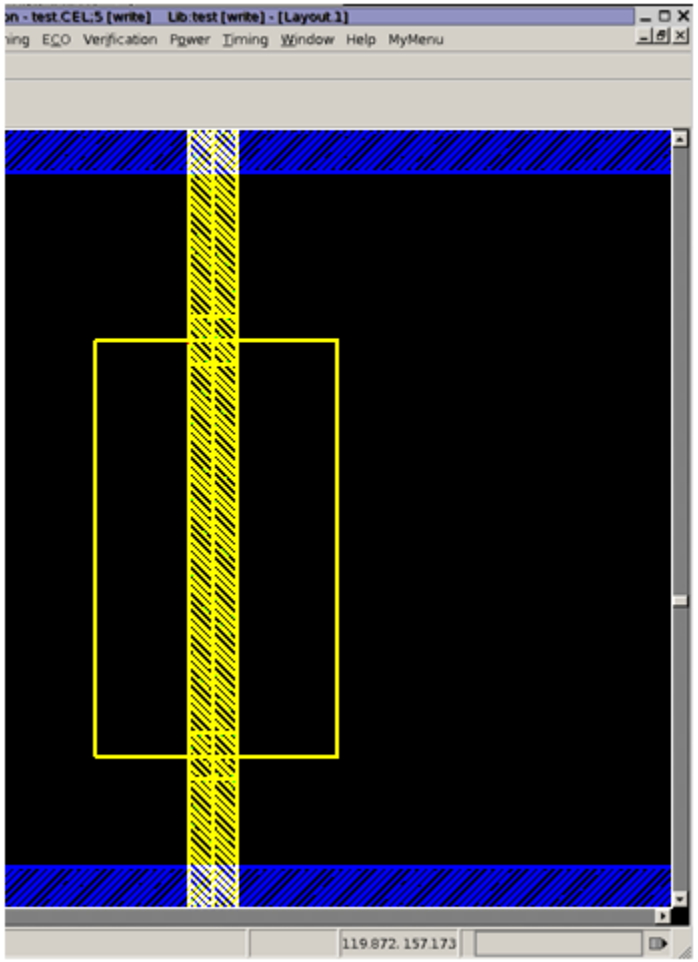


- "F1" cycles selection
- " ?" Shows more detailed information

Mainstream Editing Improvements

Split

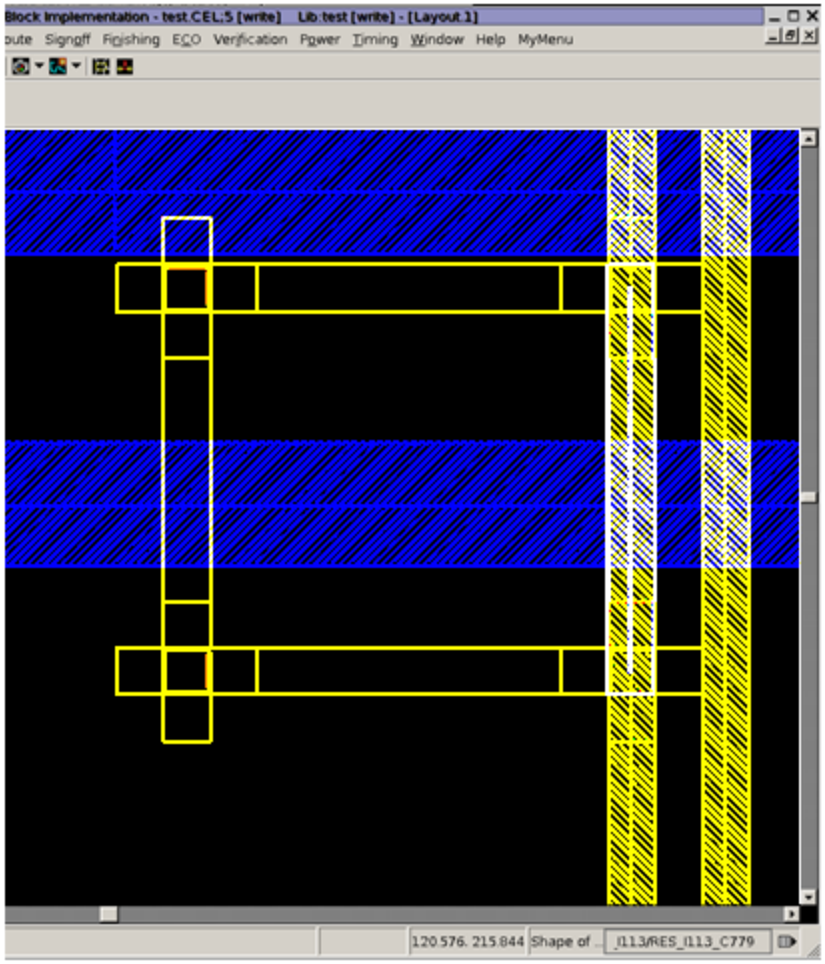
- Any angle line & rectangle
- Cycle selection of split objects
- WYSIWYG during split
- Uses mouse tool options bar
- Split of via arrays



Mainstream Editing Improvements

Stretch Wire & Window Stretch

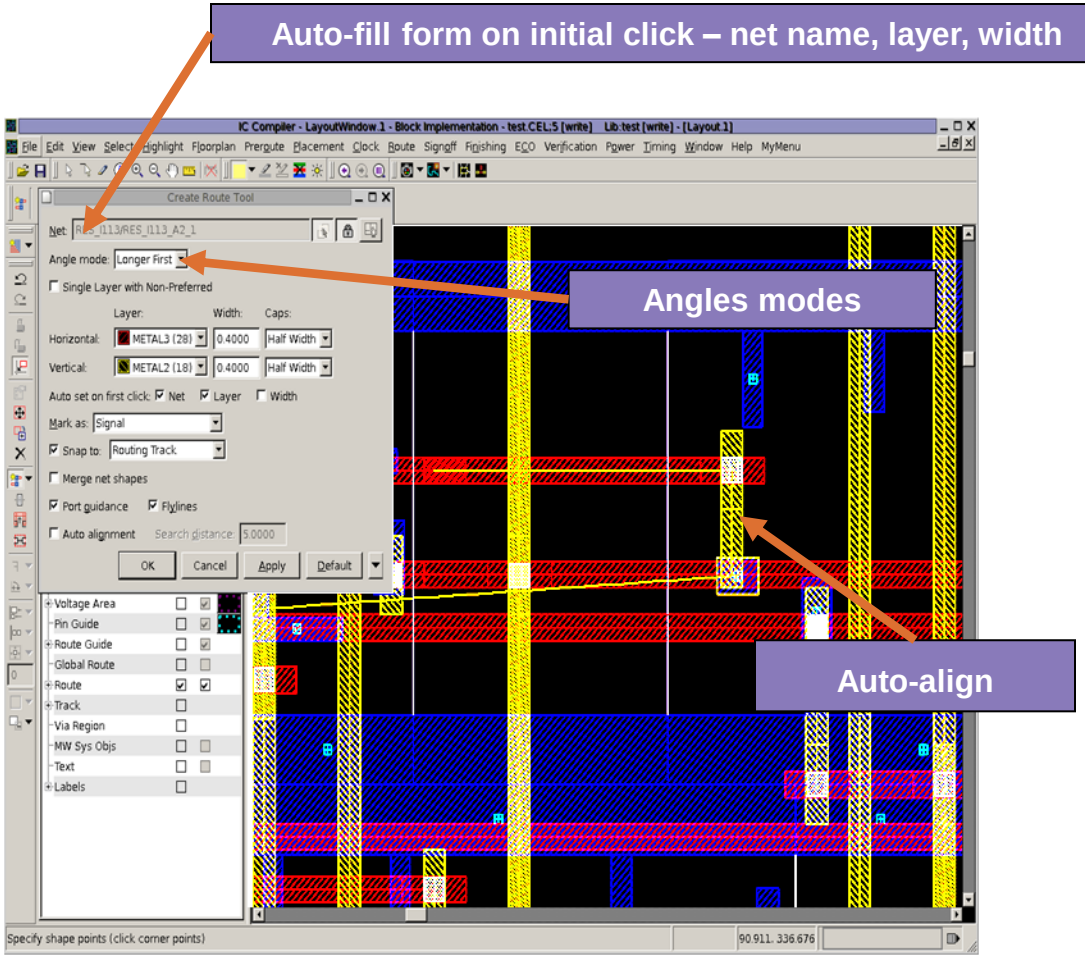
- Improved connectivity support
- WYSIWYG during operations



Mainstream Editing Improvements

Create Route Tool

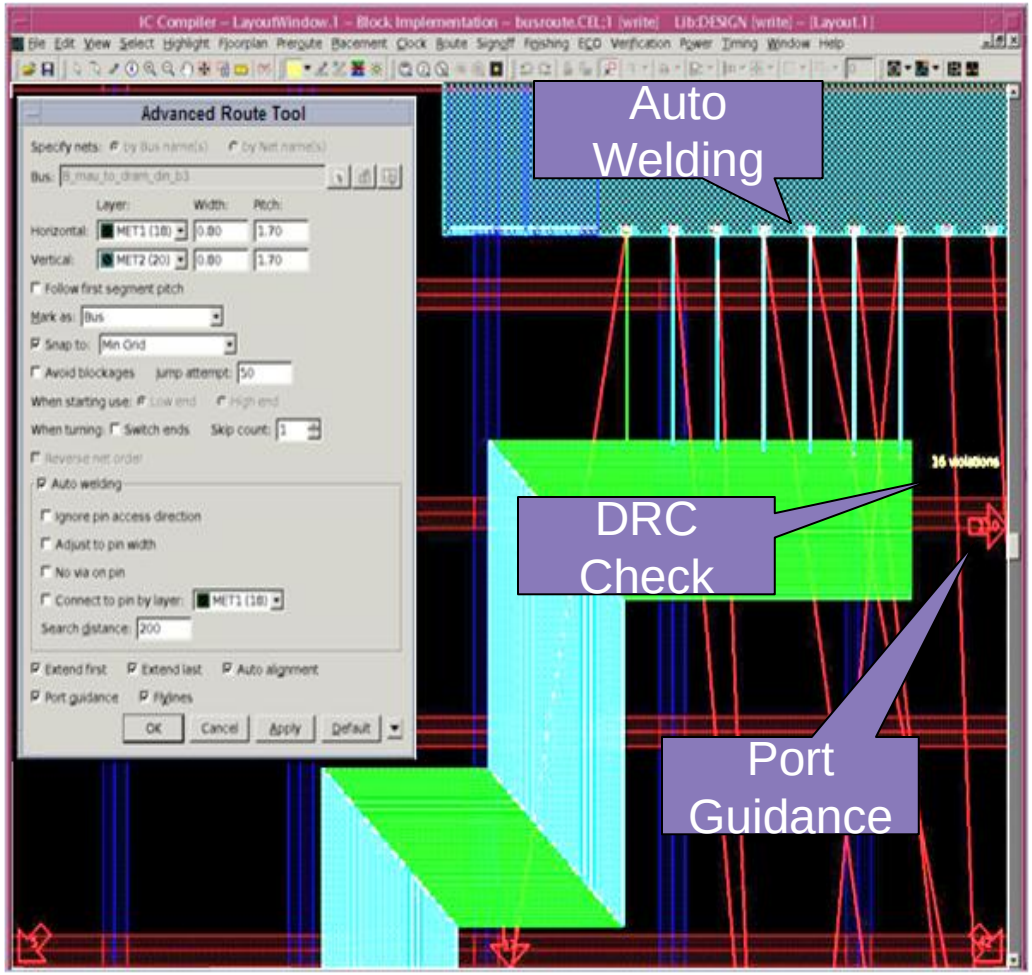
- Auto-fill form on initial click
- Angle Mode
- Port Guidance
- Open Flylines
- Auto-align
- Automatic Via Dropping



Advanced Route Tool

Bus/Net Route

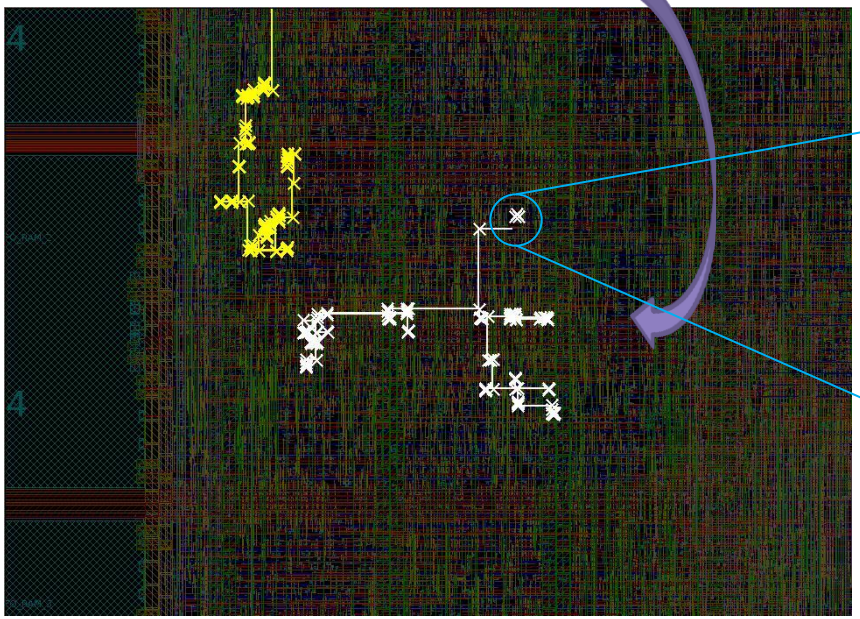
- Targeted at pre-routes and trunk routing
- Blockage avoidance
- Automatic welding and snapping
- iDRC for on the fly error checking



verify_lvs Open Locator

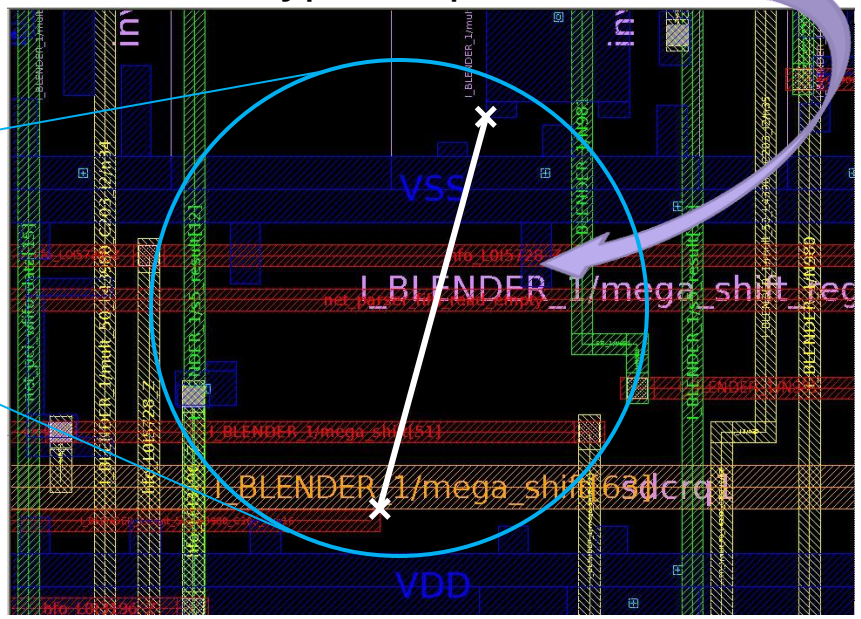
- Open Locator indicates the closest open pairs which should be connected together

Error type - Open



(Figure 1) - Net I_BLENDER_1/mult_50_L43980_C203_I2/n16

Error type – Open Locator



(Figure 2) - Net I_BLENDER_1/mult_50_L43980_C203_I2/n16

Display of same net in error types “open” & “open locator”

2012.06 Route Editing Enhancements

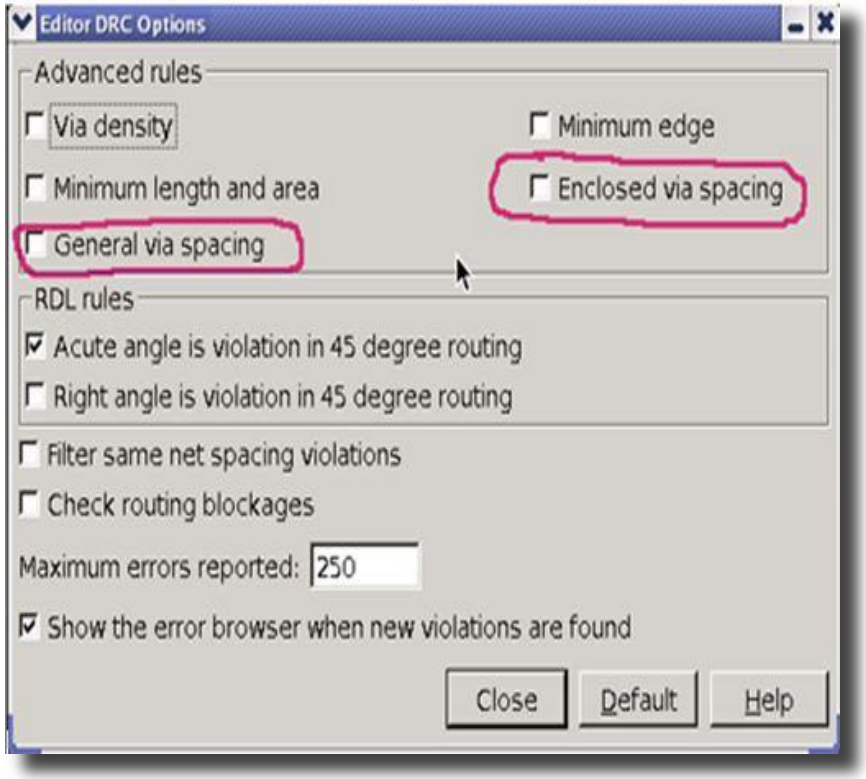
- New Via Rules Support in Route Editing
  - Enables the GUI route editor to use several new via rules introduced in the technology file for 45-nm and below process nodes

- GUI
  - Two check boxes added to the Editor DRC options dialog as options for interactive DRC checking
  - Off by default

New Via Rules Support in Route Editing

- Enhanced via selection
  - Area-based fat metal contact rule (45-nm)
  - Area-based fat metal extension contact rule (45-nm)

- New Editor DRC rule checking
  - Enclosed via spacing rule (65-nm)
  - General cut spacing rule (28-nm)



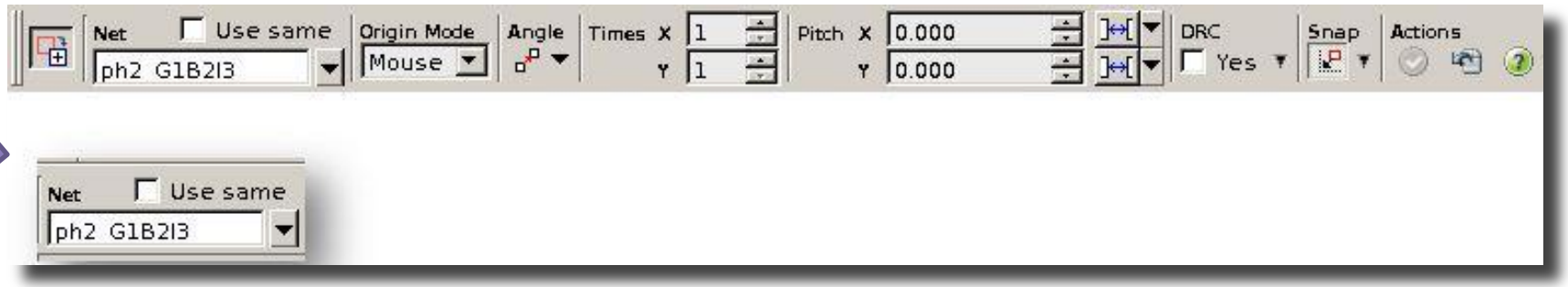
New Via Rules Support in Route Editing

- Editor DRC checks many new routing rules currently used by Zroute during route editting
- When making a net connection, more rules are used to select a via

- Flow recommendation
  - Enable the Editor DRC before performing route editing operations

Specifying the Net for a Copied Object

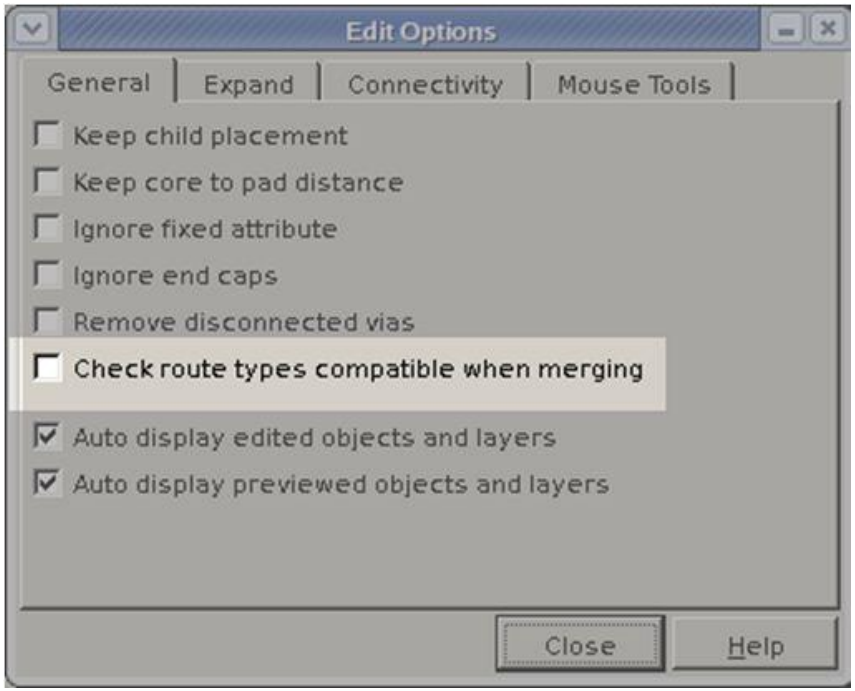
- You can specify the net for the copied object



  - Select ‘Use same’ to use the same net (which is the default)
  - You can specify the destination net by using a standard net selection control

Checking Route Type When Merging Net Shapes

- Verifies that the route types of the net shapes are the same when merging two net shapes
- If they are not the same, the merge fails



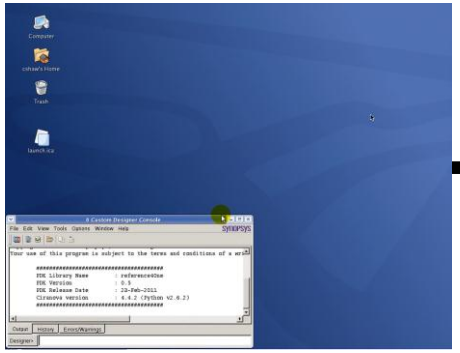
Physical Closure

- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

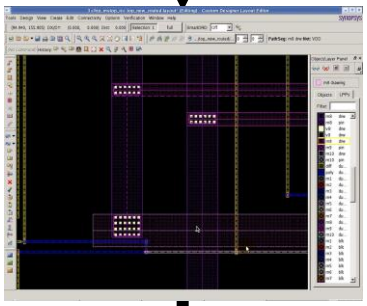
IC Compiler Custom Co-Design

- **Seamless integration between IC Compiler and Custom Designer**
- **Full roundtrip editing capabilities**
- **Robust mixed-signal layout environment**
- **Push Button set-up**
- **No techfile required!**
- **No PDK required!**
- **All rules come from IC Compiler**
- **Intuitive simple use model**

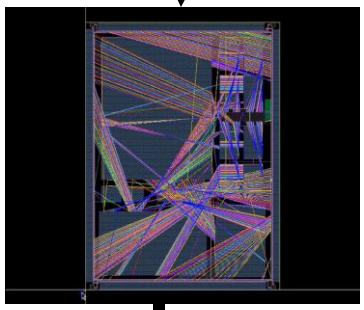
The Co-Design Flow



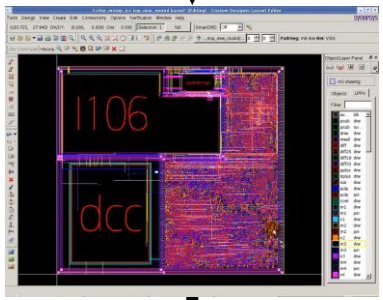
Open design in CD



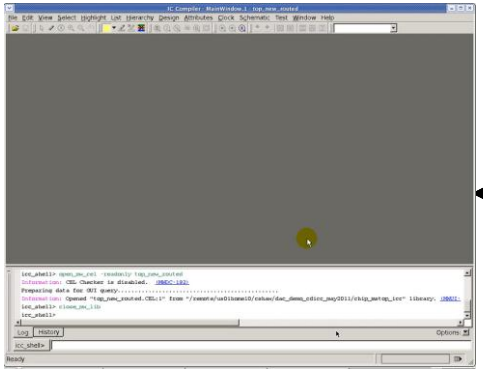
45 deg PG edit



P2P Wide Wire Routing



Add probe-pad

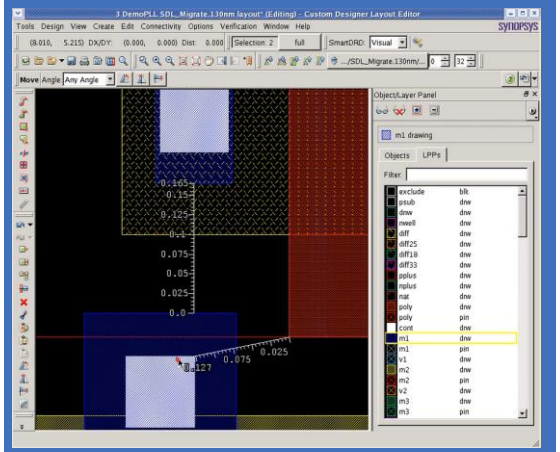


Save design back to ICC

2011.09 Release Highlights (cont.)

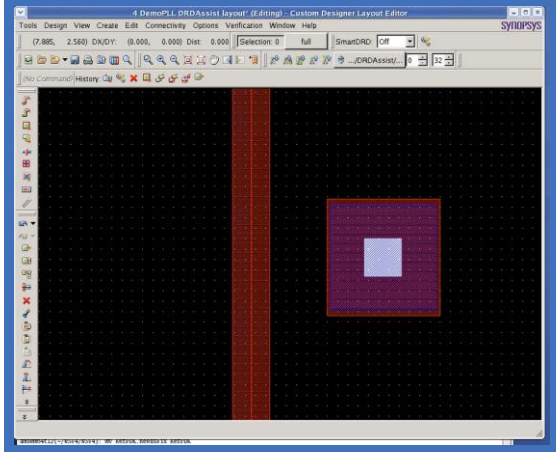
SmartDRD to Automate Complex Rule Compliance

DRD Visual



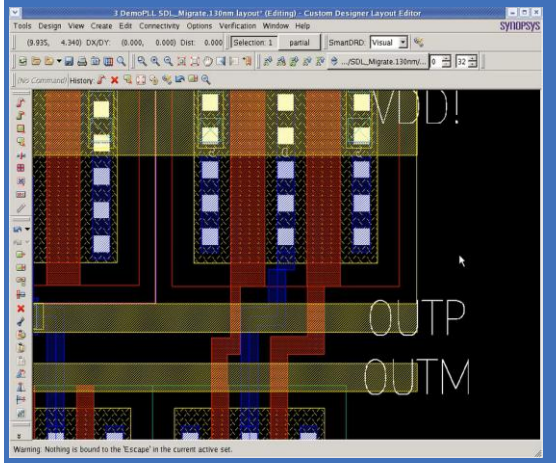
- High-capacity real-time DRC checking
- Violations are shown with error markers

DRD Assist



- Real-time soft design rule enforcement while editing
- Automatic override for ease of use

DRD Auto-fix

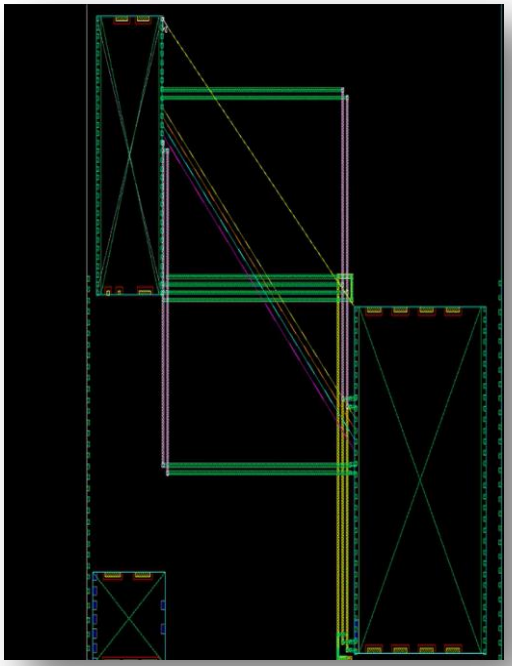


- Real-time DRC error fixing
- Simple point and click window-based operation
- Minimum perturbation algorithms automatically fix DRC violations

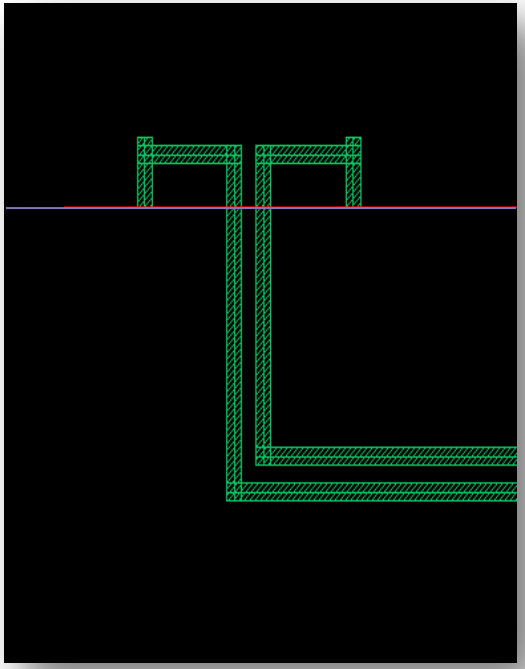
Interactive Router – Mixed Signal

For Analog & High Speed Nets in IC Compiler Designs

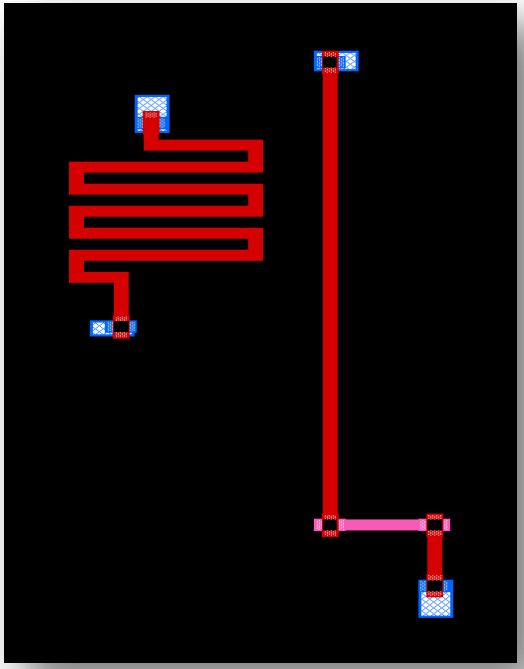
Shielded Nets



Differential Pair



Matched length or resistance*



*Autorouter Only

Included with Custom Designer CR product

Summary

- **Seamless integration between IC Compiler and Custom Designer**
- **Full roundtrip editing capabilities**
- **Robust mixed-signal layout environment**
- **Push Button set-up**
- **No techfile required!**
- **No PDK required!**
- **All rules come from IC Compiler**
- **Intuitive simple use model**

Physical Closure

- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

Sign-Off Automatic DRC Fixing Flow

- Overview

  The sign-off automatic DRC fixing flow is available as a GA feature in E-2010.12 with the following features:

  – Tightly integrated flow that repairs sign-off DRC violations on the current open cell in IC Complier

  – Incremental area-based revalidation after each repair loop

    − Improved runtime over full-chip revalidation flow

Sign-Off Automatic DRC Fixing Flow

- The automatic DRC fixing flow accelerates DRC closure by automating the manual DRC closure design step
- Design must have adequate available routing resources to enable automatic DRC fixing
- Fix rate is highly design dependent
- Not targeted for
  - High DRC counts that would not be handled manually
  - Placement congestion issues

Sign-Off Automatic DRC Fixing Flow

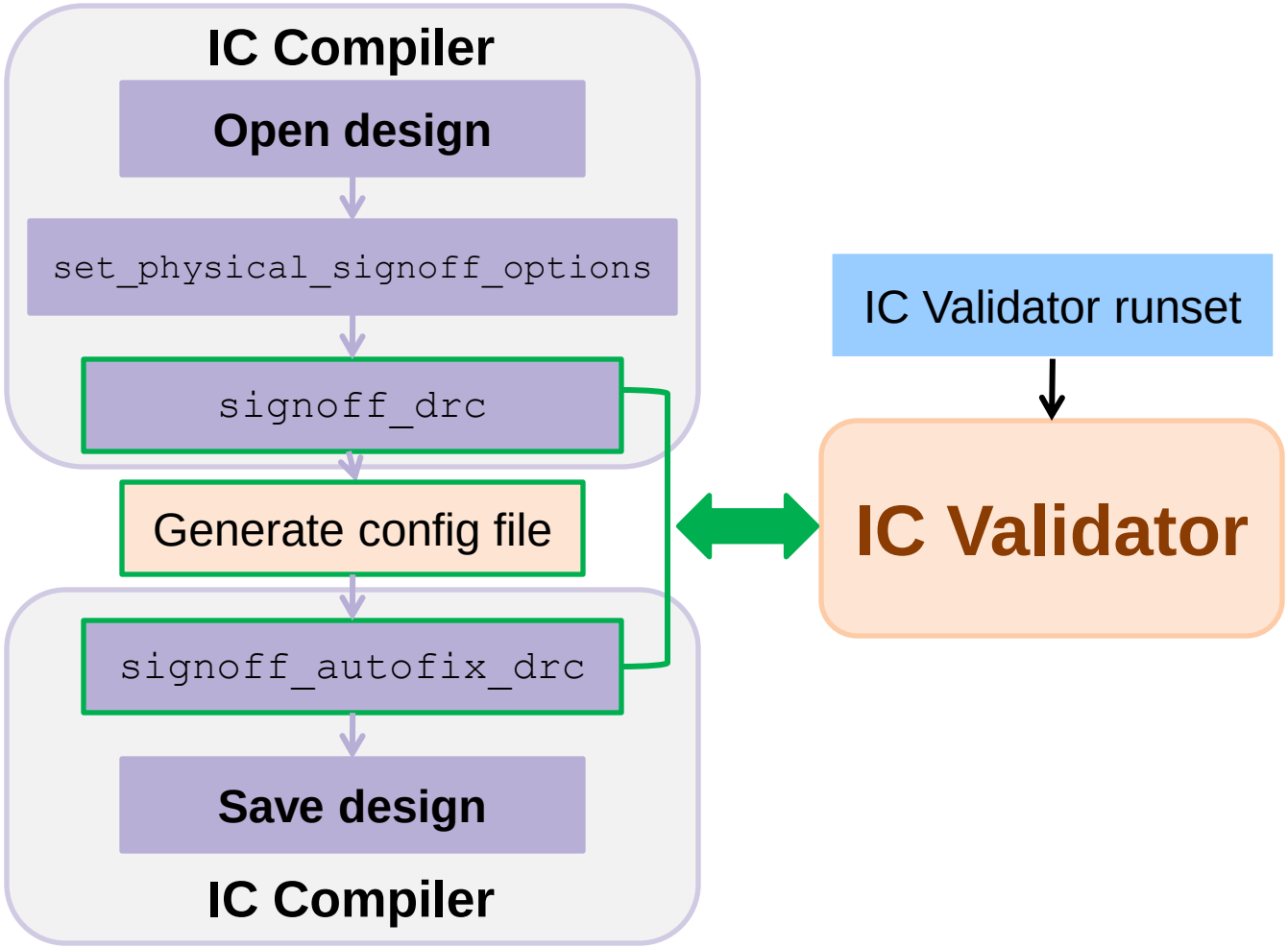
- What DRC violations can be addressed by automatic DRC fixing flow?
  - DRC violations that are not visible to the router or flagged by sign-off verification using IC Validator
    - Rules left out of ICC tech file because of low rate of occurance
    - Rules that are not supported by the router in the early stage of the development of a technology
    - There is a data mismatch between the FRAM view and CEL view of a cell or macro

Sign-Off Automatic DRC Fixing Flow

- What DRC violations are not addressed by the automatic DRC fixing flow?
  - Power and ground nets
  - Clock nets
  - Shielded nets
  - Freeze nets, including user-enter nets
  - Antenna violations
  - Violations between metal fill and routing shapes
    - The automatic DRC fixing flow is intended for DRC fixing prior to metal fill insertion

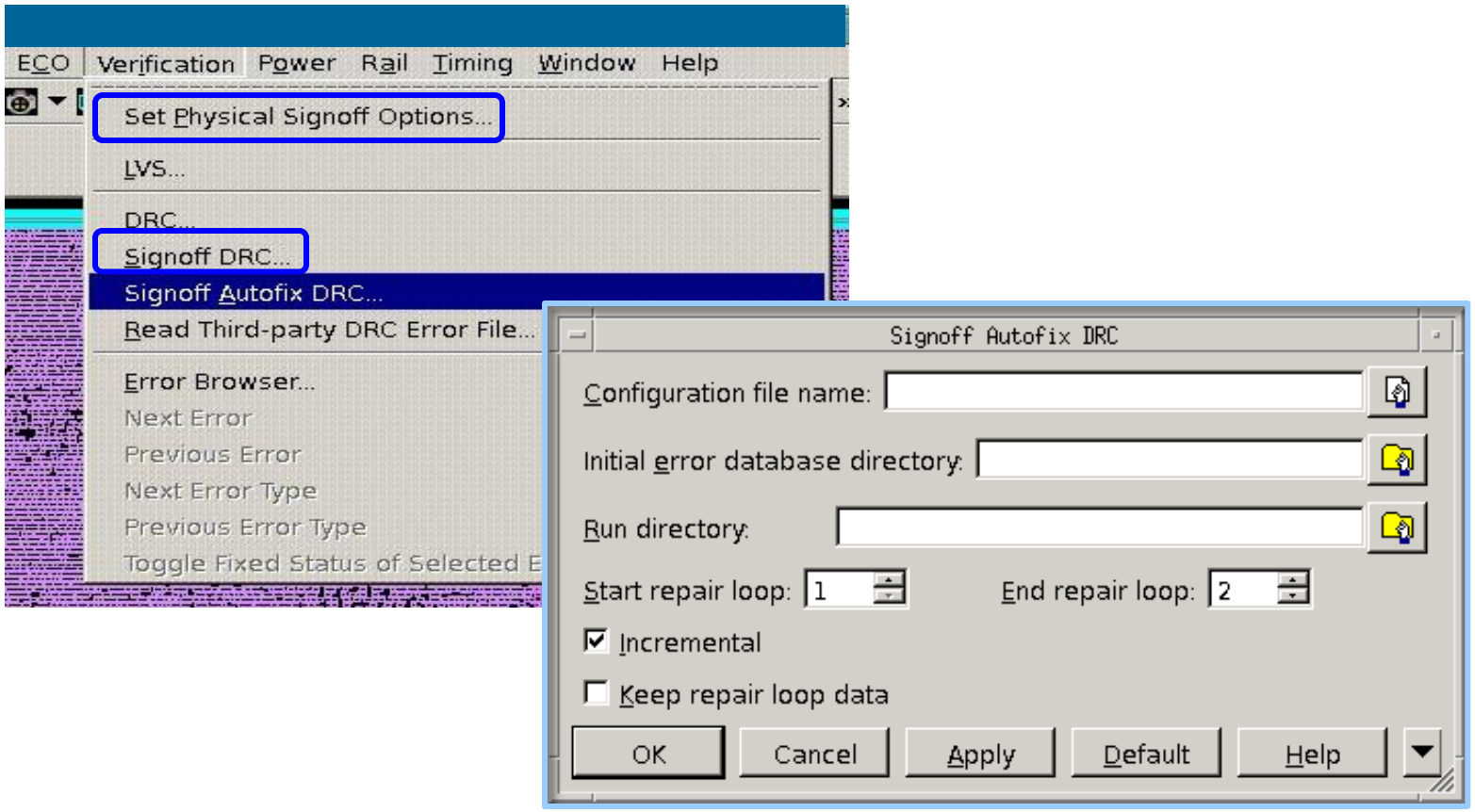
The automatic DRC fixing flow is not targeted for DRC violations that the router sees but cannot fix

Sign-Off Automatic DRC Fixing Flow



Sign-Off Automatic DRC Fixing Flow

- GUI



Sign-Off Automatic DRC Fixing Flow

- UI

```
signoff_autofix_drc
-config_file config_file_name
-init_drc_error_db directory_of_input_ICV_run
[-incremental true | false] [default: true]
[-start_repair_loop 1-10] [default: 1]
[-end_repair_loop 1-10] [default: 2]
[-keep_repair_loop_data ] [default: false]
[-run_dir output_directory]
          [default: signoff_autofix_drc_run]
```

Sign-Off Automatic DRC Fixing Flow

- Flow steps

```
icc_shell> set_physical_signoff_options \
    -drc_runset runset_file
icc_shell> signoff_drc -read_cel_view \
   -ignore_child_cell_errors \
  -user_defined_options {-holding_cell}

In UNIX shell
% $ICV_HOME_DIR/contrib/generate_layer_rule_map.pl \
      -dplog icv_dp_logfile \
      -tech_file Milkyway_technology_file \
      -o output_file

icc_shell> signoff_autofix_drc -config_file config_file \
     -init_drc_error_db input_error_db_dir
```

Sign-Off Automatic DRC Fixing Flow

- Required inputs to the sign-off automatic DRC fixing flow
  - Milkyway database for routed design
    - Near timing closure
    - Ready for manual fixing of remaining DRCs
  - CEL views for standard cells and hard macros
  - Foundry's IC Validator sign-off DRC checking runset
  - Configuration file

Sign-Off Automatic DRC Fixing Flow

- The configuration file
  - Contains the list of rules for which violations are targeted
  - Entries in the configuration file have the following syntax:

    "*mask_layer_name*","*full_DRC_error_comment*"

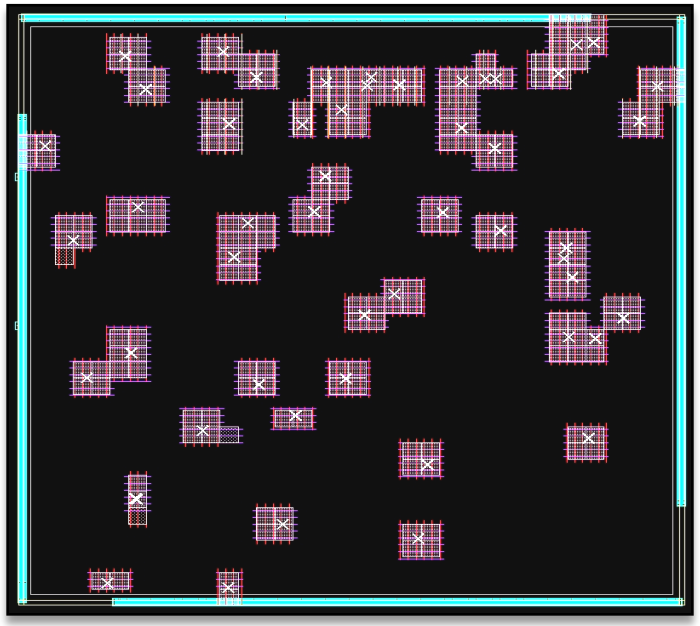
    Example: "metal3", "M3.S.1: M3 Space >= 0.07 um"
  - The specified mask layer should be an appropriate layer on which the fix is targeted for the rule

Sign-Off Automatic DRC Fixing Flow

- Outputs from the sign-off automatic DRC fixing flow
  - Auto-fixed design database as current open CEL, ready to be saved
  - IC Validator run data
    - By default, from the last repair loop only
    - With `-keep_repair_loop_data`, from every repair loop
  - Log file

Sign-Off Automatic DRC Fixing Flow

- Incremental DRC revalidation in default flow
  - Restricts `signoff_drc` validation of the auto-fixed design to only the areas affected during rerouting



Sign-Off Automatic DRC Fixing Flow

- **Important!** The final validation in the default flow (incremental = true) is **not** strictly sign-off accurate
  - Validation does not cover the full chip
  - The violations that are not targeted are not reported
- You should do a full-chip `signoff_drc` run after the default sign-off automatic DRC fixing flow to get an accurate sign-off DRC count

Sign-Off Automatic DRC Fixing Flow

- Benefits
  - Identification of DRC violations based on the foundry sign-off DRC runset
  - Targeted automatic DRC fixing, which replaces manual DRC fixing in the postroute or postroute ECO stages
  - Negligible impact on timing due to the limited scope of expected changes in the layout
  - Reduction in the number of costly iterations between place and route and full-chip sign-off verification

Sign-Off Automatic DRC Fixing Flow

- Limitations
  - The sign-off automatic DRC fixing flow does not reinsert redundant vias during repair. You should run standalone redundant via insertion after automatic DRC fixing, if needed.
- Required releases
  - IC Compiler: E-2010.12
  - IC Validator: E-2010.12
- Document
  - *IC Compiler Implementation User Guide*

2012.06 signoff_autofix_drc Enhancements

- Runtime improvement for the incremental flow
- Support for user-defined IC Validator options
- Support for selected rule checking

Runtime Improvement for the Incremental Flow

- **UI**

```
signoff_autofix_drc
   -incremental_level off | low | high
```

	`off`	`low` (default)	`high`
DRC checking	Full design	Areas targeted by automatic DRC fixing only	Areas targeted by automatic DRC fixing only
DRC fixing	Full design	Full design	Areas targeted by automatic DRC fixing only

  - When set to **`high`**,
    - Targets improved runtime with possibly higher DRC count
    - Cannot use with **`-keep_repair_loop_data true`**

signoff_autofix_drc Flow Enhancements

- UI support for
  - User-defined IC Validator options
  - Selected rule checking

The following new options are added to the **signoff_autofix_drc** command.

```
signoff_autofix_drc
    -user_defined_options user_options
    -select_rule rule_names
    -unselect_rule rule_names
```

Support for User-Defined IC Validator Options

- UI

  signoff_autofix_drc
  -user_defined_options *user_options*

  – Specifies additional options for the IC Validator command line
  – Specified string is added to the command line used to invoke DRC checking in IC Validator
  – Example

```
signoff_autofix_drc -config_file config_file \
   -user_defined_options {-holding_cell}
```

Support for Selected Rule Checking

- UI

  signoff_autofix_drc
  -select_rule *rule_names*
  -unselect_rule *rule_names*

  – By default, all rules are checked
  – Specifies the rules to check or exclude
  – If you specify both the **-select_rule** and **-unselect_rule** options, the unselected rules are removed from the selected rules
  – The rule names are specified in the COMMENT section in the runset file
    - You can specify the rule name or number
    - You can use wildcards

Error Categorization by IC Validator

- signoff_drc now categorizes all errors by net types and route types of polygons that interact with the error markers
- Results are stored in the signoff_drc error view
- User can use IC Complier Error Browser to filter the certain categories so that the user can selectively analyze the interested violations.

Commands and UI

```
signoff_drc
    -categorize_errors true | false
```

- Default: **`true`**
- The option -categorize_error_types is no longer supported

Usage

- **`signoff_drc`**
  - Default setting recommended

- **`signoff_autofix_drc`**
  - “-categorize_errors” set to false for signoff_drc calls between the repair loops during ADR flow
  - “-categorize_errors” set to true for last signoff_drc call of ADR flow

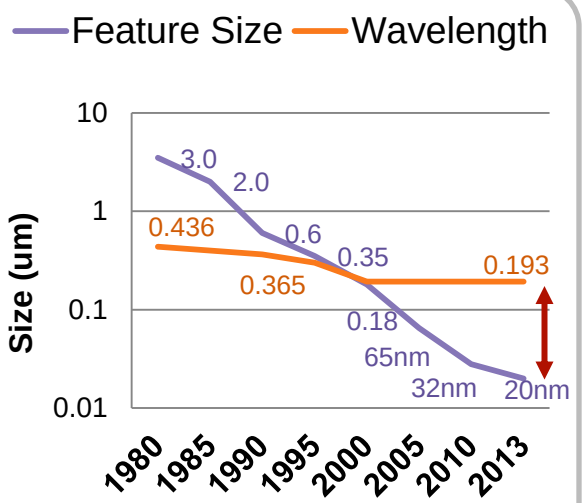
Physical Closure

- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

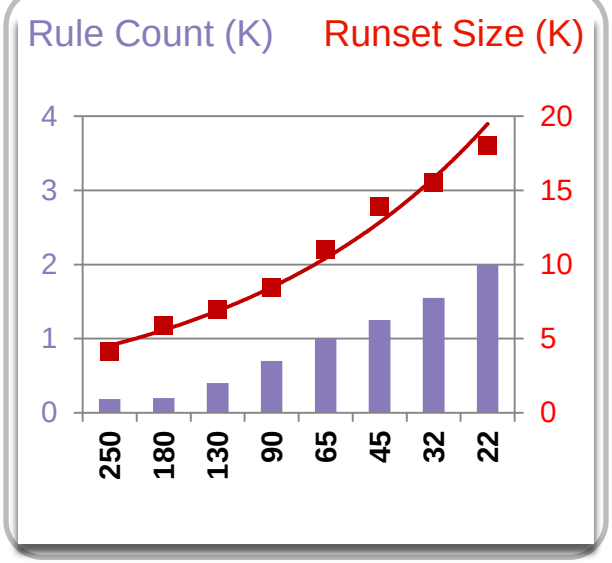
Manufacturing Challenges at ≤28nm

Increased Physical Design Impact on Yield

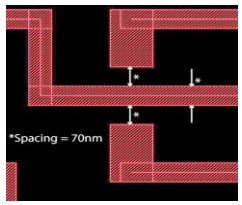
Sub-Wavelength Gap



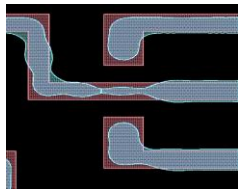
More Highly Complex Rules



Complex Feature Dependencies



DRC-clean



Litho Hotspot

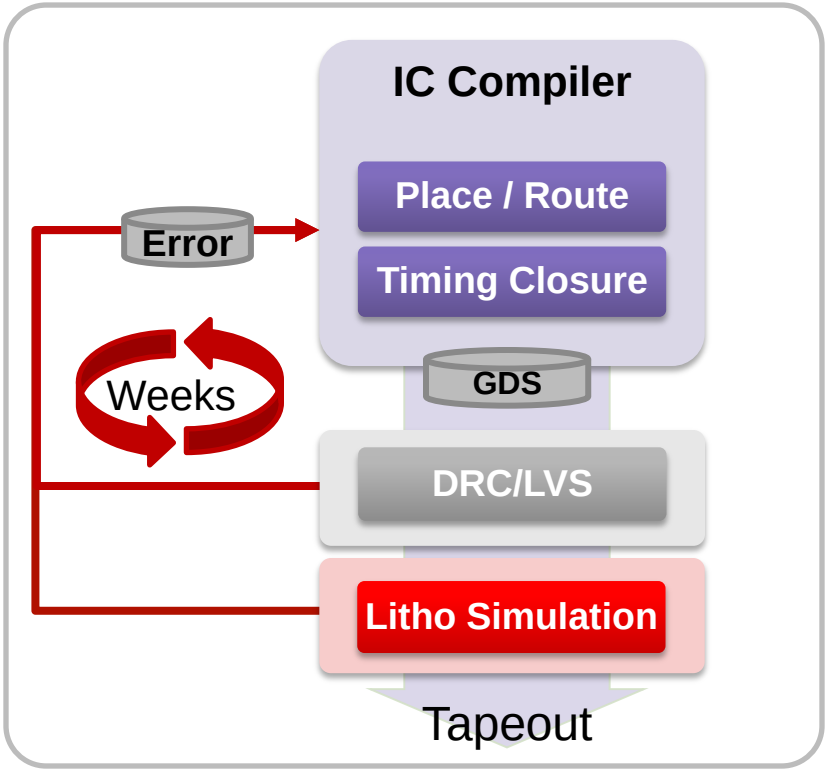
Need for Effective Prevention During Design

SYNOPSYS 25 Years

Traditional “DFM” Flow

Widening Gap with Design Causing Late-Stage Surprises

1. Violations Detected Late
   - Gating tapeout deadline
   - Days of simulation
2. Debugging is Slow
   - Large file streaming
   - Unfamiliar tool environments
   - Limited design context
3. All Fixes Are Manual
   - Productivity bottleneck

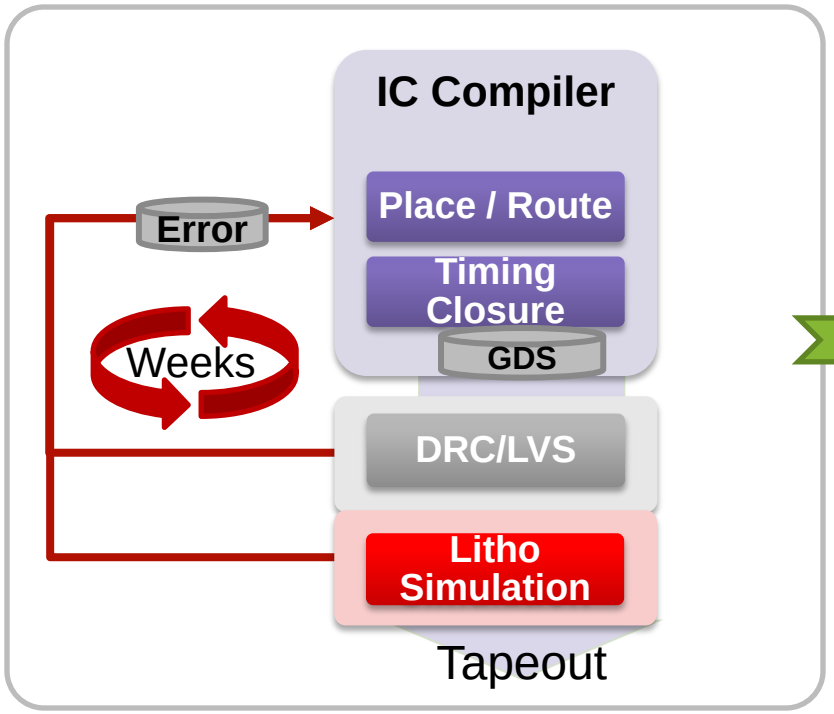


Clear Need for Tighter Design-Verification Integration

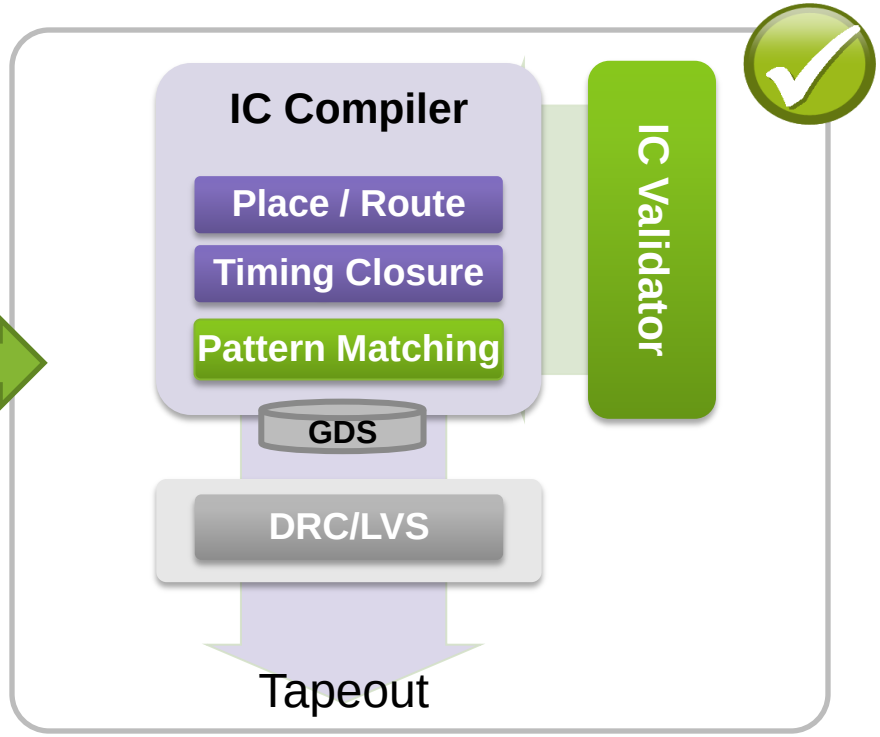
Solution: In-Design Physical Verification

Enables Pattern Based Checking for Faster Closure

Traditional: Iterative Fix-Verify



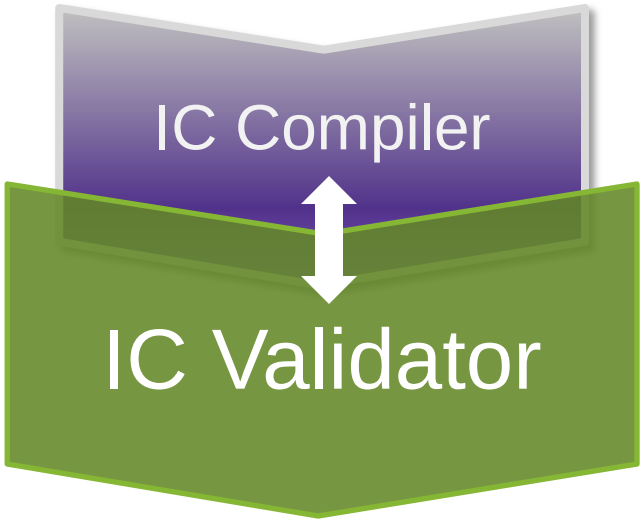
In-Design PV: Pattern Matching



Eliminate Late-Stage Surprises & Manual Fixes

IC Validator

Proven For "In-Design" Physical Verification



High Performance and Scalability

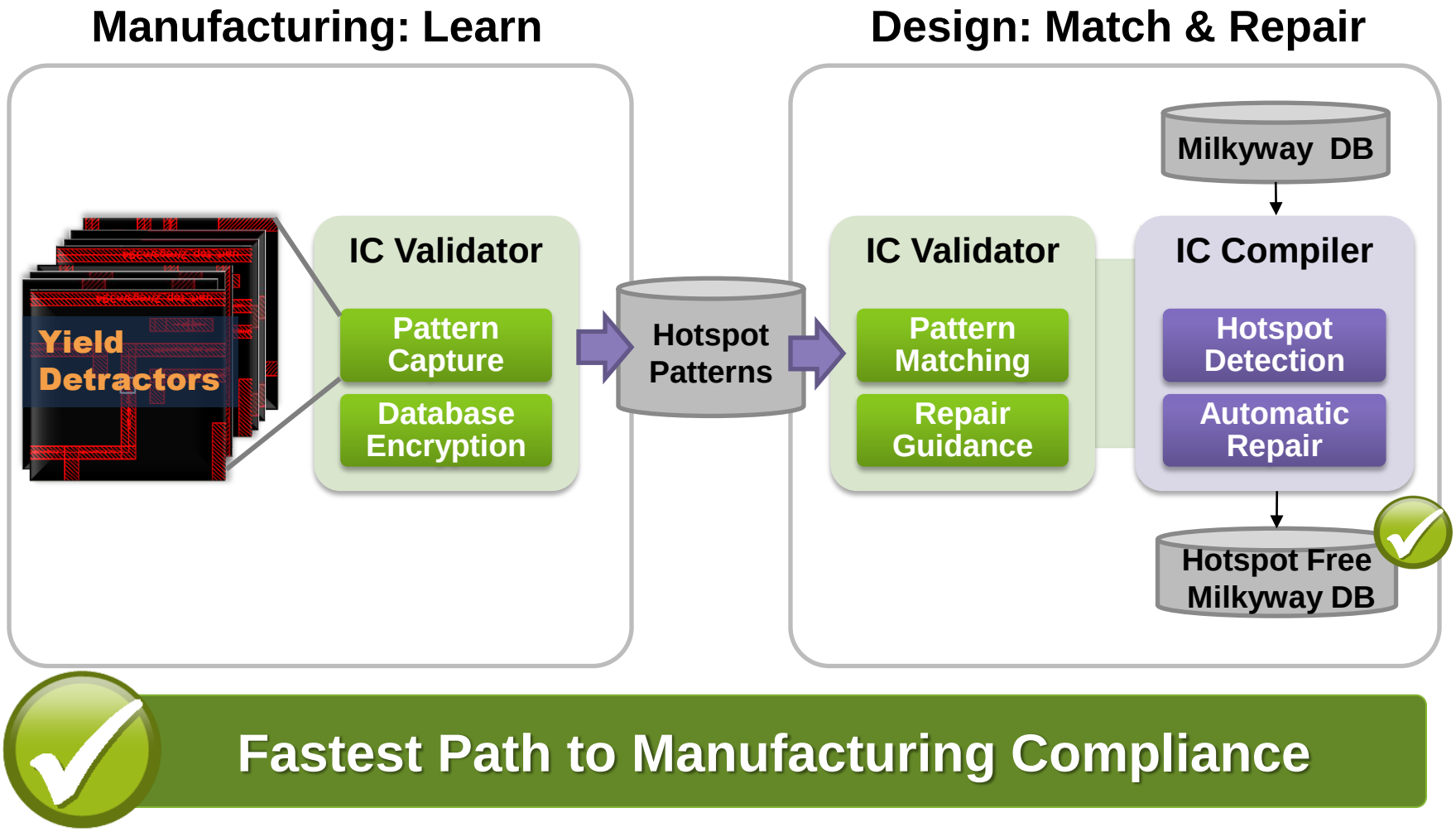
Sign-off Quality Checking

Seamless Integration with IC Compiler

Productivity For The Physical Designer

Pattern Based Verification Flow

Built on IC Validator Pattern Matching Technology

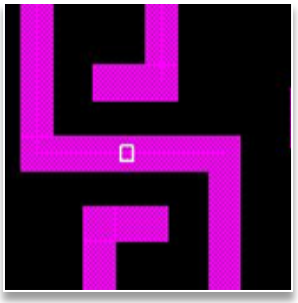


Fastest Path to Manufacturing Compliance

IC Validator Pattern-Matching

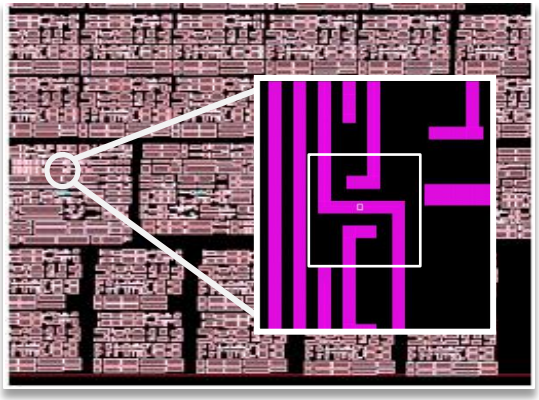
Patented Technology Enables Ultra Fast Detection

Direct Capture



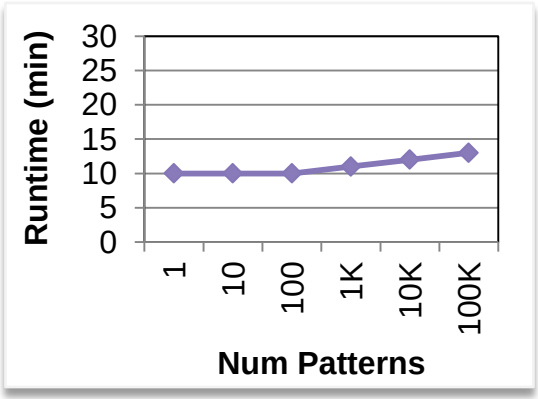
No translation, modeling or rule creation

Accurate Matching



Lossless accuracy for direct and fuzzy matching

Highest Scalability



Zero runtime penalty per pattern

Eliminates Simulation and Convoluted Rules

Results: IC Validator Pattern Matching

Ultra Fast Detection >10,000 Faster than Simulation

Performance of DFM Verification : ICV Pattern Matching

snug
Synopsys Users Group
SINGAPORE 2011

Design Name	A	B	C	D	E	F	G	H
DRC+ ICV Hotspots Found	14	8	35	199	133	1	1	11
Runtime: Multi-threading (8 threads) (minutes)	1.17	0.72	1.22	6.52	3.22	0.62	0.67	1.17
Runtime: Normalized to 1 thread (minutes)	9	6	10	52	26	5	5	9
Peak Memory (Mb)	71	55	102	314	324	38	37	80
Memory Usage (Mb)	248	248	248	248	247	248	247	247

Minutes!

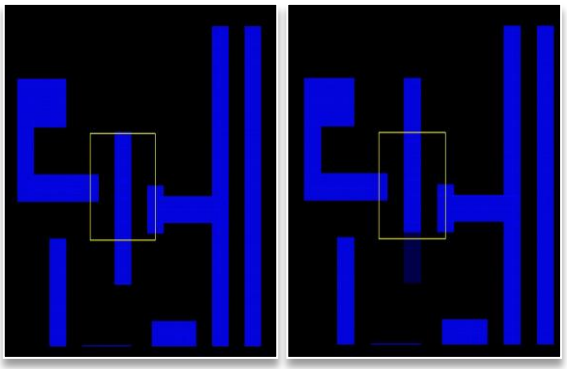
21

SYNOPSYS® 25 Years

Automatic Hotspot Repair in IC Compiler

In-Design Technology Enables Pushbutton Correction

Highly Localized



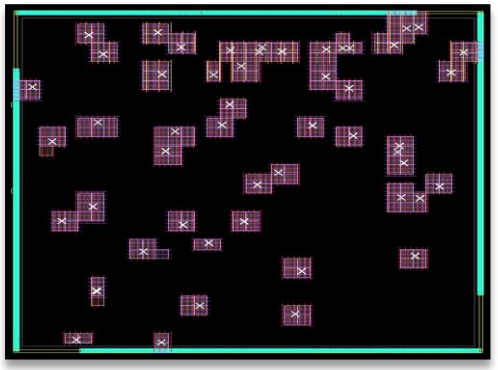
Negligible physical impact

Router-Driven

- Full Property Visibility
- Timing Aware
- Proven Algorithms

Highest quality final layout

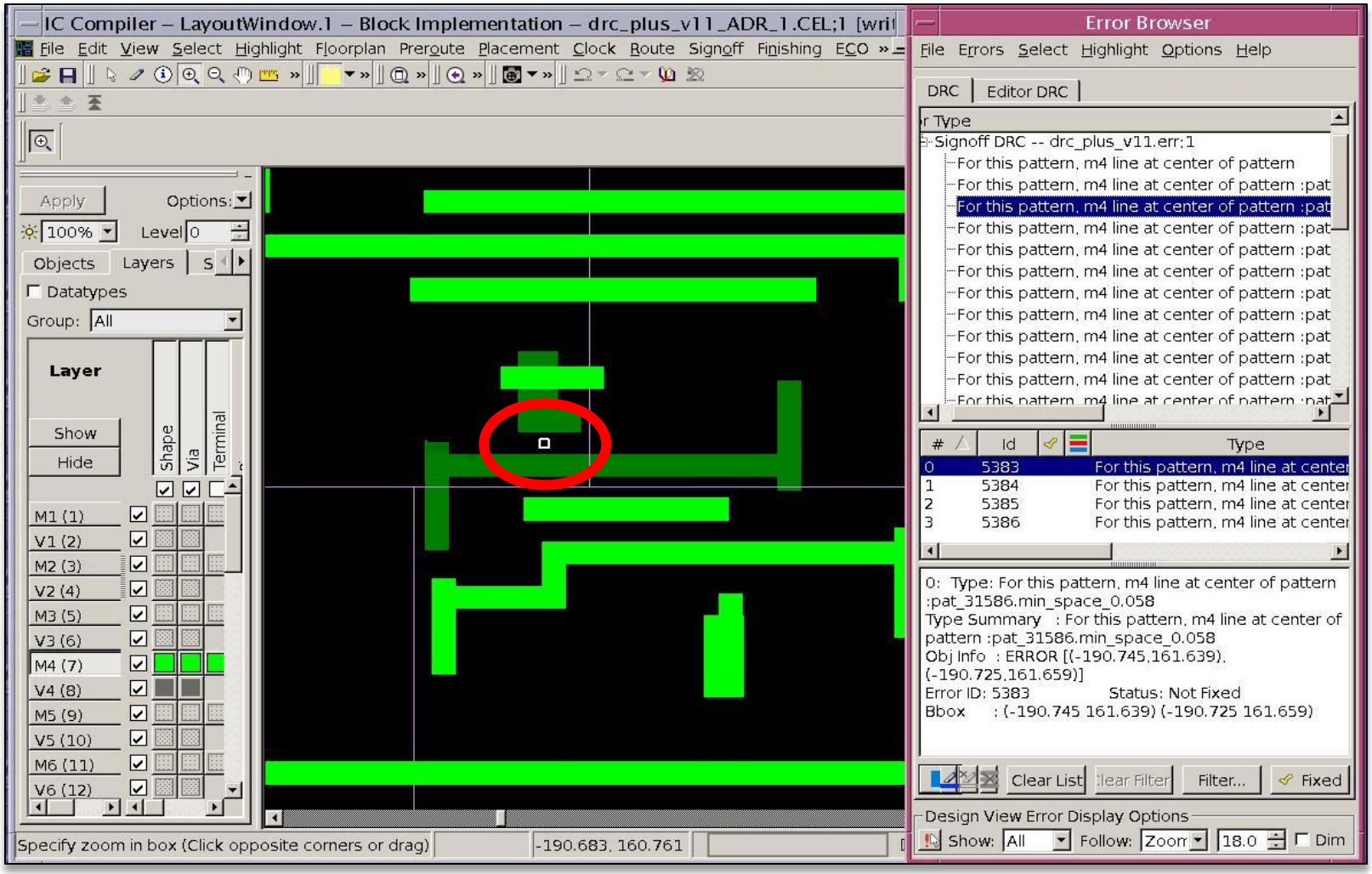
Incremental Validation



Fast repair analysis

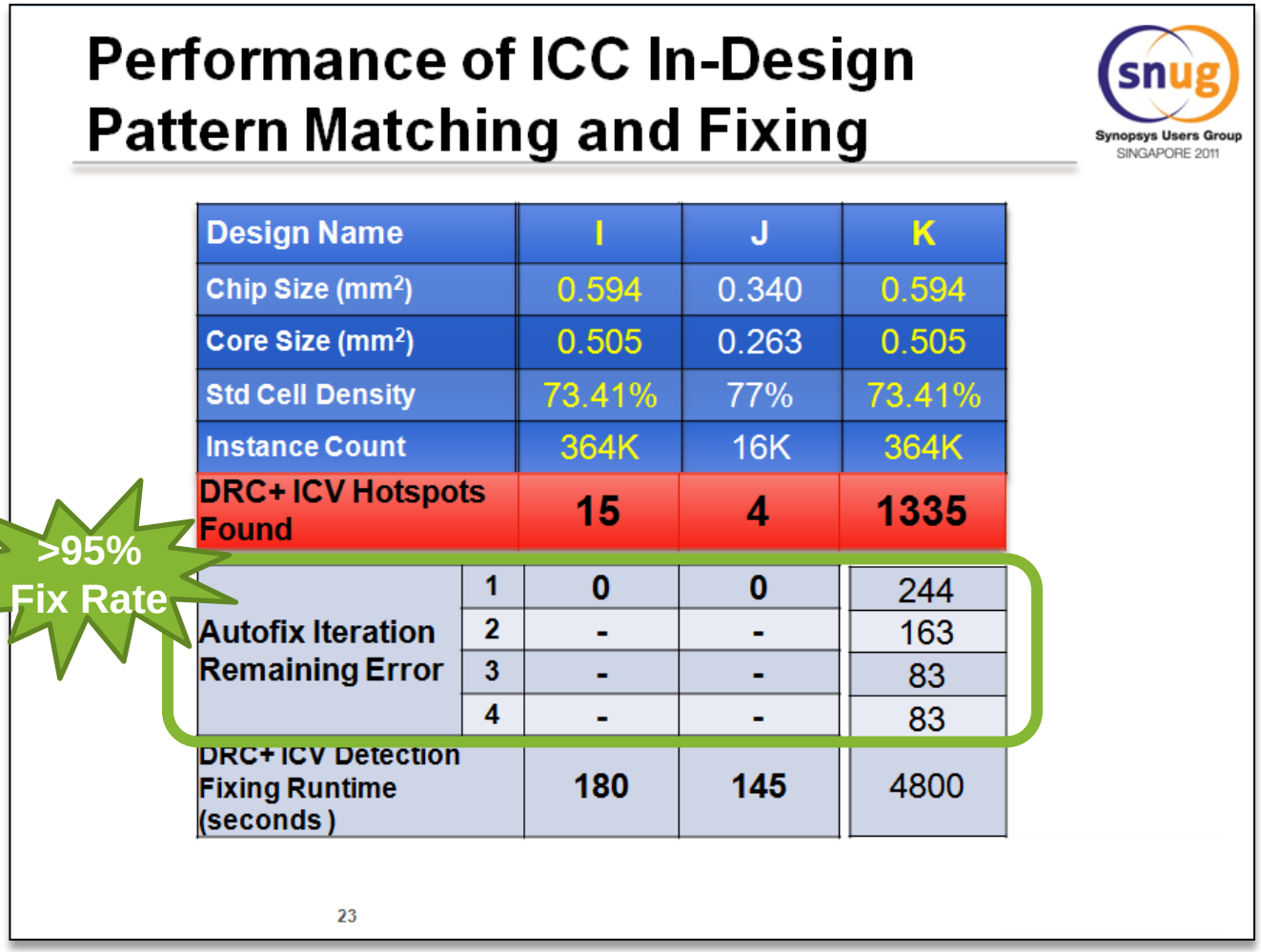
Eliminates Manual Layout Fixes

Automatic Hotspot Repair in IC Compiler



Results: Automatic Hotspot Repair

Fast and Effective Prevention Approach



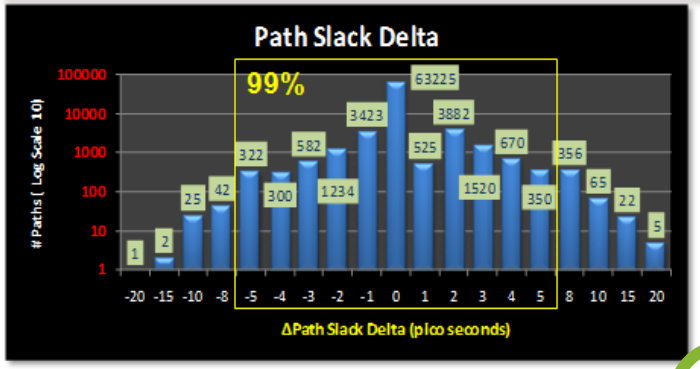
Design Name		I	J	K
Chip Size (mm^2)		0.594	0.340	0.594
Core Size (mm^2)		0.505	0.263	0.505
Std Cell Density		73.41%	77%	73.41%
Instance Count		364K	16K	364K
DRC+ ICV Hotspots Found		15	4	1335
Autofix Iteration Remaining Error	1	0	0	244
	2	-	-	163
	3	-	-	83
	4	-	-	83
DRC+ ICV Detection Fixing Runtime (seconds)		180	145	4800

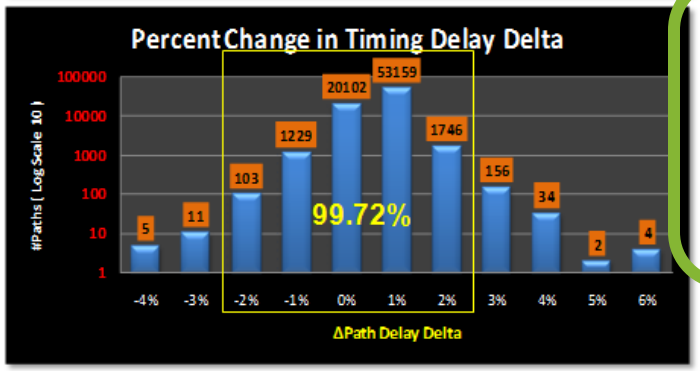
Results: Automatic Hotspot Repair

No Impact on Design Timing

Timing Impact Analysis Results:

snug Synopsys Users Group SINGAPORE 2011





Design analyzed is with 1252 of 1335 hotspots violations fixed inside ICC with 4 Iterations.

- high fixing rate of 94% at 4th iteration with minimal timing impact

Risk Free

- 99% of timing paths has timing delta of only ±5ps.
- 99.72% of the slack paths that changed have path slack changes of ± 2%.

27

SYNOPSYS® 25 Years

Summary and Availability

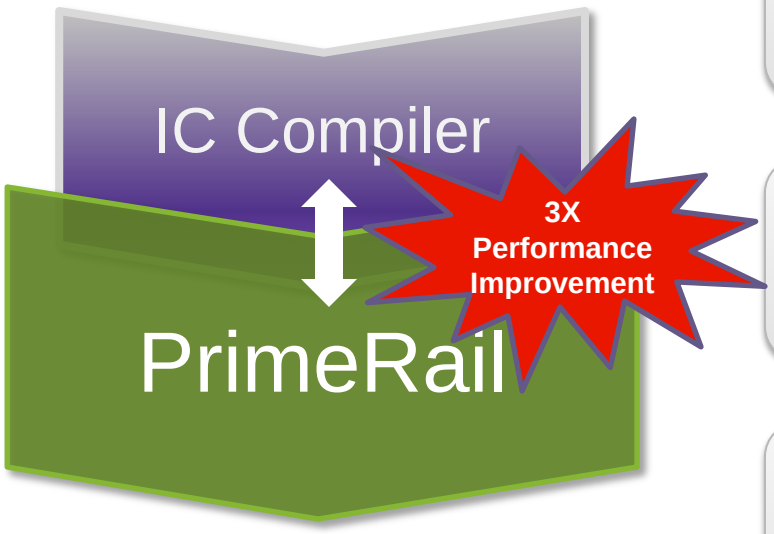
- At 28nm and Below Manufacturability Needs to be Treated as a Key Design Consideration
  - No room in schedule for late-stage surprises
- Ideal Approach: Pattern-based Verification With IC Compiler And IC Validator
  - IC Validator Pattern Matching delivers >10K faster performance without sacrificing accuracy
  - In-design technology offers high automated repair rates without impacting design timing
- Qualified and Deployed to Production by Leading Foundries and IDMs

Physical Closure

- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

In-Design with PrimeRail

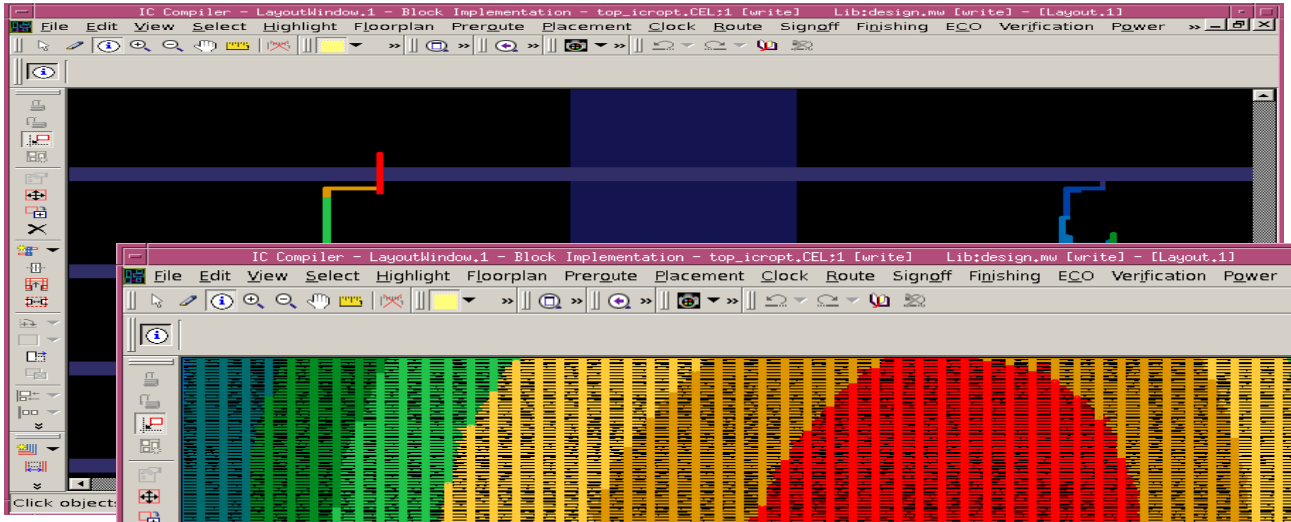
Improving Productivity for Physical Designers



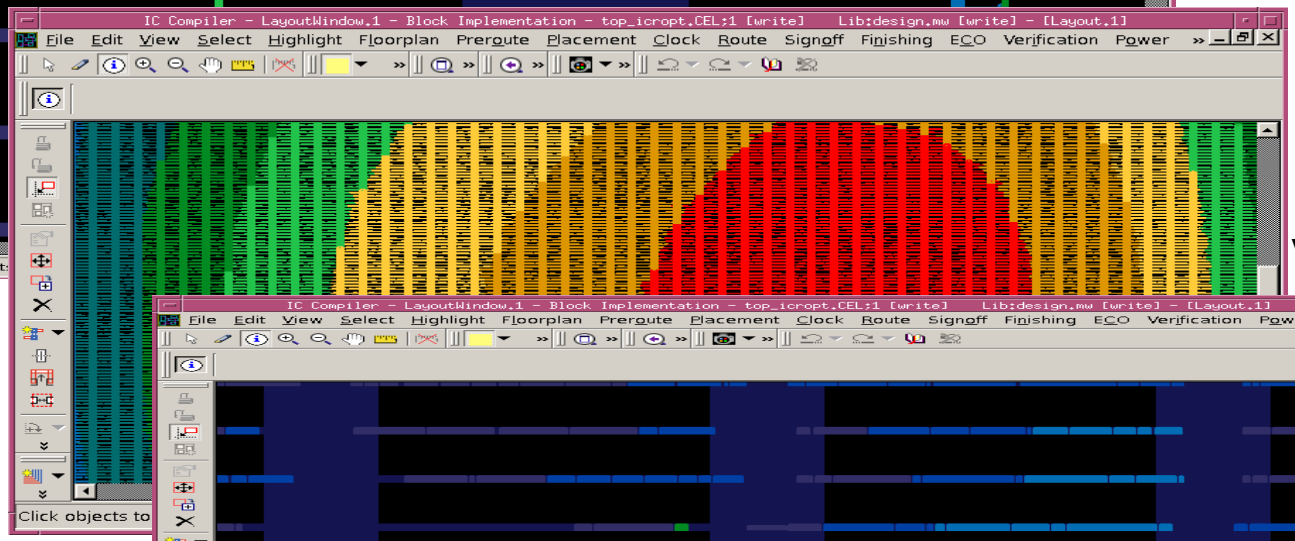
- High Performance and Scalability
- Intelligent Rail Analysis using Smart Filters
- Seamless Integration with IC Compiler

Advanced Rail Analysis Within P&R

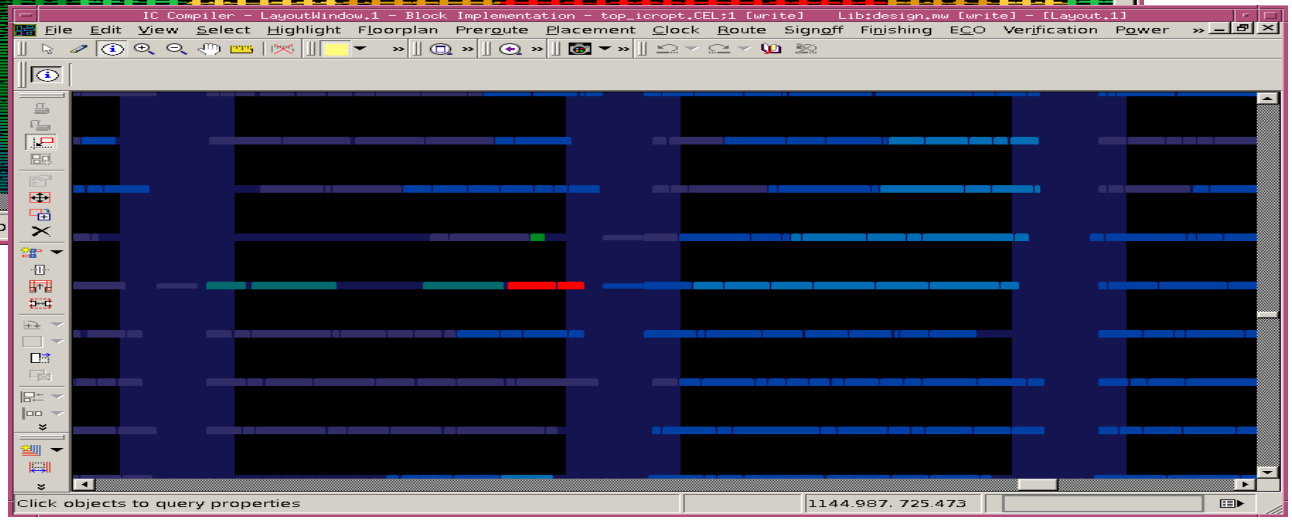
Viewing Rail Analysis Results In IC Compiler



Resistivity Map



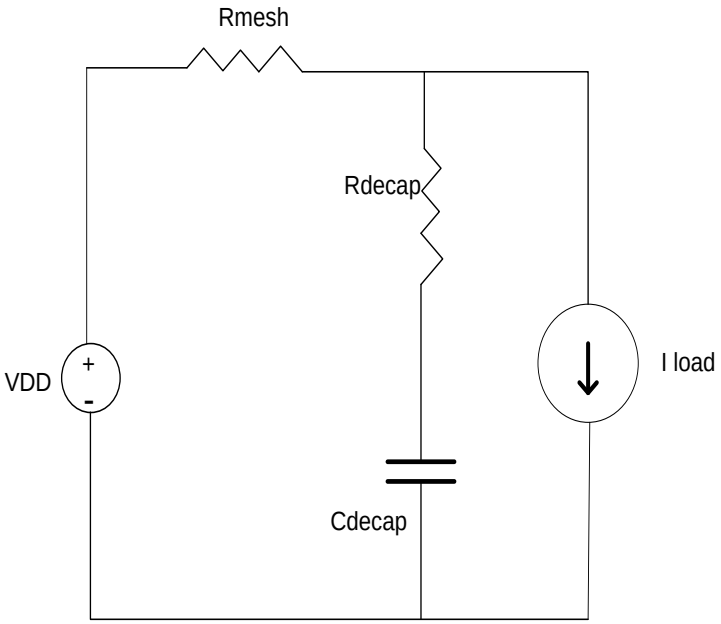
Voltage Drop Map

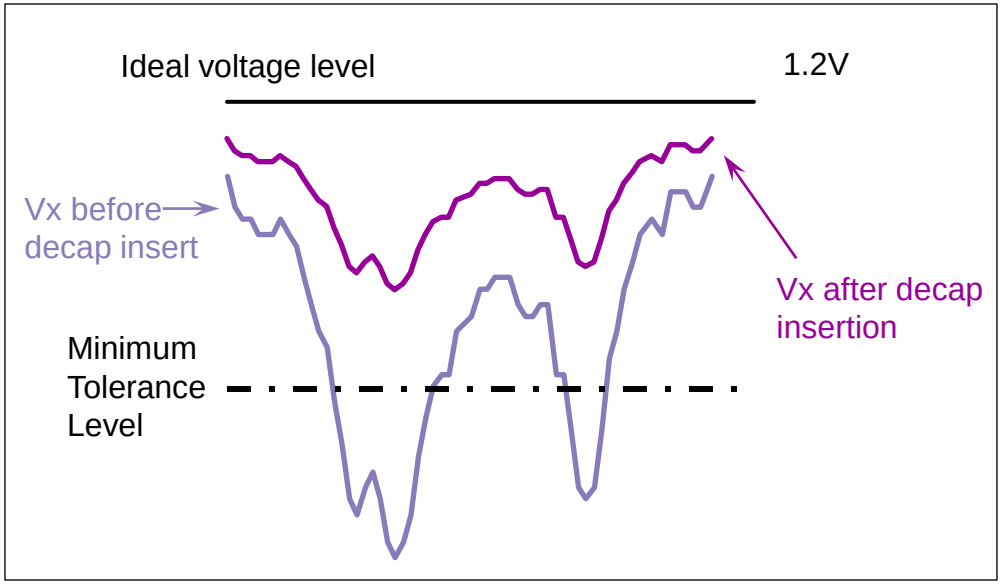


Electromigration Map

In-Design Decap Insertion

- Decaps are on-chip decoupling capacitors that are attached to the P/G network to decrease noise effects.

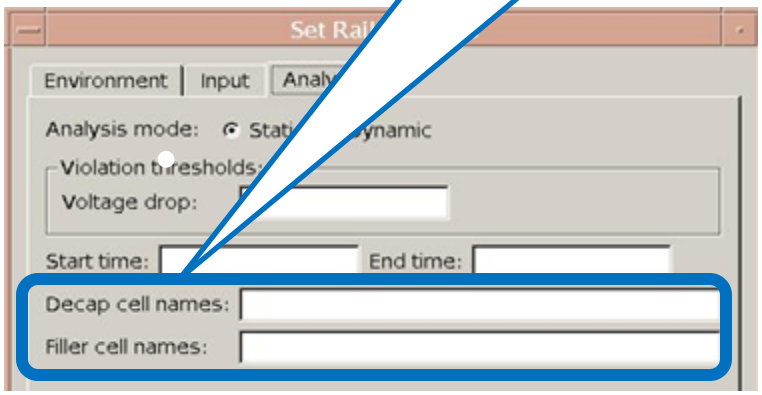




Decap Analysis setup in ICC

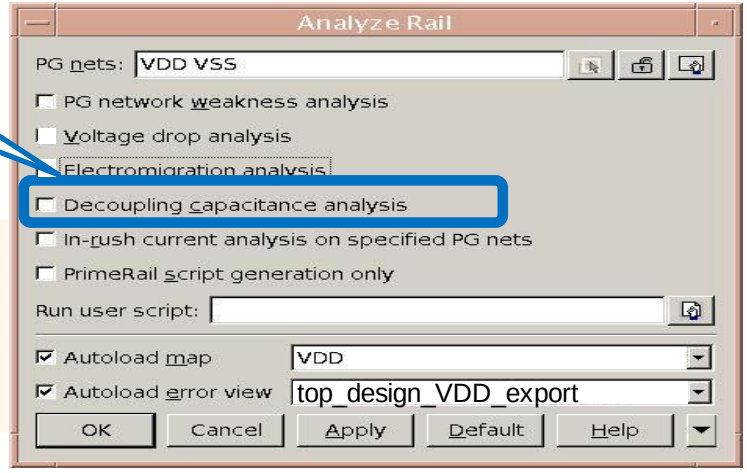
- "set_rail_options" needs to specify 'filler' & 'decap' masters
- Pre-requisite: Filler cells needed to be inserted in advance

Specify decap and filler cell masters



Decoupling capacitance analysis selection needed

- "analyze_rail –decap {VDD}" for decap analysis




```
DECAP: total cap value from decap cells 2.38363e-08
*****
DECAP: After inserting all filler cells to design, the voltage drop

DECAP: reduction is 69.2844% (8.13614 mV -> 2.49906 mV), which is greater than user target 10%

DECAP: Will reduce the amount of cap to meet target
DECAP: decap insertion analysis succeeded
```

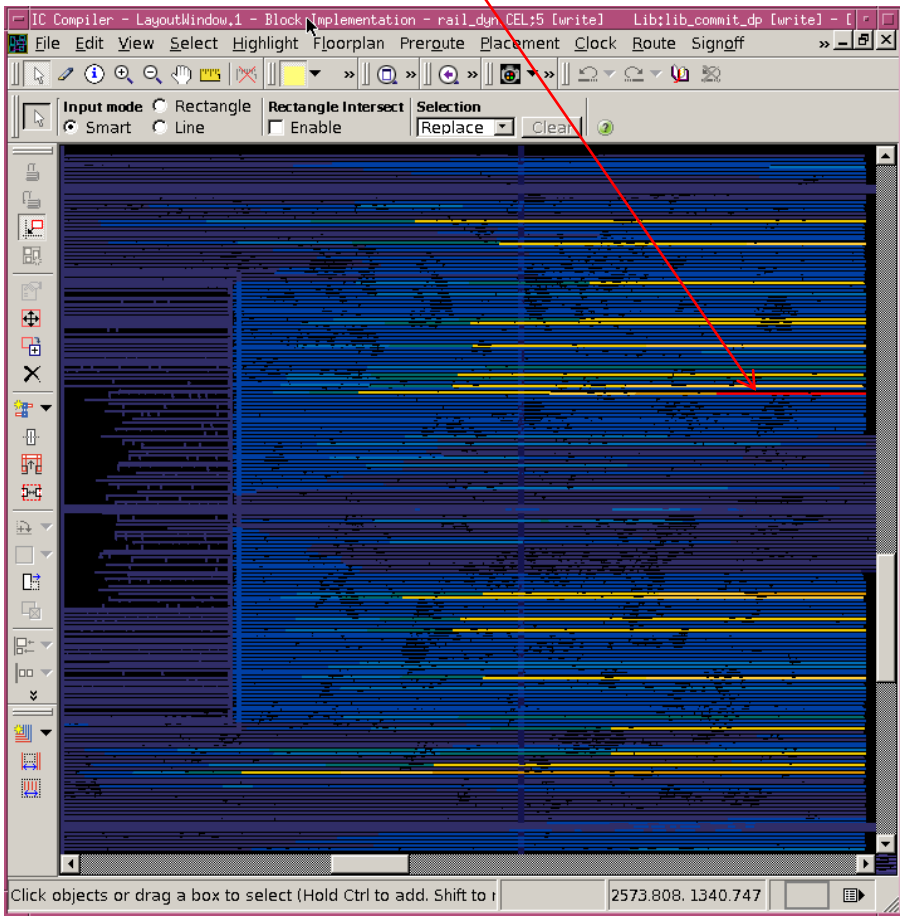
- For decap insertion, output file needs to be sourced in ICC:

```
icc_shell> source  ./pr_raill/rail_VDD_decap
```

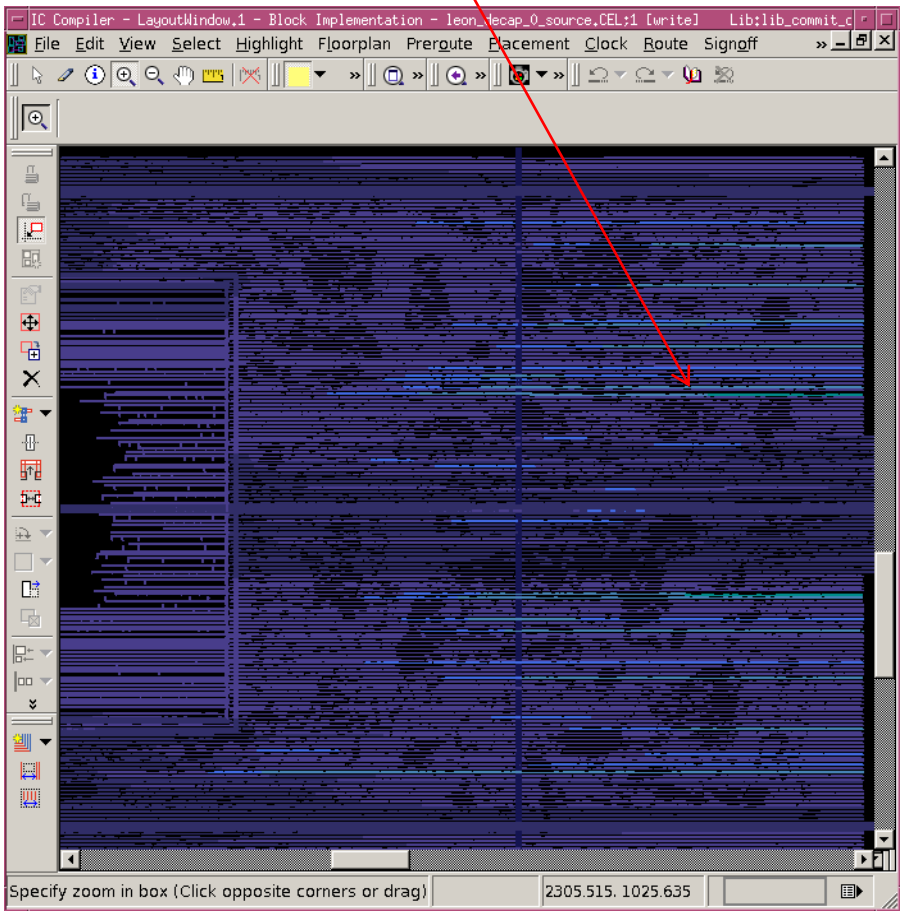

SYNOPSYS 25 Years

Decap Analysis and Insertion Results

Before decap insertion:
peak drop = 195.2 mV



After decap insertion:
peak drop = 95.0 mV




```
icc_shell> set_rail_options -filler_lib_cells {FILL1 FILL2} -decap_lib_cells {DECAP1 DECAP2}
icc_shell> analyze_rail -decap VDD
icc_shell> source  ./pr_raill/rail_VDD_decap
```

2012.06 - Evolution of In-Design Rail

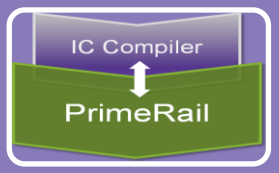
Continuous Improvement for Designer Productivity

2013+

Improved dynamic performance

- 3x dynamic performance improvement
- Robust static auto-fixing flow
- Voltage regulator support

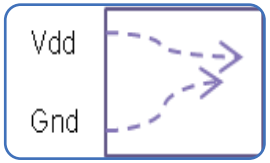
2012



Improved static performance

- **Analyze up to 10 Million instances in a hour**
- **Robust rail integrity checks**
- **Seamless integration with IC Compiler**

2011



IC Compiler driven rail analysis

- Auto P/G pin propagation
- Allow user defined supply voltage from IC Compiler
- Improve message

2010

Continuous improvements to increase productivity

- P/G and extraction reuse
- Data consistency check
- Enable rail analysis early in the design phase

2009

Direct invocation and report

- Push button flow in IC Compiler
- Direct access to rail analysis results

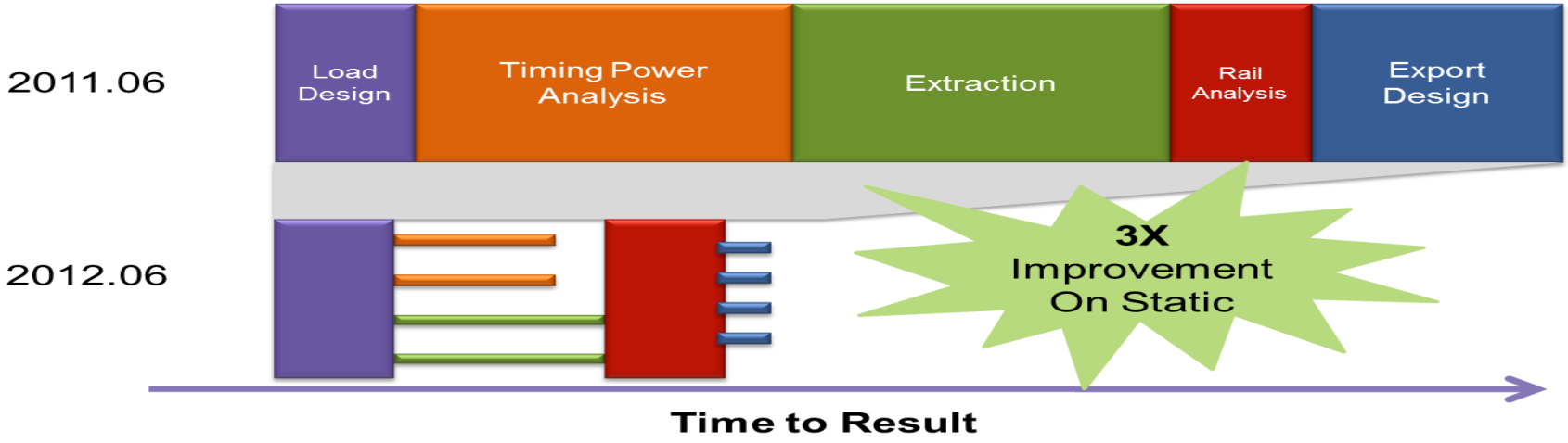
In-Design Static Rail Highlights

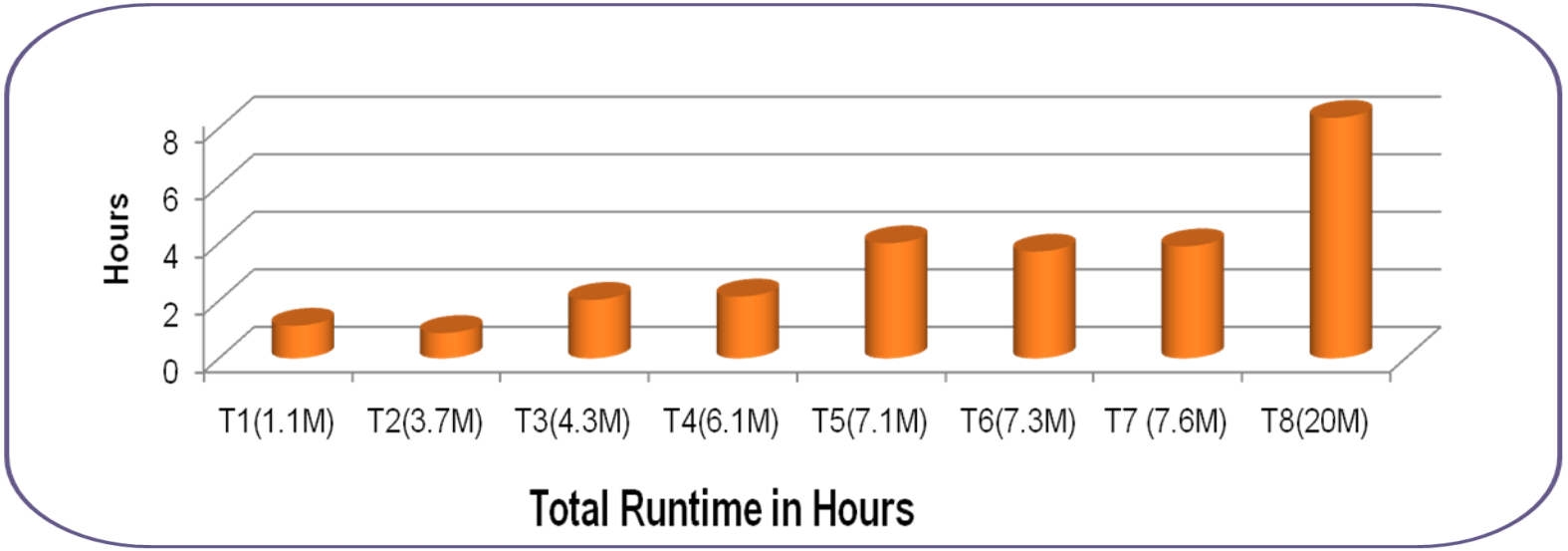
3X Faster, More Intelligent Rail Checks, Easier to Use

	2011.06	New in 2012.06
Performance	• 40M instances in ~12 hrs • 2M instances in ~50min	3X faster; multicore **Up to 10M instances in ~60min**
Integrity Checks	• Unconnected cell/pin • Floating net • Floating geometry • Missing via	• Floating geometry grouping • Missing via pre-filtering • Stacked via checks • Flexible user control on overlapping characteristics
Core Analysis	• Static IR Drop analysis • Static EM analysis	• More MT-CMOS control • Flexible TAP point creation
Usability	• "Push-button" ICC menu items • Fixing guidance • Enhanced script-ability	• SNPS Tcl for all commands, flexible script-ability • Hot swap In-Design (tcl/scheme)
Graphical Debugging	Error browsing and debug via ICC compiler GUI	Standalone ICC-like viewer w/ advanced error browsing**

Competitive Performance

*Multi-core Parallelization Speeds Up Runtime By **3X***





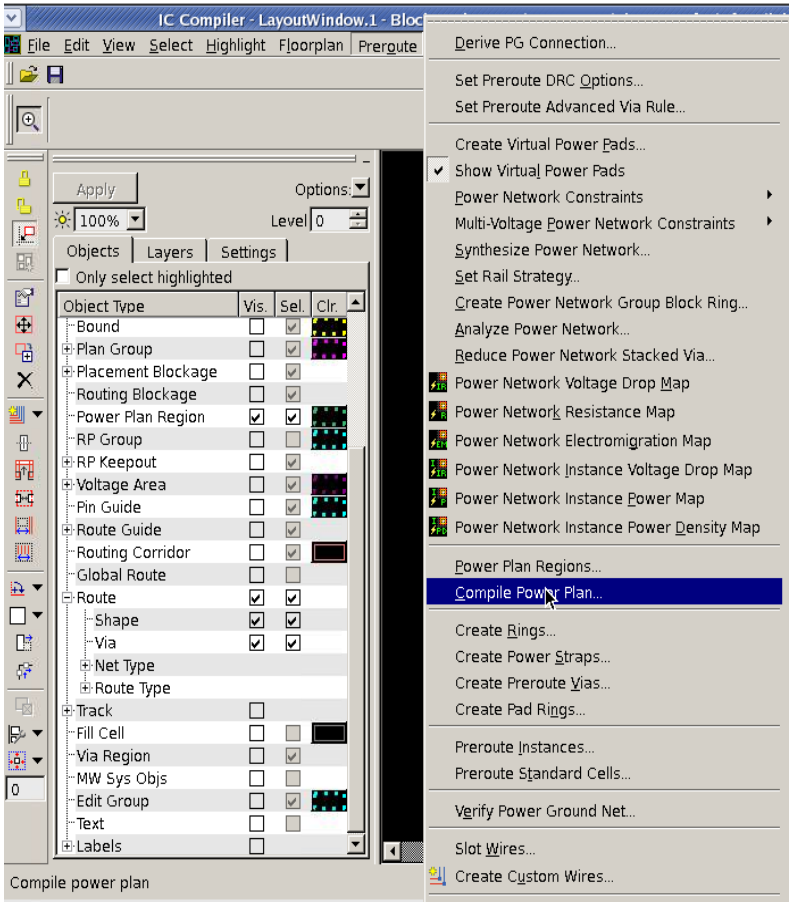
The In-Design TPNS and Rail Integrity Pairing

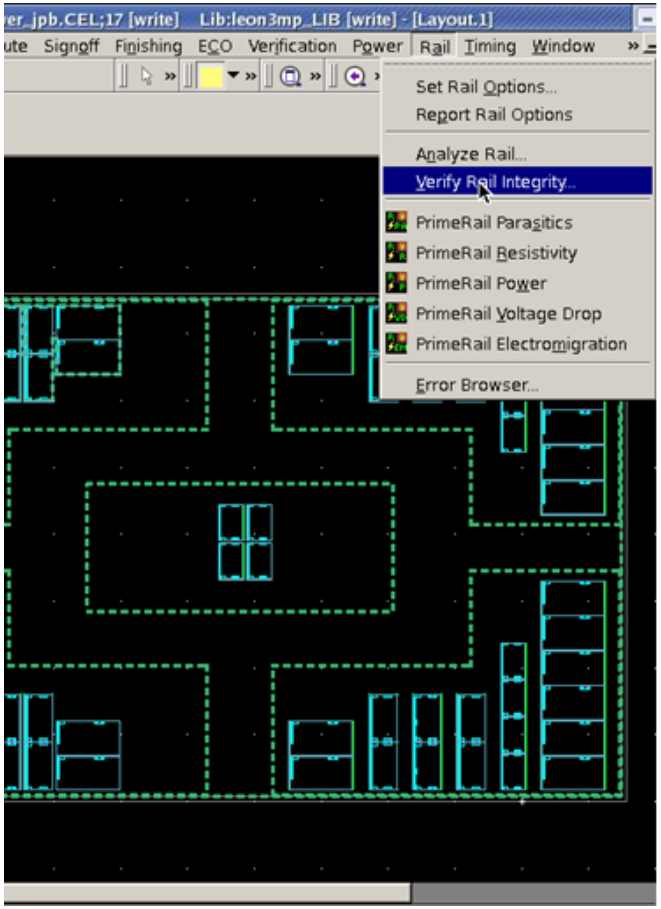
Rail Integrity Checking is flexible during the build process

***TPNS-**"Compile Power Plan"*

***Rail Integrity-**"Verify Rail Integrity"*

New

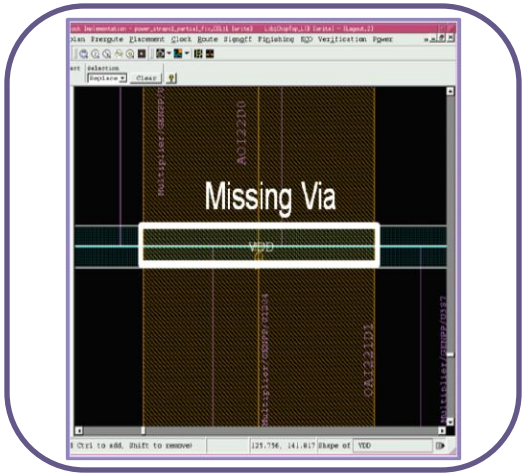


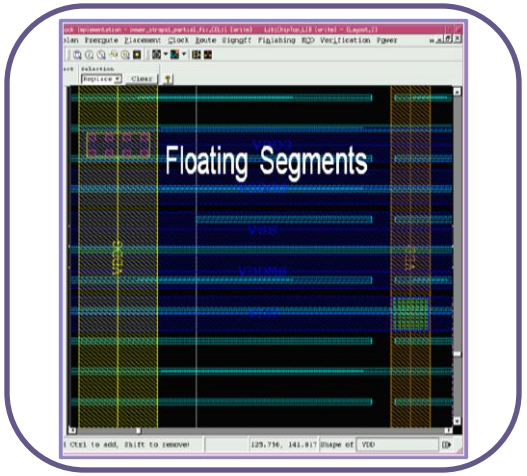


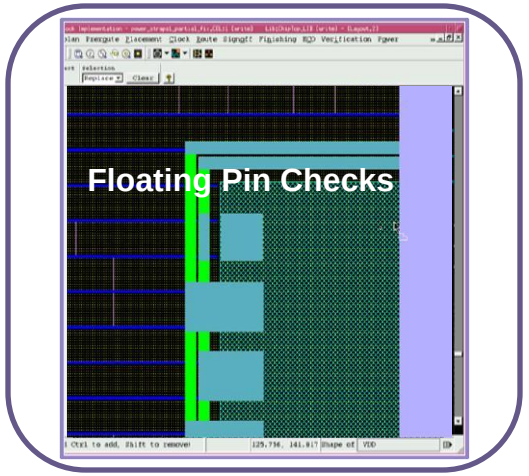
SYNOPSYS 25 Years

Integrity Checking Refresher

Basic diagnostics for clean IR/EM Analysis

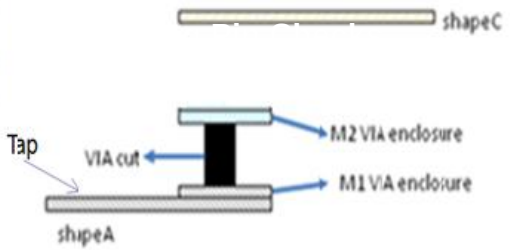






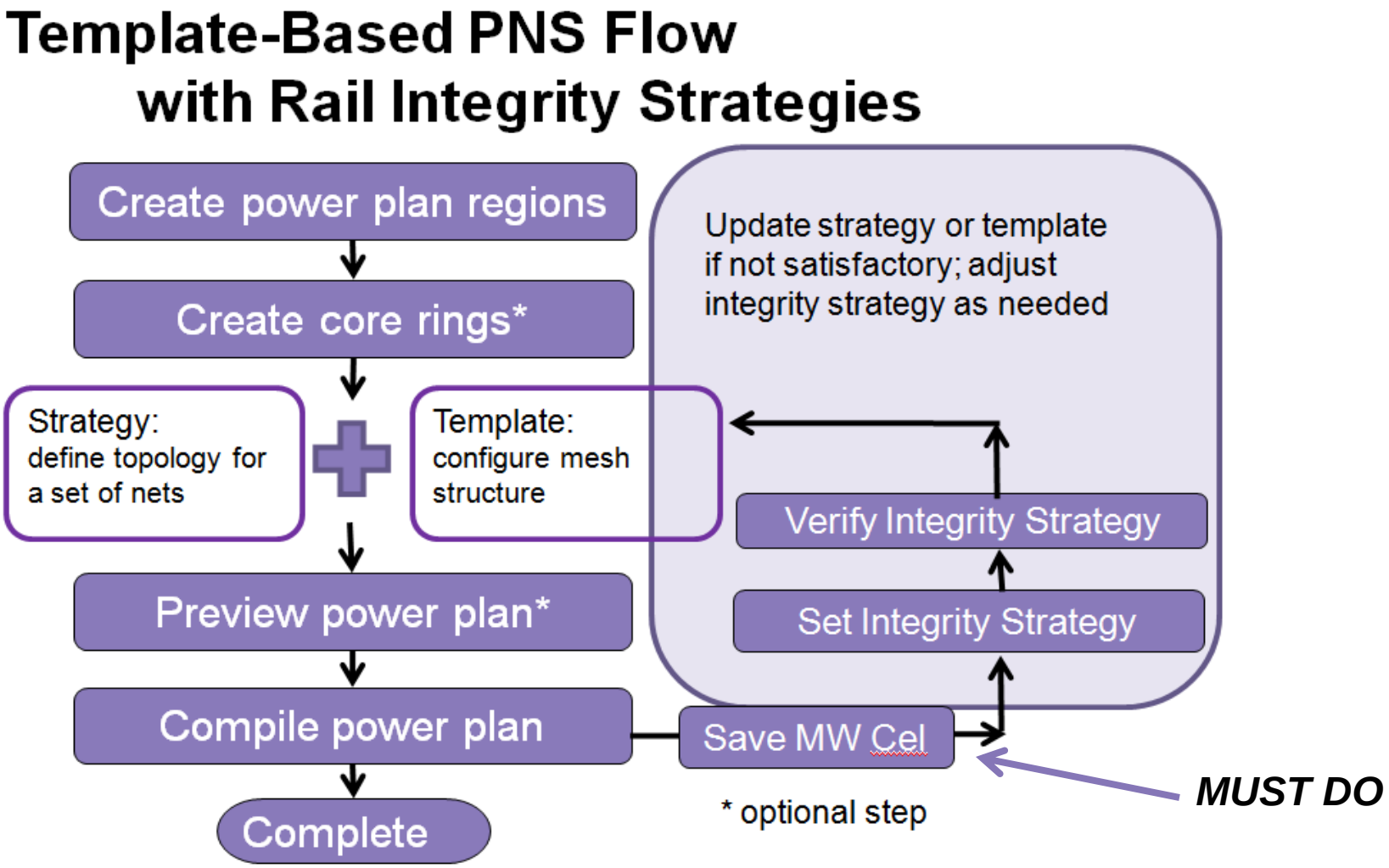
- Ease-of-debugging through Smart Filters
  - Better and more seamless information
  - Enhanced rules for object targeting
  - Polygon processing controls
  - Pinpoint area filtering by coordinates/objects

Dangling Vias



Rail Integrity, A Complement to Template PNS

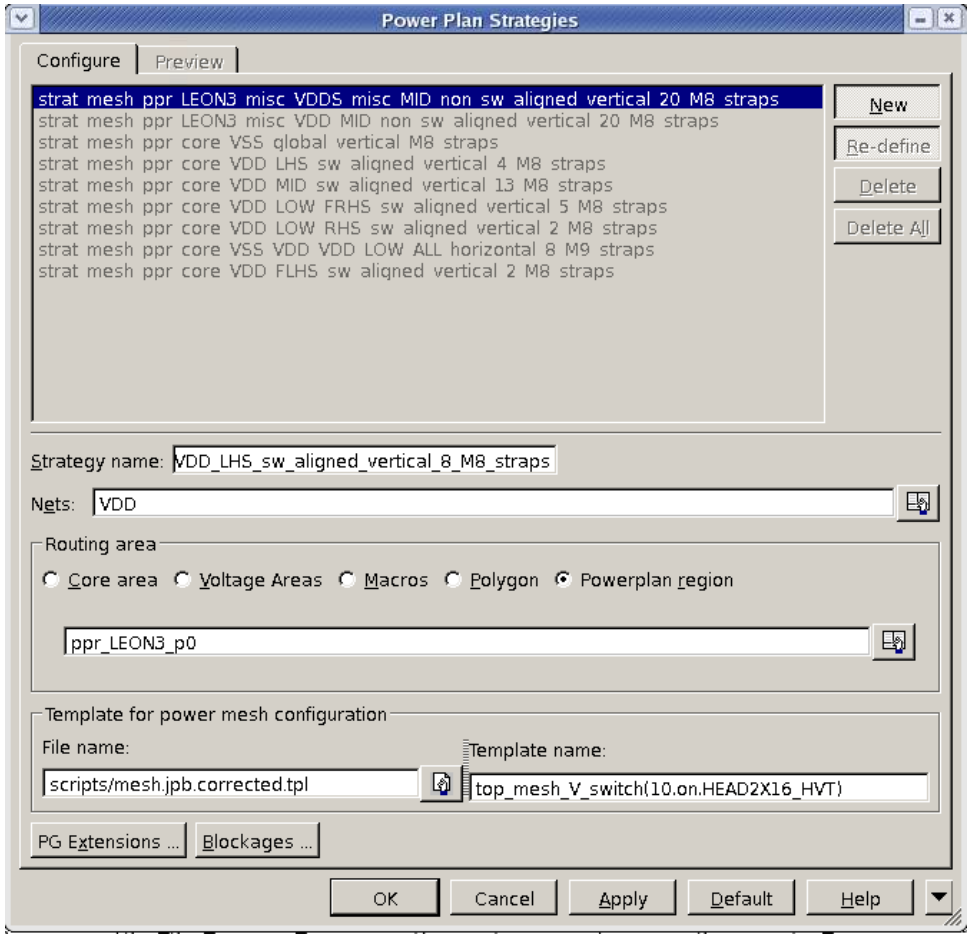
Augments the Implementation Flow



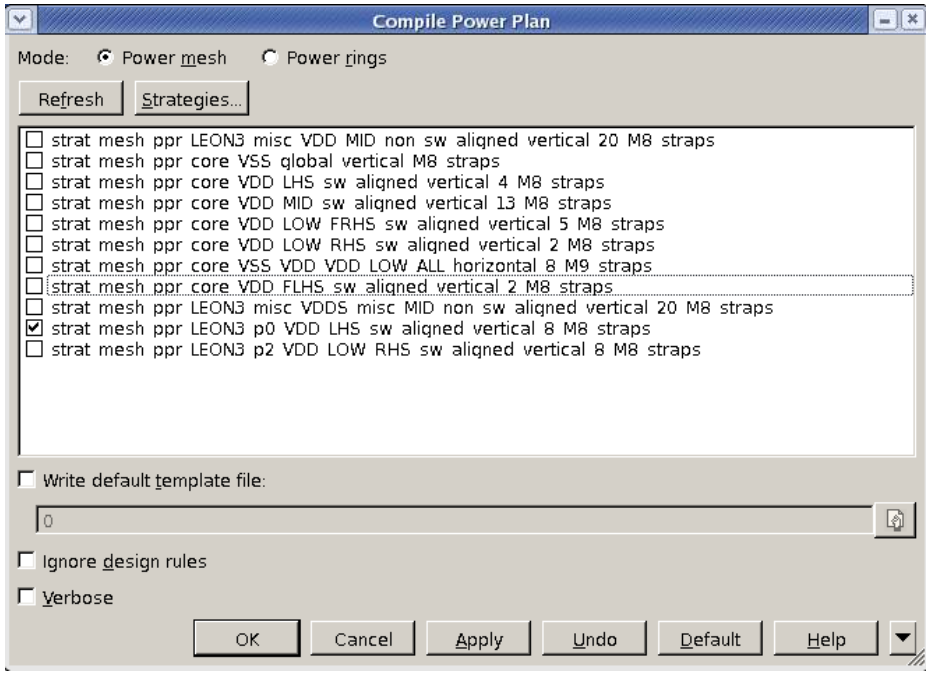
Power Plan Strategy Management

(Re)Define Custom TPNS Strategies

Power Plan Strategy Editor



Compile Power Plan (Strategy Invocation)

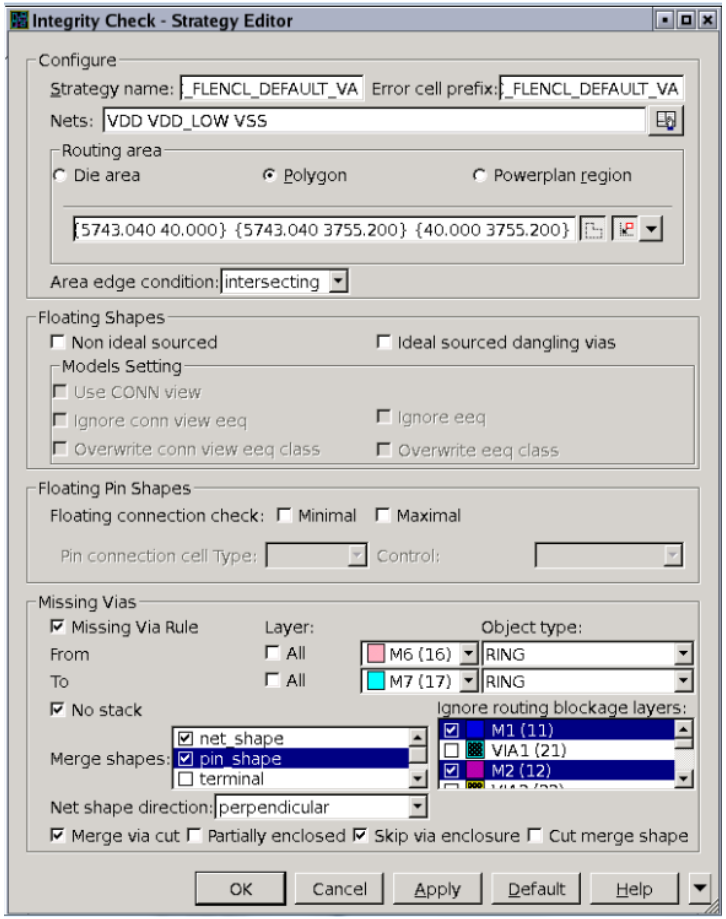


Rail Integrity Strategy Management

(Re)define/Invoke Custom Rail Strategies

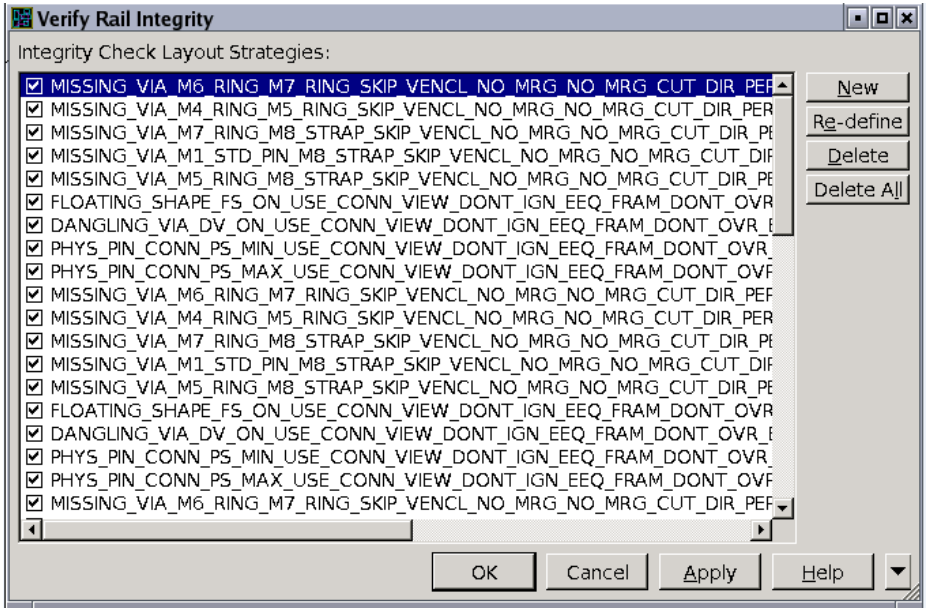
Rail Integrity Check Strategy Editor

New



Verify Rail Integrity

New

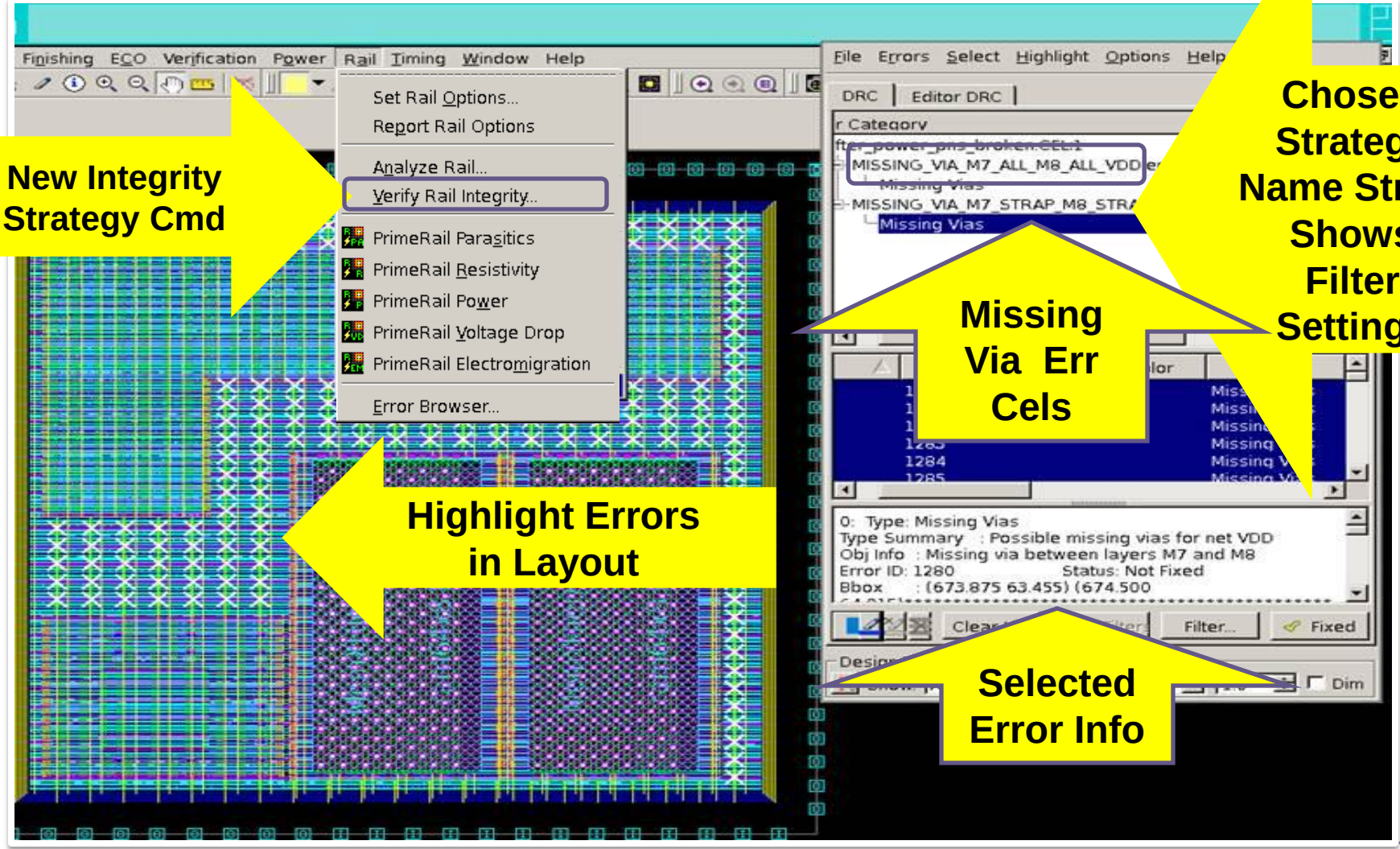


Note: Subject to Change
(SP1/SP2 additions expected)

SYNOPSYS® 25 Years

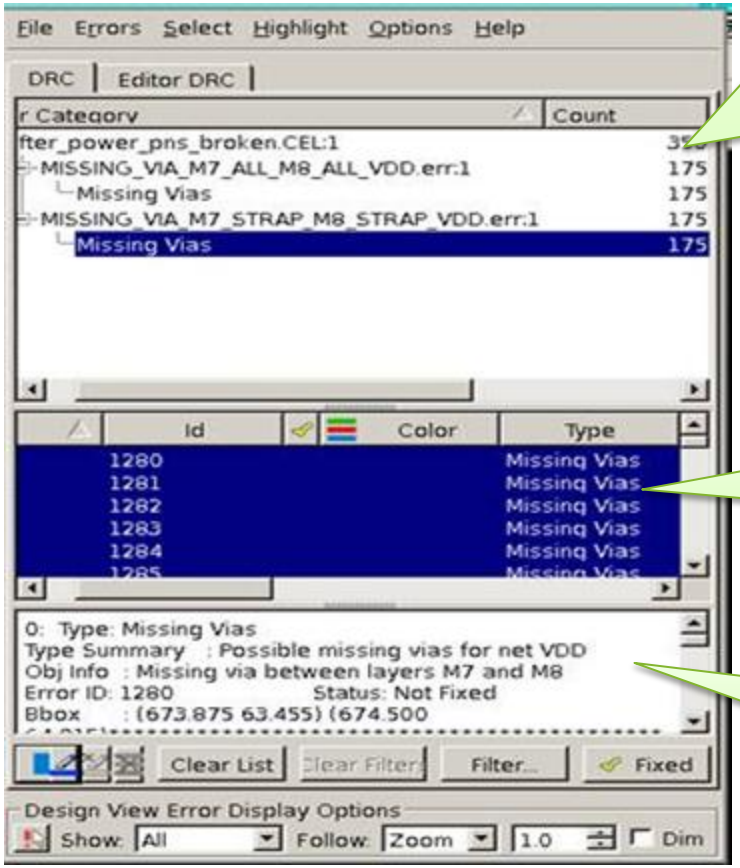
Strategy generates Multiple Error Cels

A separate error cel per supply net



Find & Fix Issues

Early Detection and Fixing Guidance



Query by error types

- Voltage Drop violations
- Electromigration (EM) violations
- Missing vias
- Floating segments
- Dangling Vias
- Floating pin shapes

Browse individual error

Display error details and fixing guidance

Summary

- 3x Multi-Core Static IR/EM Analysis Speedup
  - Up to 10 Millions instances in an hour
- Multiple Core Rail Integrity Engine Improvements
  - New Strategy Based Integrity UI is well aligned to IC Compiler's TPNS flow
  - Improved flexibility and versatility of filters for customization of Integrity results
- Improved error information enables more user customization of resulting collection information
  - For user ECO changes to the database
  - For user diagnostic work in IC Compiler

Physical Closure

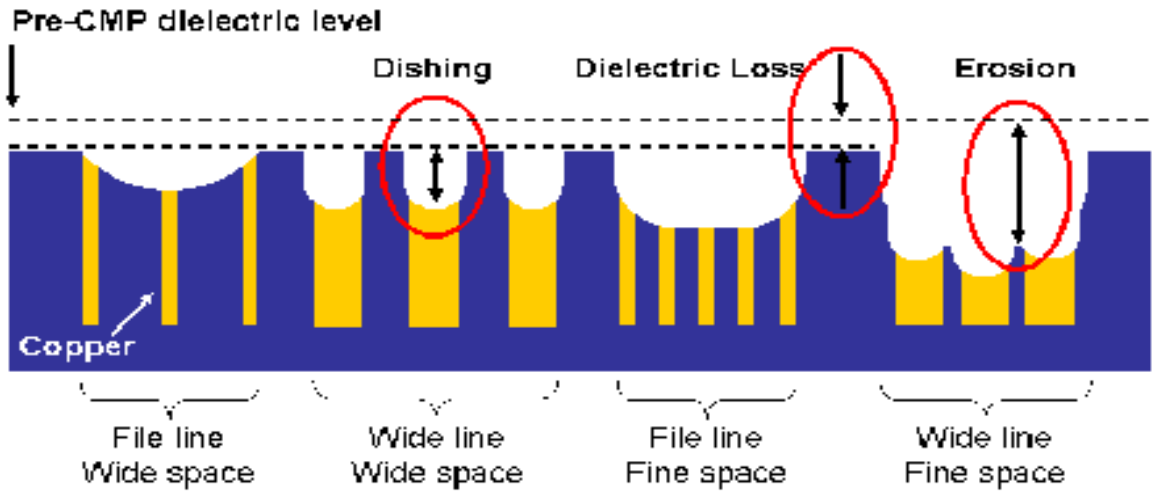
- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

In-Design Metal Fill

- Why is metal fill required?
- Different types of fill
- Traditional Vs In-Design fill flow
- Signoff_metal_fill features

Why is metal fill required?

- Reduce dielectric thickness variation
- Increase planarity
- Uniform Patten density



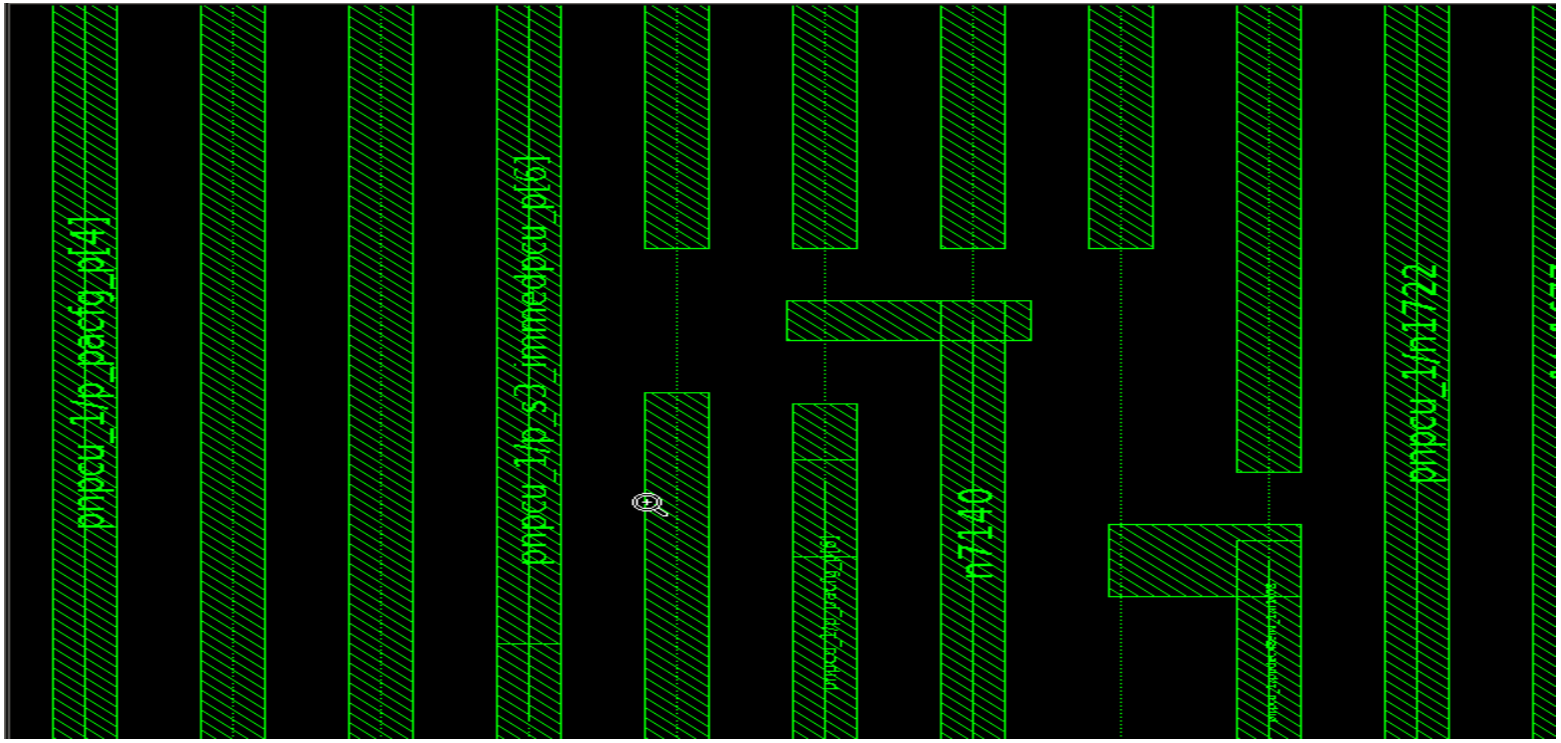
Different types of fills

- Pattern based:
  - Track based floating fills
  - Track based Tied fills
  - Staggered fills
- Design based
  - Density driven fill (Fill to Target)
  - Timing preserve fill

Types of fills

Track based floating fills

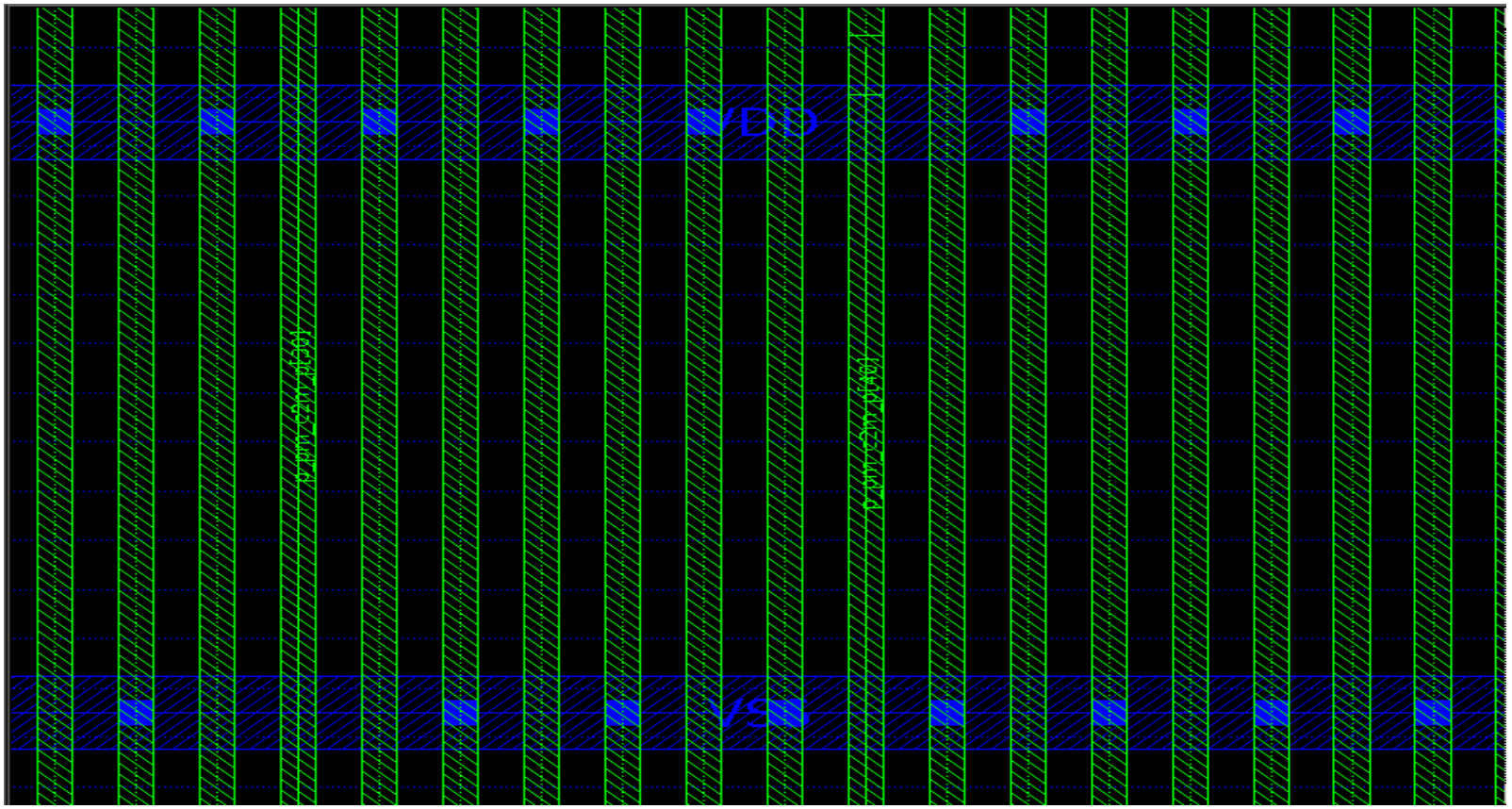
- Floating fills are added on tracks
- ICC metal track info used
- Only Milkyway input format supported



Types of fills

Track based tied fills

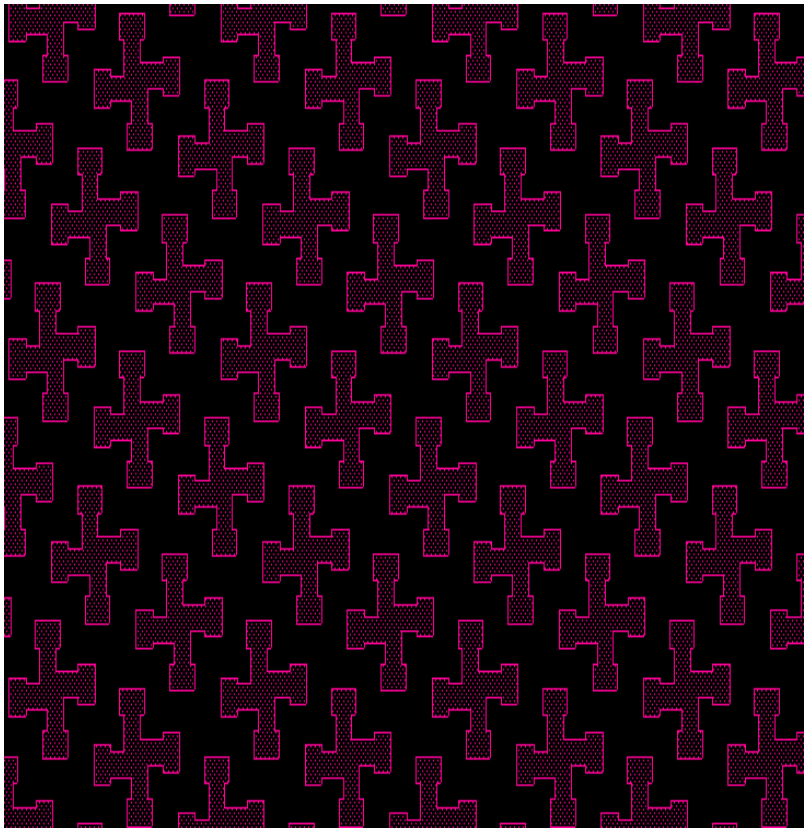
- Fills are tied to specific nets
- Mostly on PG nets for IR drop

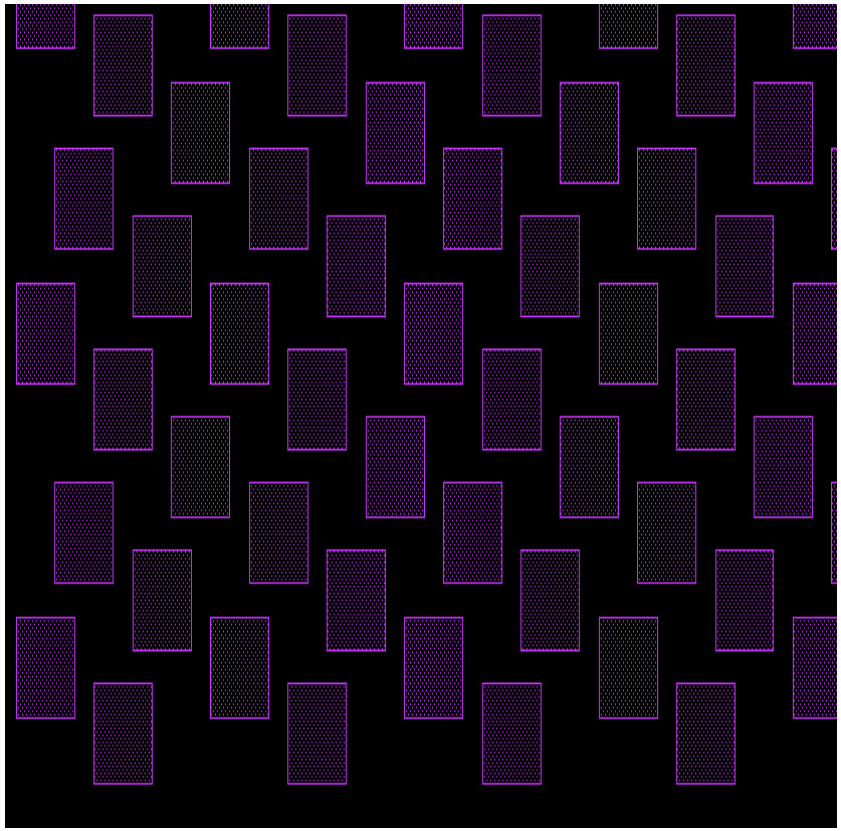


Types of fills

Staggered fills

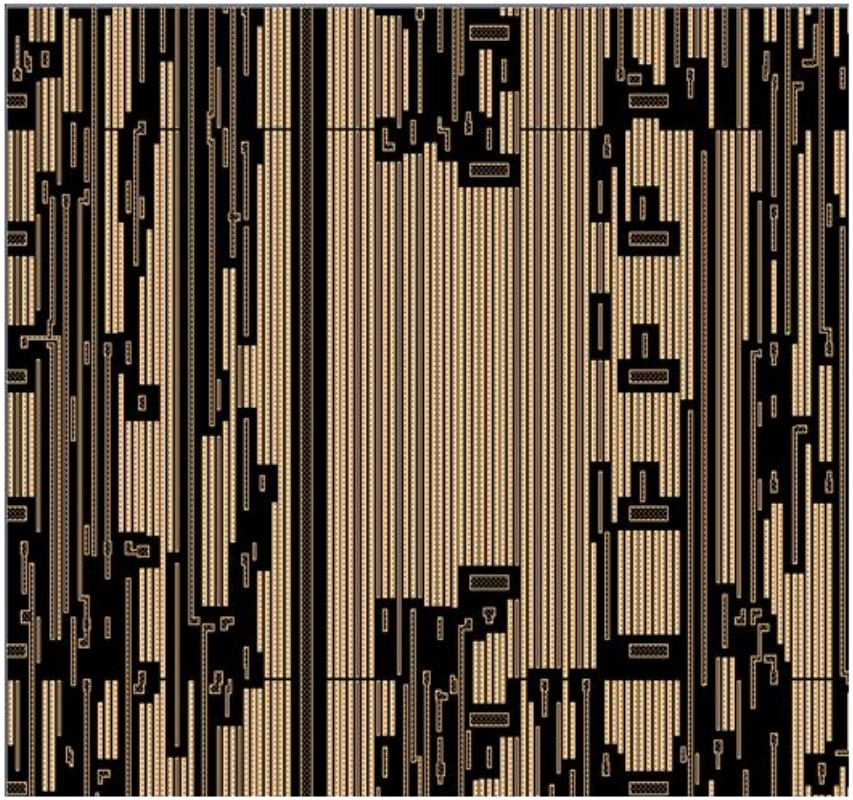
- Technology rule specific fills
- Not on tracks



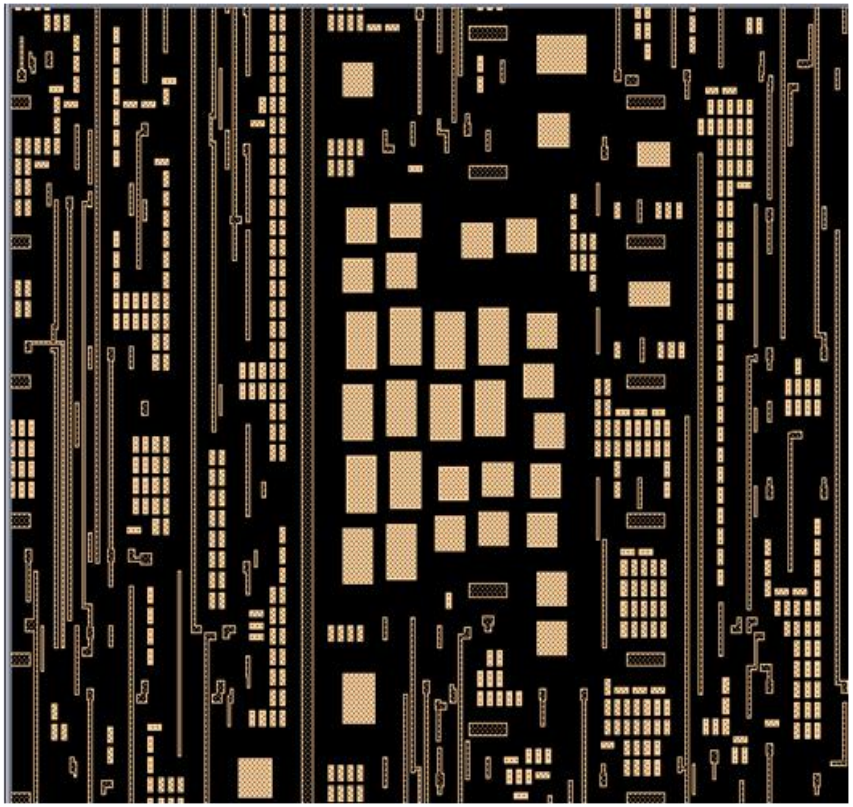


Fill comparison

Track based fill Vs staggered fill



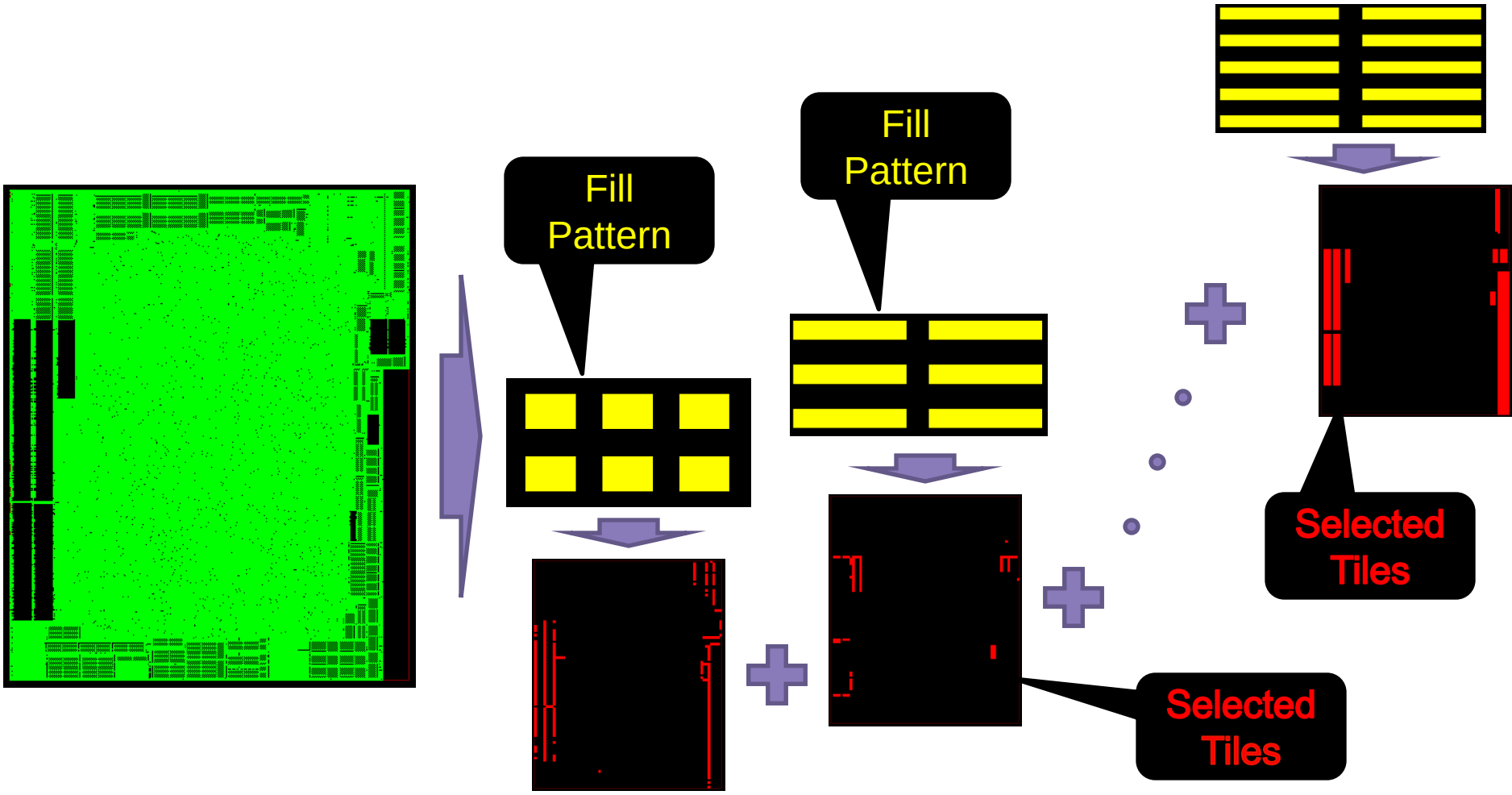
Router-Inserted Metal Fill



Signoff-Foundry Inserted

Density driven (Fill-to-Target)

Single-Pass Flow Eliminates Iterations Due to Density



Density driven (Fill-to-Target)

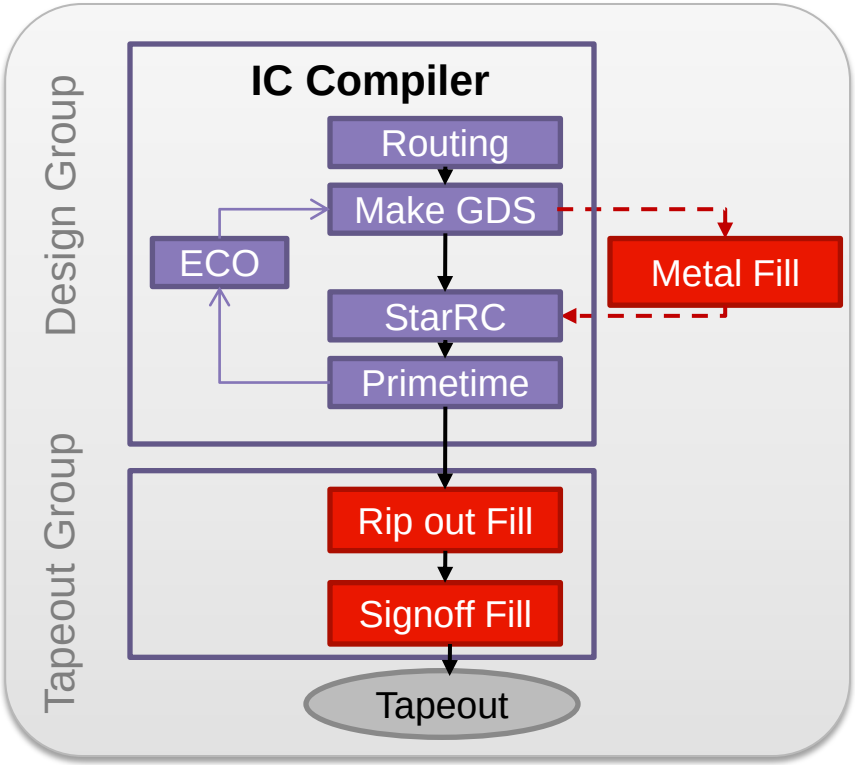
Key Advantages

- Able to support variety of fill target modes
- May address different fill target goals within different part of the same metal layer
- Meeting density and perimeter targets at the same time

Traditional Fill Flow

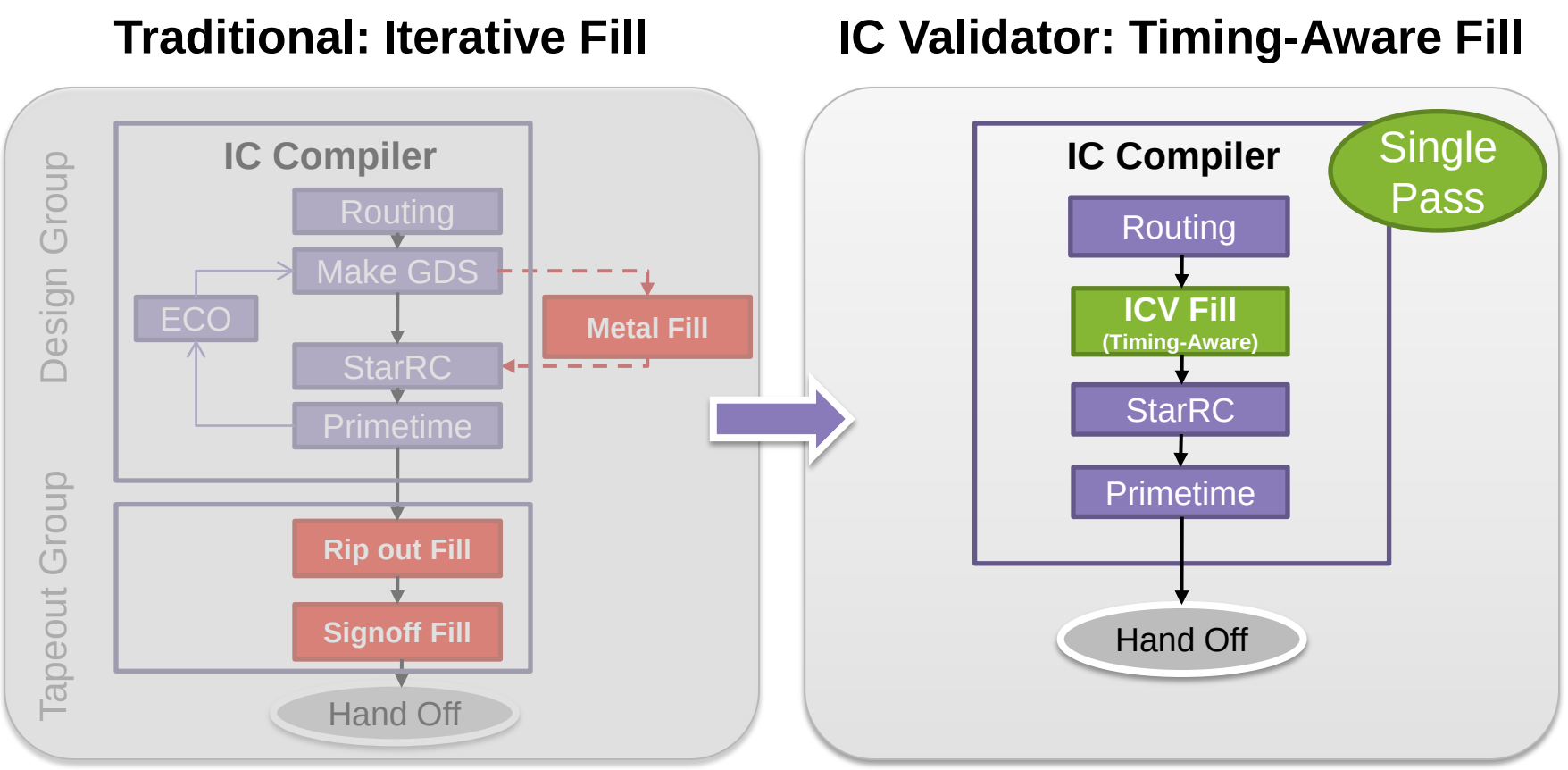
Fill Effect on Design Convergence Delaying Closure

- Impact on Timing and Signal Integrity
  - Long Iterations with Signoff Analysis
  - Lengthy and non-Convergent ECOs
- May Not Meet Physical Signoff Requirements
  - Density, Gradient, etc.



In-Design Fill Flow

Efficiently Balances Density and Timing Requirements



Running In-Design PV Fill in ICC

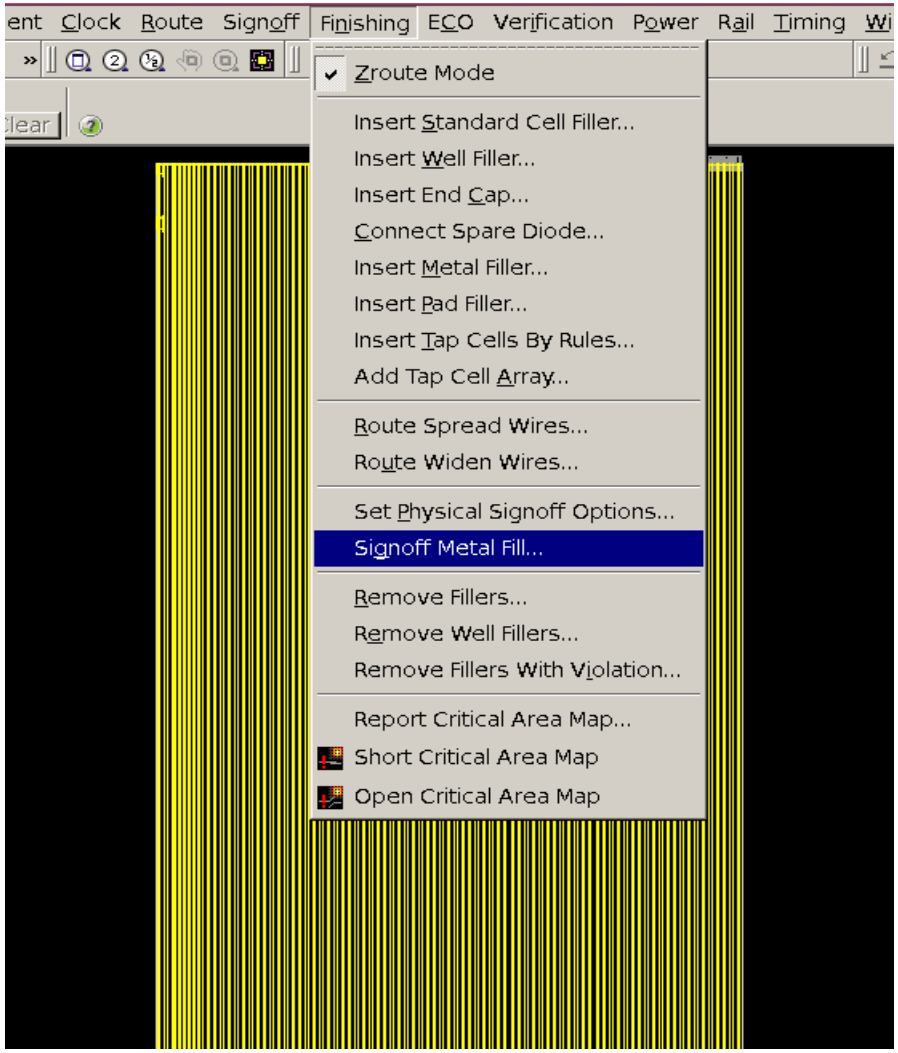
ICC 2010.12-SP3

- GUI

  Finishing->

  - *Set Physical Signoff Options*
  - *Signoff Metal Fill*

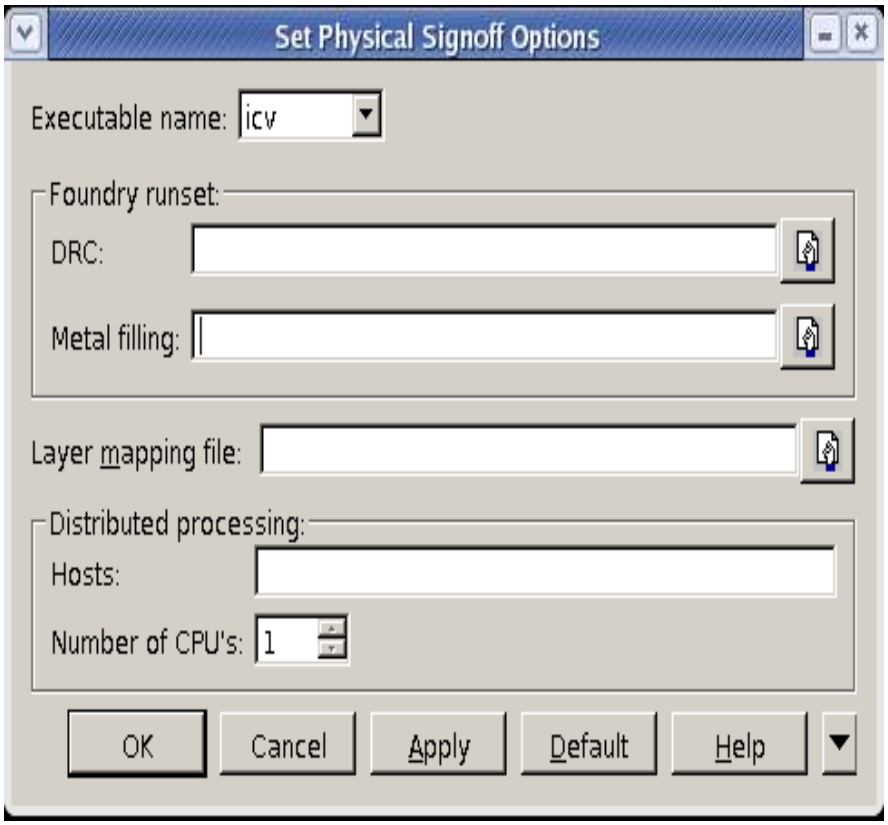
- Tcl commands:
  - *set_physical_signoff_options*
  - *report_physical_signoff_options*
  - *signoff_metal_fill*



Running In-Design PV Fill in ICC

Setup

- Tcl commands:
  set_physical_signoff_options
  report_physical_signoff_options
- Foundry fill runset (Required)
- Layer mapping file (Optional)
  - if MW layers have different layer number than the runset
  - Only Milkyway mapping file format supported
- Distributed Processing (Optional)
  - Will be retired in future release
  - Use set_host_options command in ICC
    - LSF, GRID, localhost

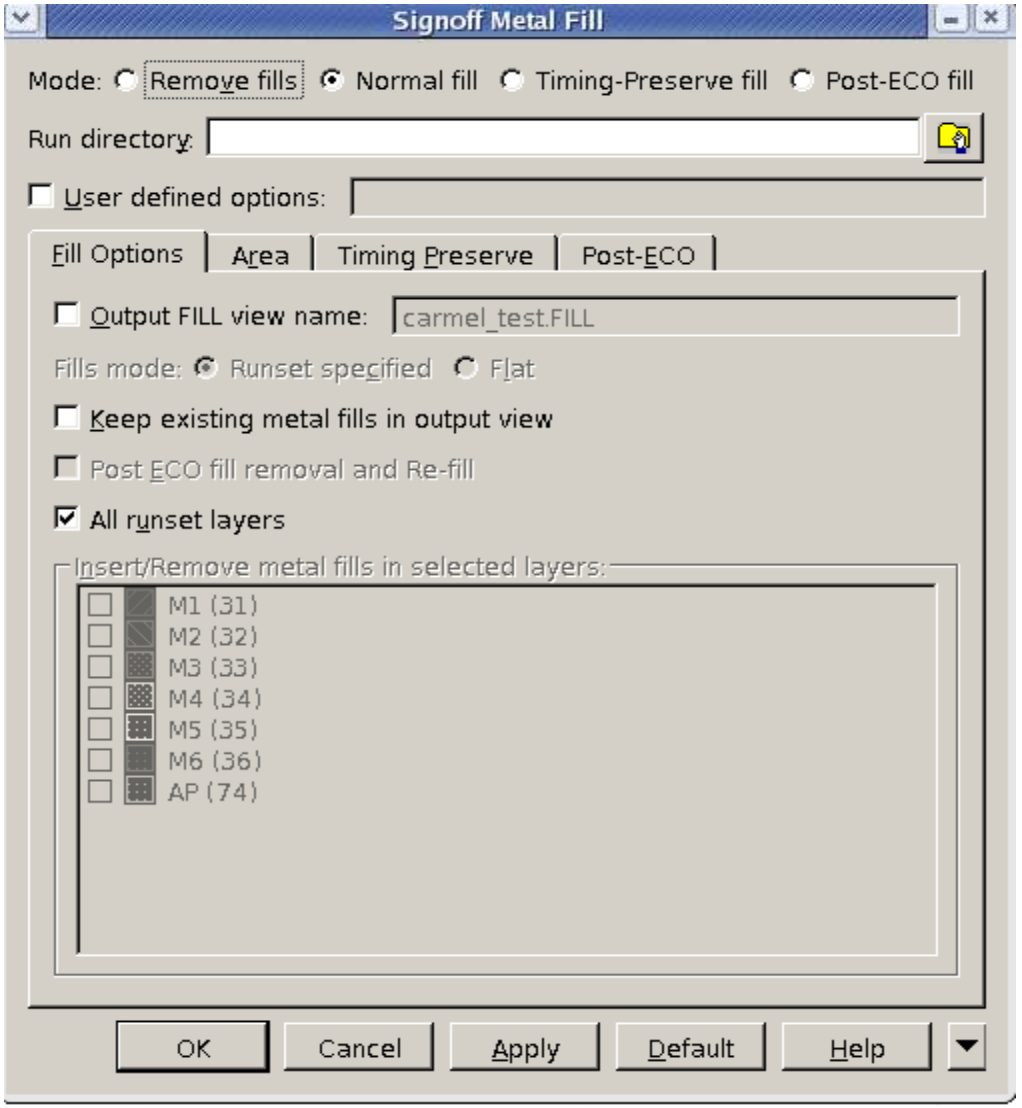


Running In-Design PV Fill in ICC

Execution

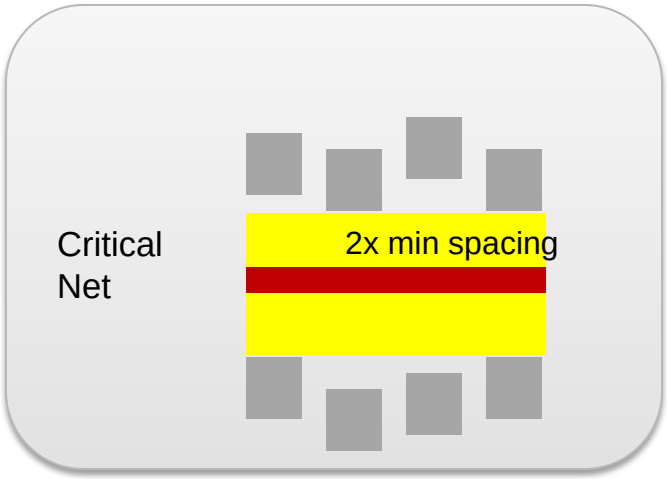
- Tcl command:
  signoff_metal_fill

- Features:
  - Area based
  - Timing Aware
  - Layer based
  - ECO based

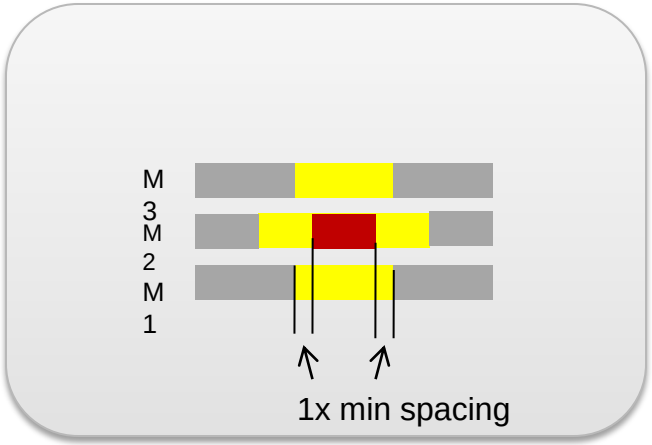


Running In-Design PV Fill in ICC

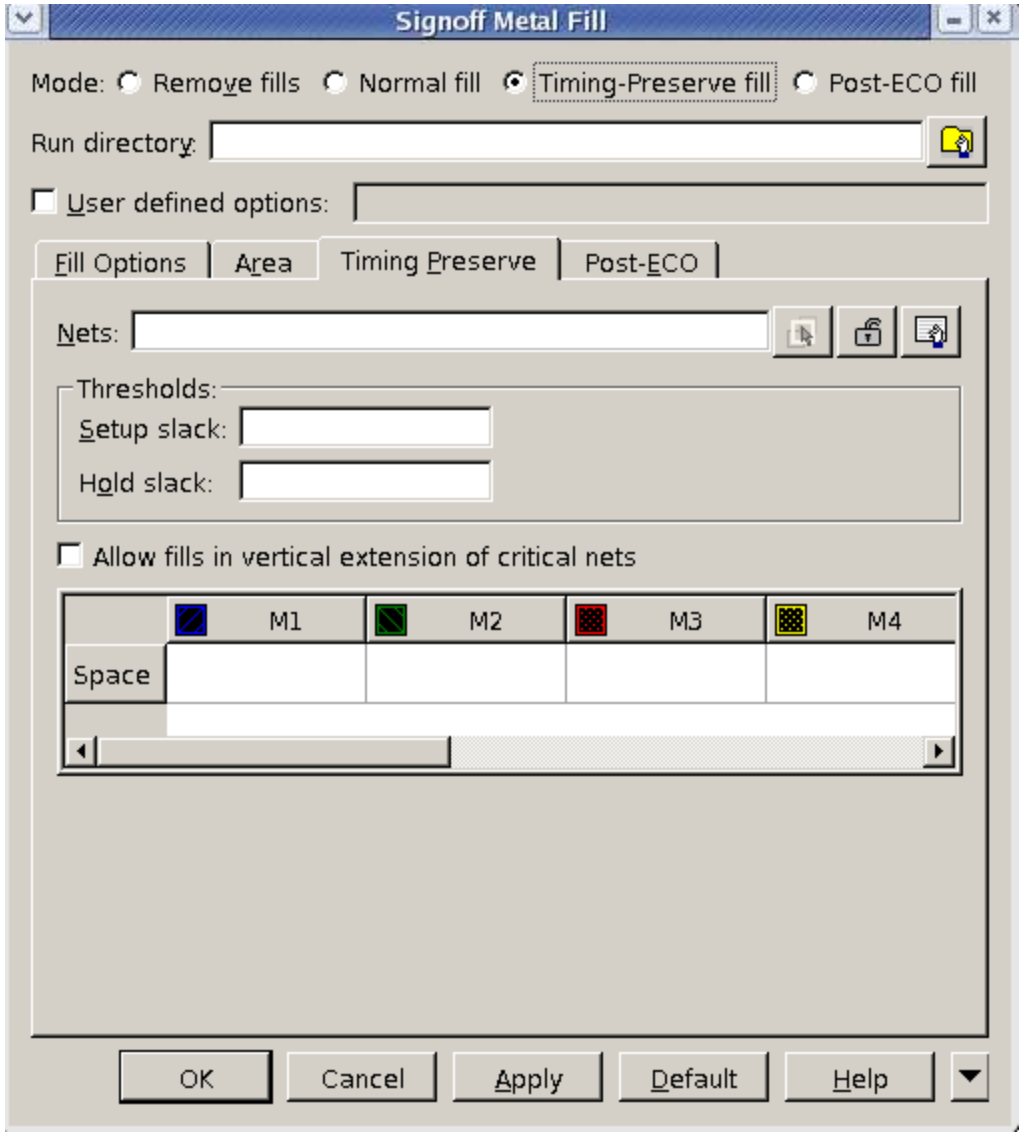
Timing preserve fill



Same layer fill



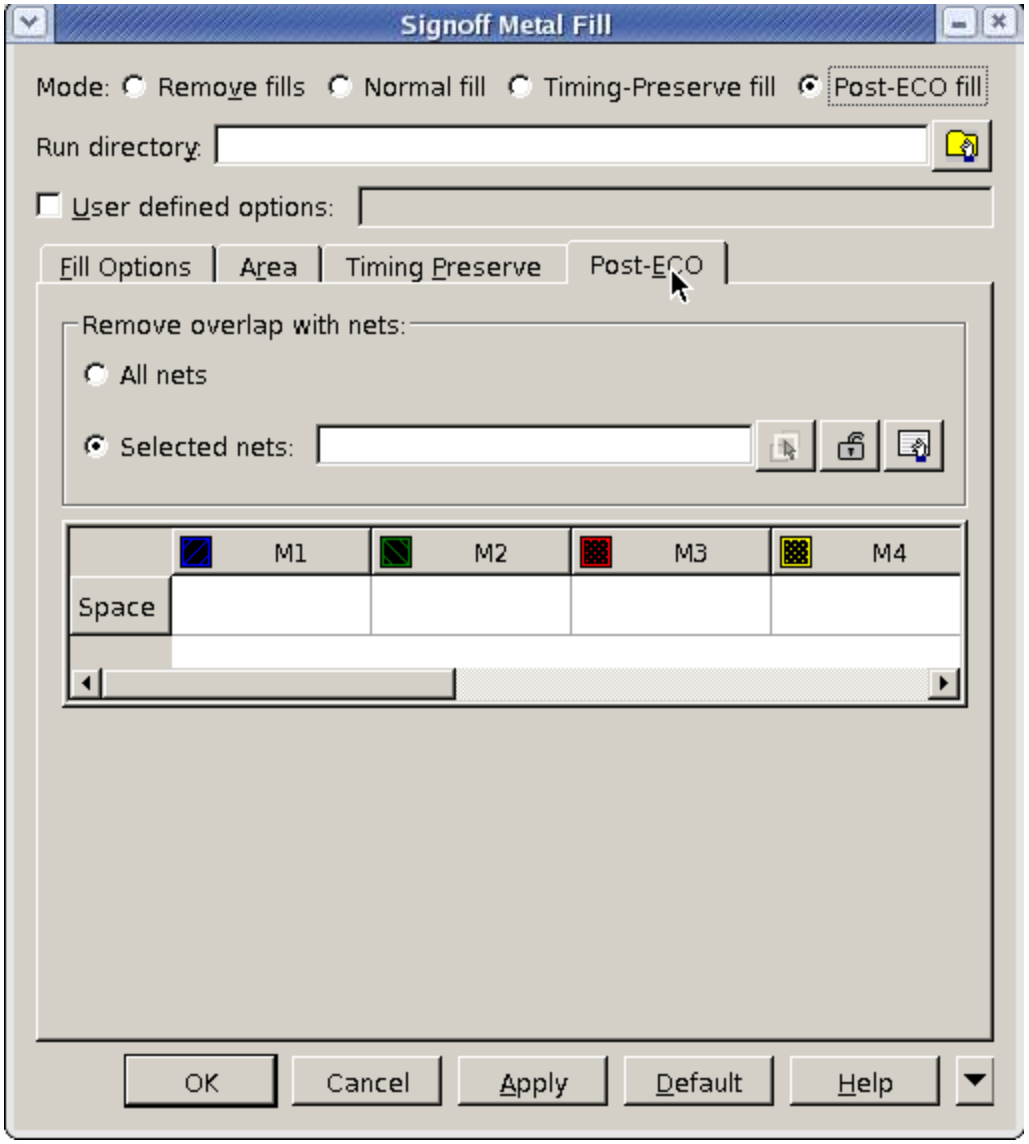
Adjacent layer fill



Running In-Design PV Fill in ICC

Post ECO fill

- Net based fill overlap removal
- All nets/selected nets
- Per layer spacing control
- Do not use ICC's trim_fill_eco command



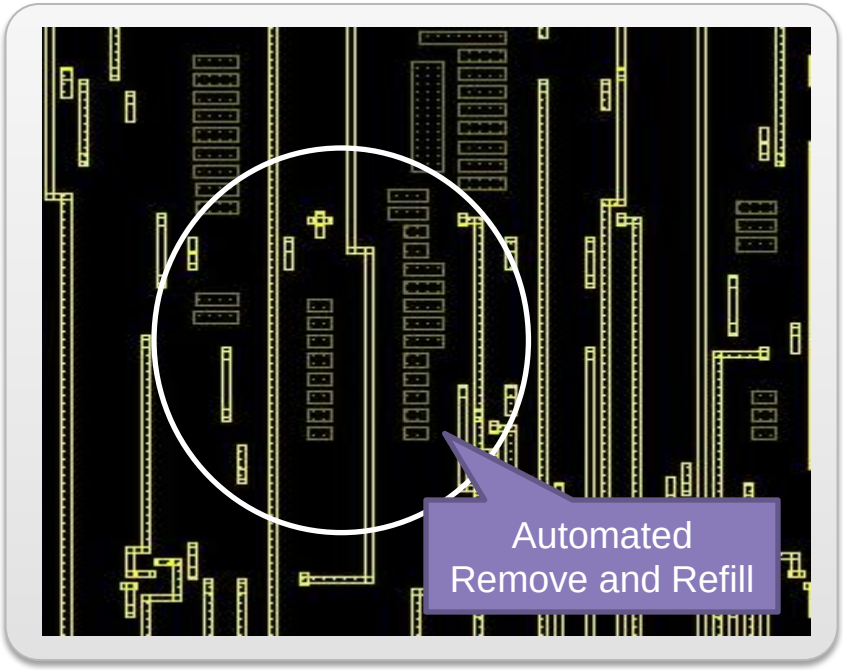
Results: Automated Fill ECO

Leading IDM, 40nm, 100um² Block

Post-ECO Fill Overlap

- ✗ Manual, suboptimal density
- ✗ >1 day turnaround

In-Design PV – Auto Fill ECO



- ✓ Automated, signoff quality
- ✓ 44 mins

2011.09 enhancements wrt fill

Less disk space

- Fill structures uses GDSREF&GDSAREF mw objects
- ~2x improvement in fill db size
- Benefit using only in-design fill with MW
- Not backward compatible

2012.06 Enhancements

- Two new features are added for the IC Compiler-IC Validator metal fill flow
  - Timing- and density-aware metal fill insertion flow
    - Timing-aware metal fill insertion might result in additional density errors when large spacing is applied for removing fill near the critical nets. The enhanced density-aware flow checks and fixes the density errors.
  - Post-ECO DRC-based fill removal
    - This feature removes, based on the foundry's rules, the fill that overlaps with cells and routes after ECO changes.

Timing- and density-aware metal fill insertion flow

- Benefits
  - The enhanced flow both minimizes the impact of metal fill on timing and avoids additional density errors

Timing- and density-aware metal fill insertion flow

- Commands and UI
  - Timing- and density-aware metal fill flow

```
signoff_metal_fill
  -fix_density_errors true | false
```

    - Default: **`false`**
    - Use only with **`-timing_preserve_*`** options
    - No GUI support

Timing- and density-aware metal fill insertion flow

- Usage
  1. Specify the density rules in the technology file.

```
DensityRule     {
    layer               = "M2"
    windowSize          = 125
    minDensity          = 10
    maxDensity          = 85
}
```

     - Density rules can be found in the foundry's DRC runset

  2. Run timing- and density-aware metal fill insertion.

```
signoff_metal_fill -fix_density_errors true \
  -timing_preserve_setup_slack_threshold 0.1 \
  -space_to_critical_nets { m1 0.28 m2 0.28 … }
```

Post-ECO DRC-based fill removal

- Benefits
  - Enables fill removal after ECO changes based on foundry- and technology-specific fill spacing rules related to both cells and routes

Post-ECO DRC-based fill removal

- Commands and UI

```
set_physical_signoff_options
  -fill_removal_runset_include_file filename

signoff_metal_fill
 -remove_overlap_by_rules off | min_spacing |
   list | foundry_keyword
```

  - Default: **off**
  - No GUI support

Post-ECO DRC-based fill removal

- Usage
  1. Prepare the fill removal runset include file.

     You must provide the tool with a fill removal runset include file that specifies the foundry's fill spacing rule values. The details of how to prepare the file will be available in Solvnet article 035828.

  2. Specify the fill removal runset include file.

```
set_physical_signoff_options \
  -fill_removal_runset_include_file filename
```

Post-ECO DRC-based fill removal

- Usage (continued)

3. Run DRC-based fill removal
   - To list the supported foundry and nodes; currently **tsmc28** or **tsmc40**

```
signoff_metal_fill -remove_overlap_by_rules list
```

   - To perform fill removal using the specific foundry's rules, such as the TSMC 40-nm rules:

```
signoff_metal_fill -remove_overlap_by_rules tsmc40
```

   - To perform fill removal using twice the minimum spacing values in the technology file:

```
signoff_metal_fill \
   -remove_overlap_by_rules min_spacing
```

   - To perform fill removal using user-specified spacing values:

```
signoff_metal_fill \
   -remove_overlap_by_rules min_spacing \
   -space_to_critical_nets { m1 0.28 m2 0.28 … }
```

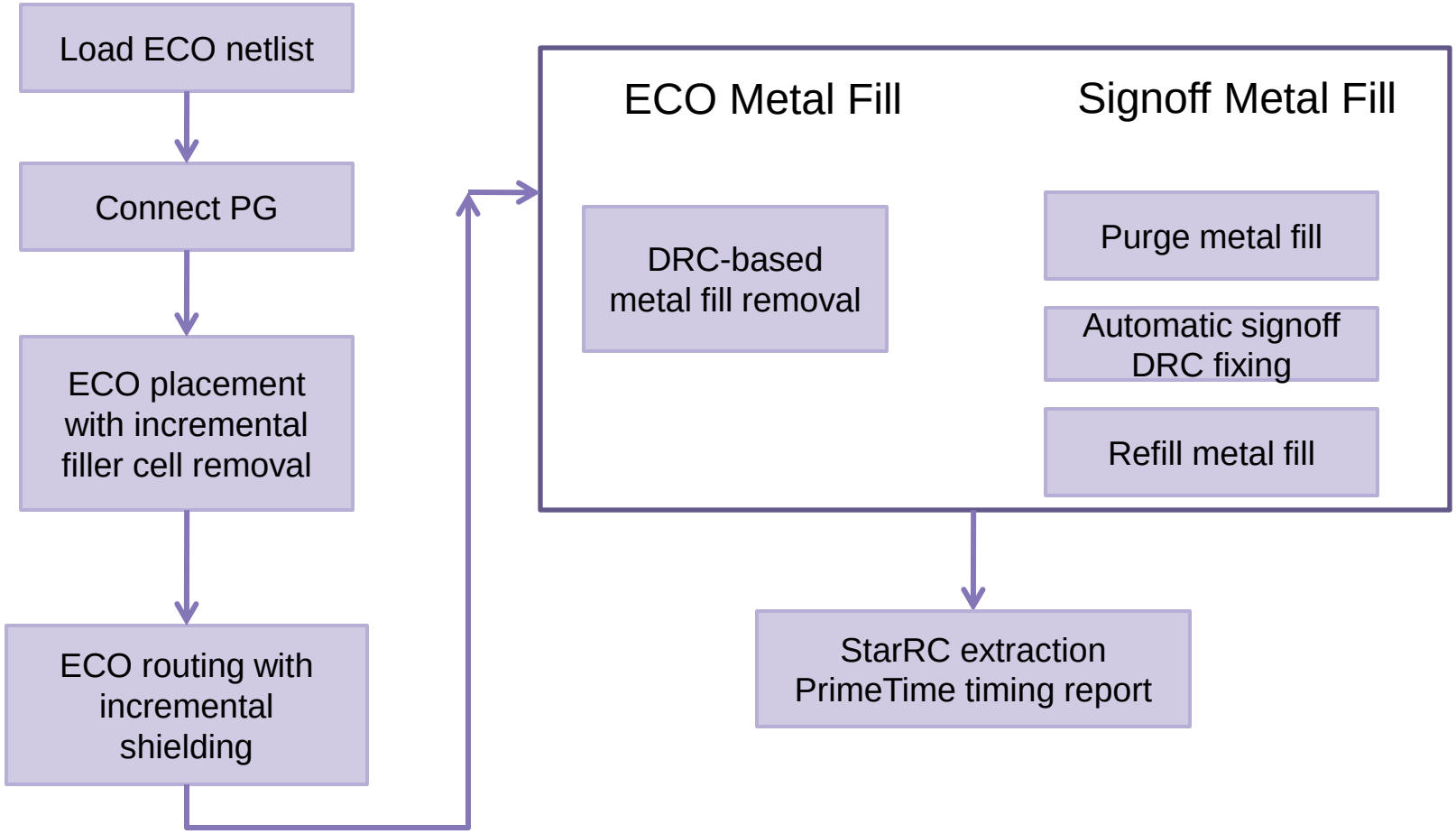
Physical Closure

- ICC route editing
- ICC-CD link
- In-design verify and autofix
  - DRC
  - Lithography
- In-design rail analysis
- In-Design signoff metal fill
- ECO metal fill flow
- Reference Methodology Update

Enhancements

- **IC Compiler and IC Validator incremental metal fill flow improvement**
  - Timing- and density-aware trim fill
  - Post-ECO DRC trim fill
- **Incremental automatic signoff DRC fixing flow runtime improvement**
  - Sparse cell (bounded boxes) and limited layers (incremental validation)
  - Fix only DRC violations targeted by automatic signoff DRC fixing
- **Placement ECO runtime improvement**
  - `place_eco_cell`
  - Channel-aware ECO placement
  - Incremental filler cell removal

ECO Reference Methodology Flow



Summary

- G-2012.06 ECO 1.25x to 1.8x runtime improvement
- DRC trim metal fill feature
  - Delivers 2x to 6x runtime improvement
  - Minimal timing degradation
- Timing-driven metal fill insertion versus minimum density
  - Delivers minimum density over timing
  - Minimal runtime and memory difference
  - Small timing degradation

ICC Design Closure Summary

- Timing Closure
  - Signoff_opt
  - PrimeTime Cheetah timing and DRC ECO flow
  - Enhanced ECO placement and buffering
- Physical Closure
  - Excellent built in route editing capabilities
  - Seamless IC Compiler/Custom Designer link
  - In-design ICV:
    - check and fix with ICV for DRC and Lithography
    - Signoff metal fill and ECO flow
  - In-design PrimeRail:
    - IR / EM analysis, rail integrity checks and decap insertion

Thank You


SYNOPSYS®
25
Years