

Agenda

**DAY
3**

4

Clock Tree Synthesis (Lab continued)



5

Routing



6

Chip Finishing



CS

Customer Support

Unit Objectives



After completing this unit, you should be able to:

- Explain the three *routing operations*
- Perform pre-routing checks and setup
- Route and optimize the design using *route_opt*
- Perform functional ECOs

Introducing Zroute

- Zroute supports advanced design rules for technologies 45nm and below
- Three characteristics separate Zroute from traditional routers:
 - State of art routing technology
 - Concurrent DFM optimizations
 - Multi-threaded operation
- The “classic” router is still available
 - Used in previous versions of IC Compiler



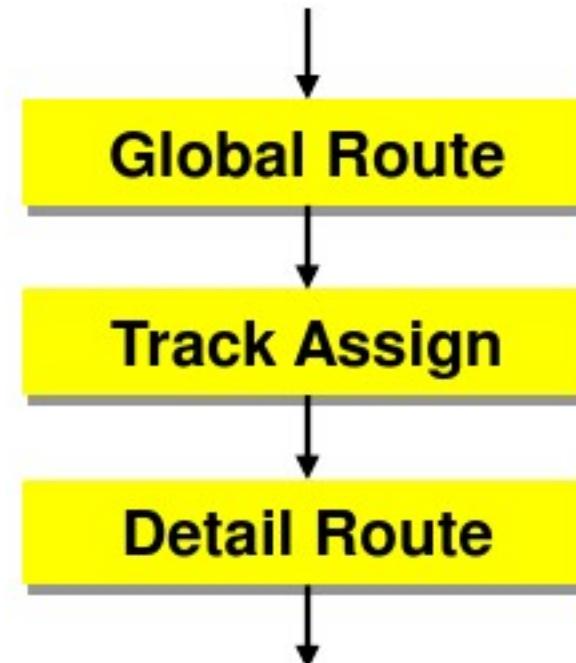
10X Speed-up, higher QoR, better manufacturability

Zroute Feature Summary

- **Fully multi-threaded**
- **Concurrent:**
 - Antenna rules optimization
 - Redundant via insertion
- **Wire spreading, widening and shielding**
- **Soft DRC rule support**
- **Reduced total wire length**
- **Fewer vias, jogs and notches**
- **Simplified flow with fewer options required**

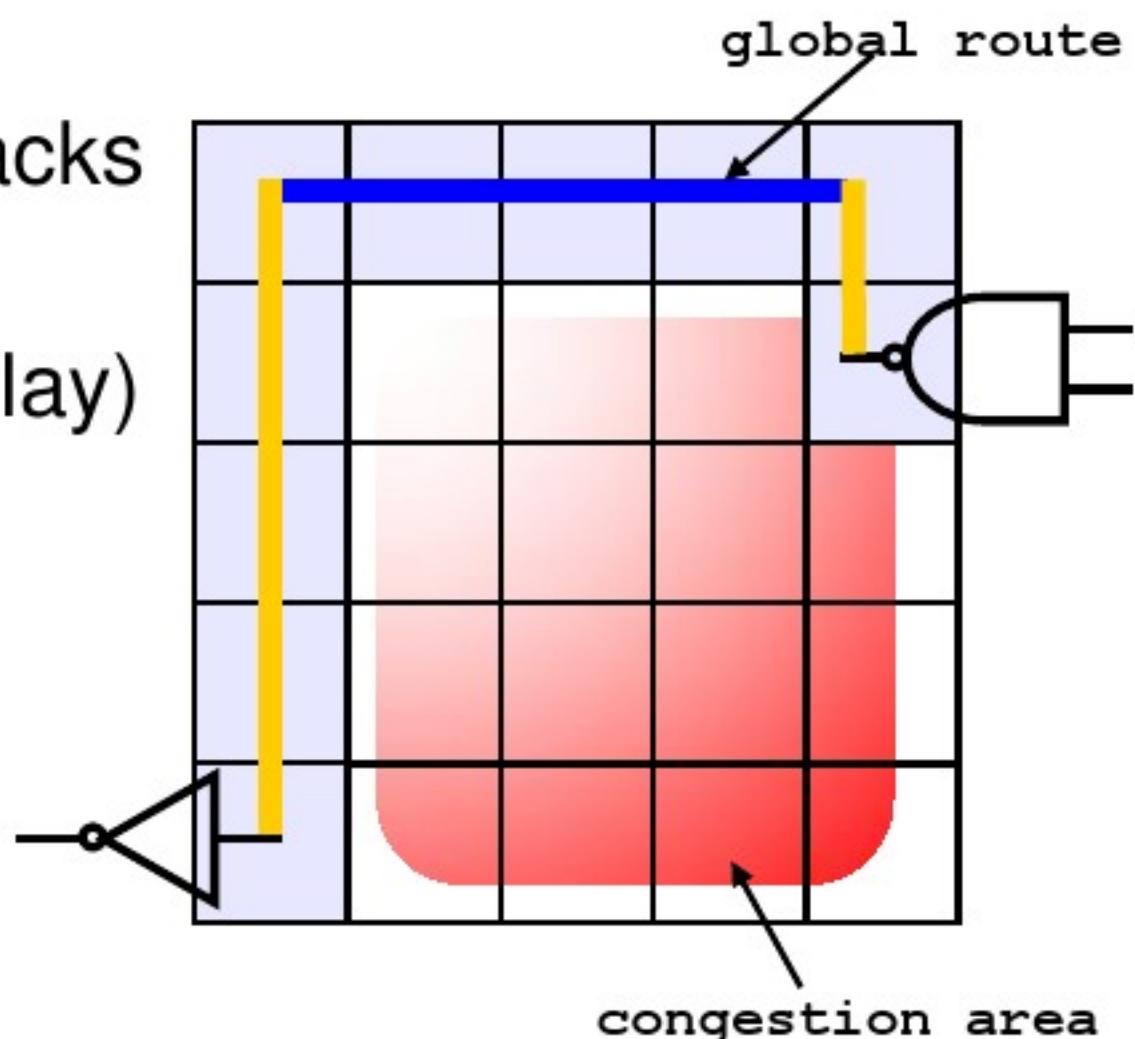
Routing Operations of route_opt

- **route_opt performs:**
 - Global Routing
 - Track Assignment
 - Detail Routing
- **After route_opt all nets will be completely connected**
- **Also performs concurrent optimization of:**
 - Timing, area and power
 - Buffering DRCs (max_cap/max_transition, etc)
 - Physical DRCs



Route Operations: Global Route (GR)

- GR assigns nets to specific metal layers and global routing cells (Gcells)
- GR tries to avoid congested Gcells while minimizing detours:
 - Congestion exists when more tracks are needed than available
 - Detours increase wire length (delay)
- GR also avoids:
 - P/G (rings/straps/rails)
 - Routing blockages

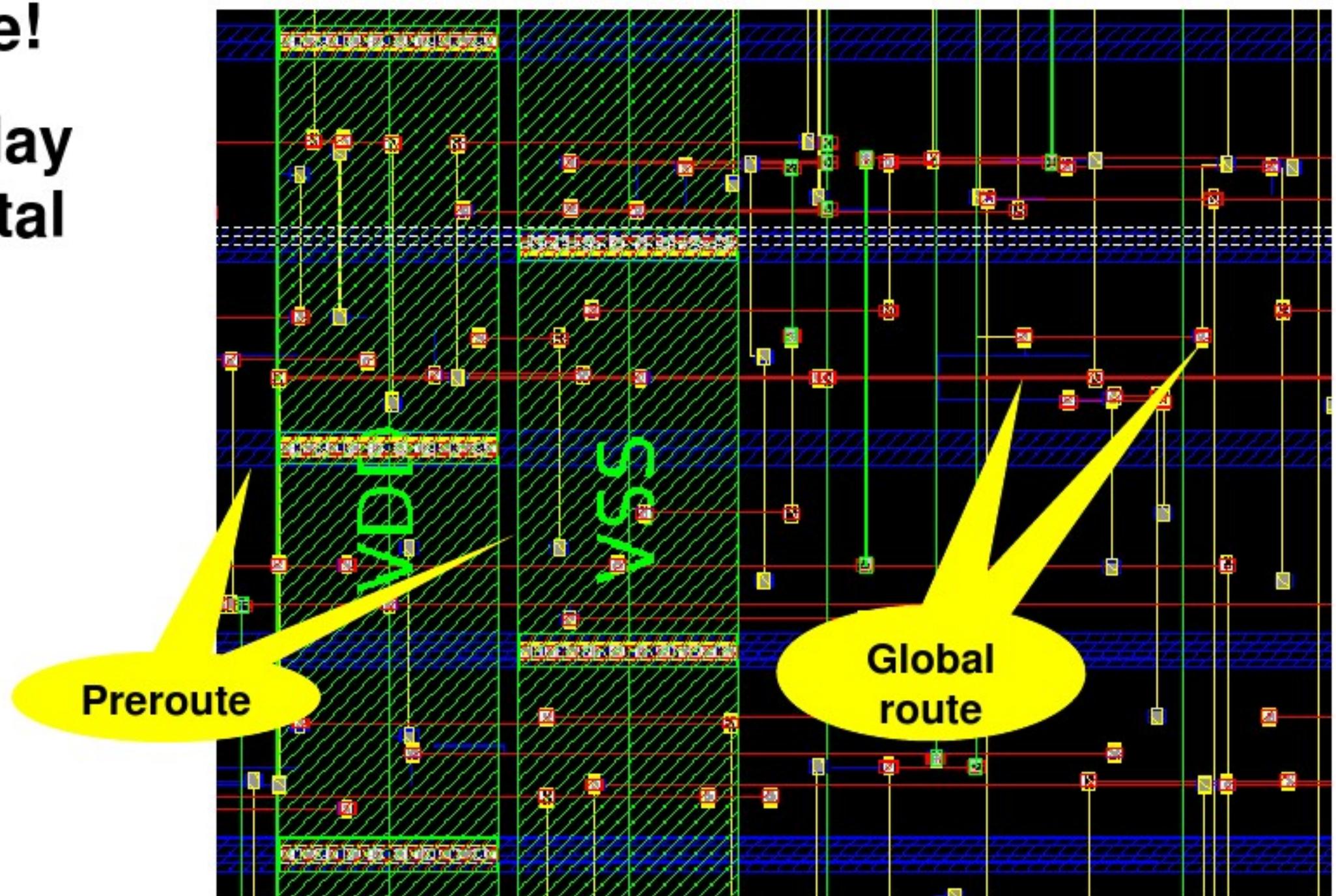


Metal traces exist after Global Route. **True or False?**

Route Operations: Global Route Summary

Answer: False!

GR does not lay down any metal traces.



Route Operations: Track Assignment

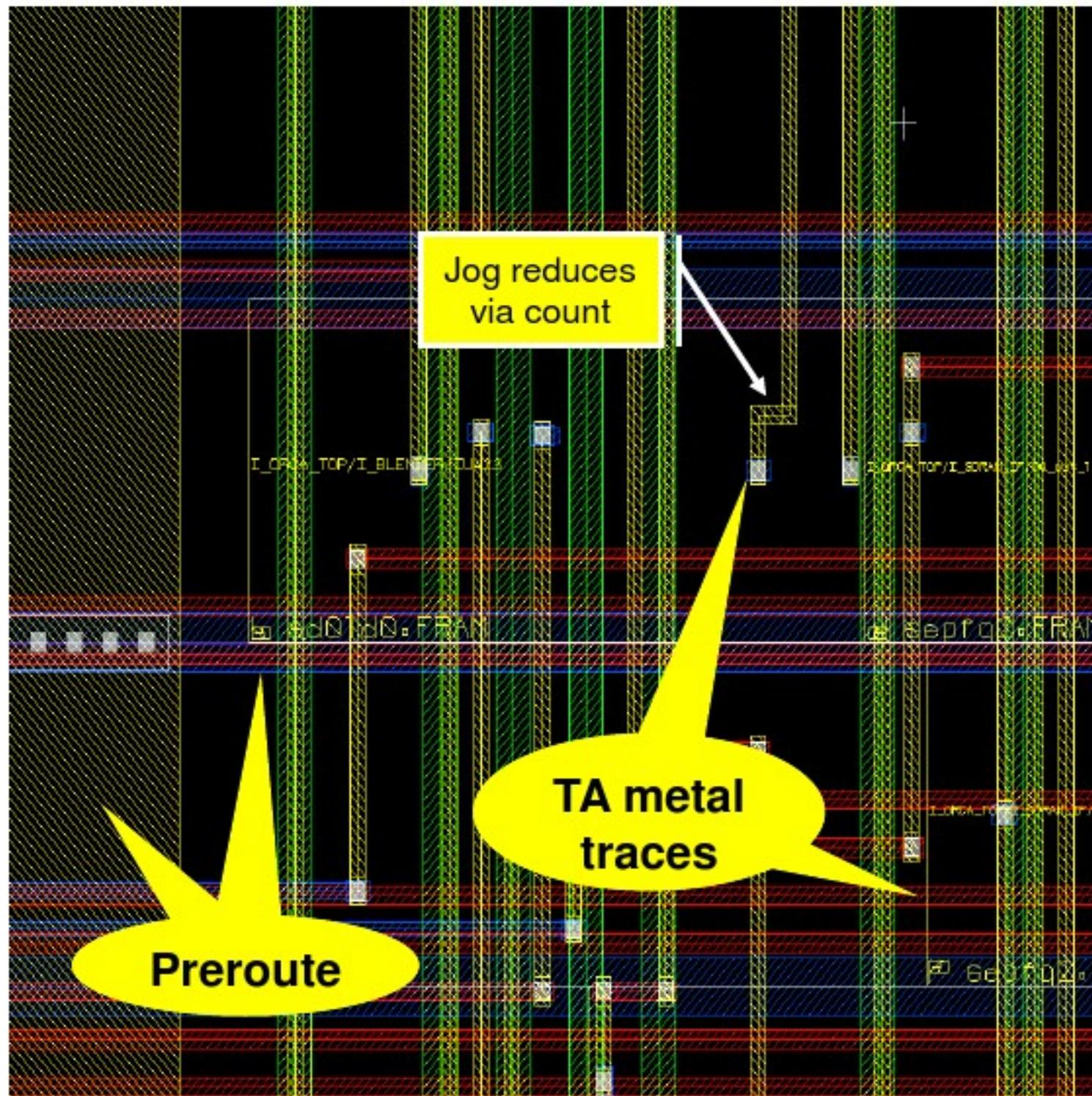
■ Track Assignment (TA):

- Assigns each net to a specific track and lays down the actual metal traces

■ It attempts to:

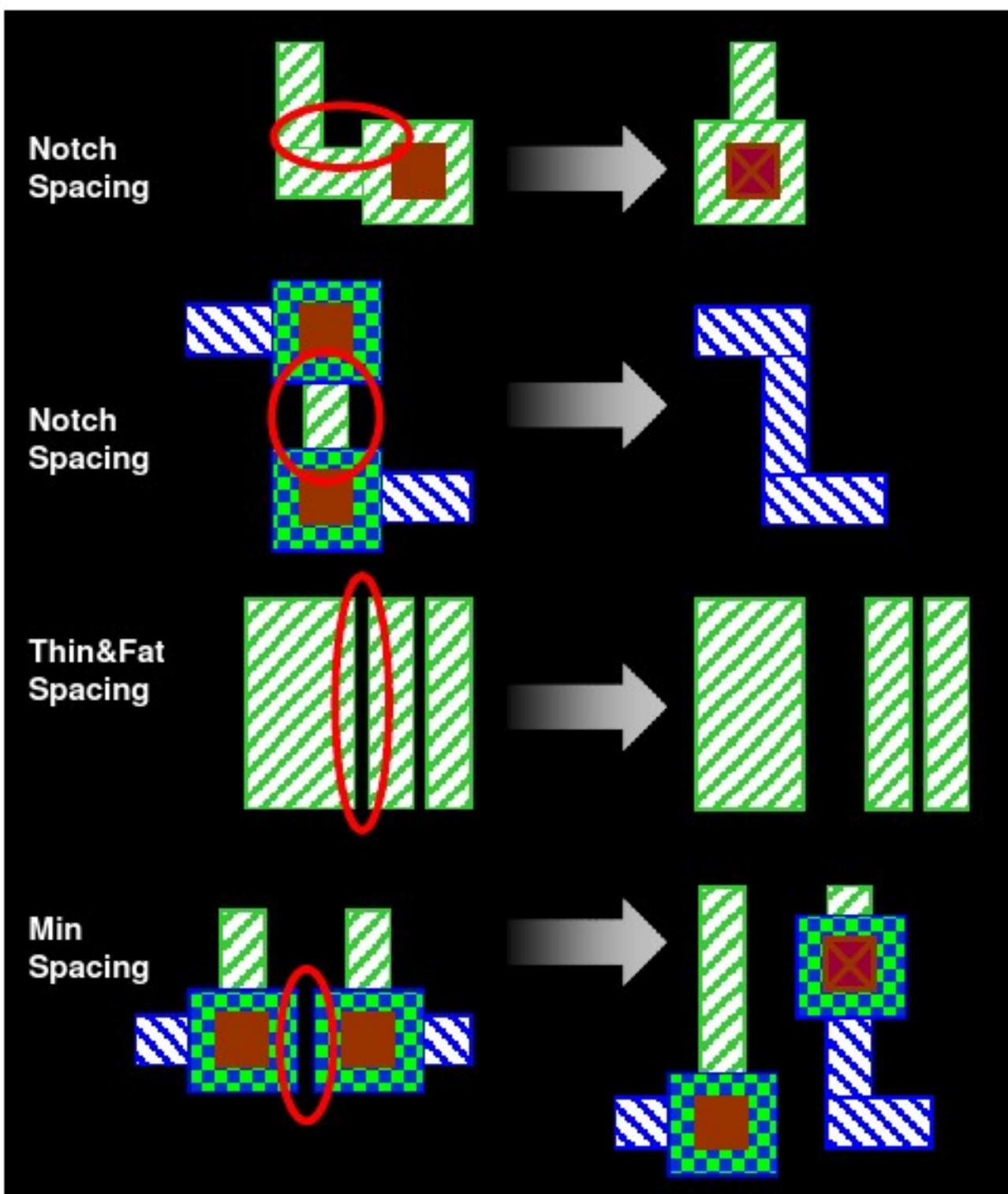
- Route each layer in its preferred direction
- Make long, straight traces
- Reduce the number of vias

■ TA does not check or follow physical design rules

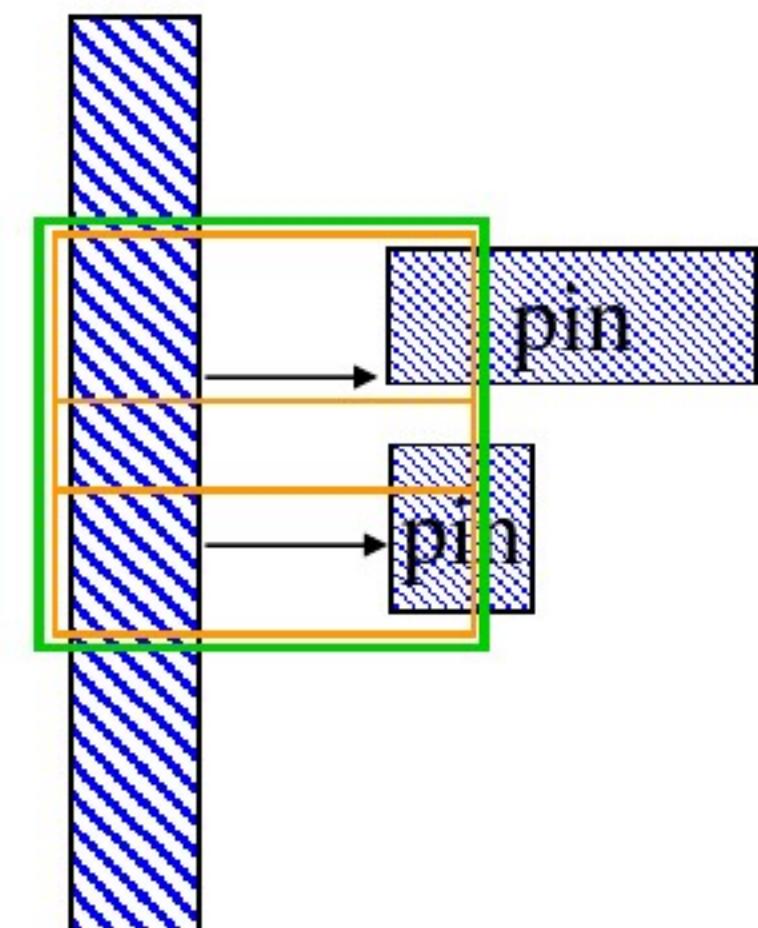


Route Operations: Detail Routing

Detail route fixes physical design rule violations



wire



Violations of identical rules that are in close proximity to each other will be merged and reported as one violation

Test for Understanding



- 1. What does Global Routing do in congested areas?**
- 2. Assignment of nets to metal layers is done during the Track Assignment stage. T or F?**
- 3. Will IC Compiler route a metal trace in the “non-preferred” direction?**
- 4. Will detailed route merge DRC violations on different layers, i.e. Metal-Metal spacing and Metal2-Metal2 spacing?**

Unit Objectives: Perform Pre-routing Checks



Design Status, Start of Routing Phase

- Placement - completed
- CTS – completed
- Power and ground nets – pre-routed
- Estimated congestion - acceptable
- Estimated timing - acceptable (~0ns slack)
- Estimated max cap/transition – no violations

```
report_constraints -all
```

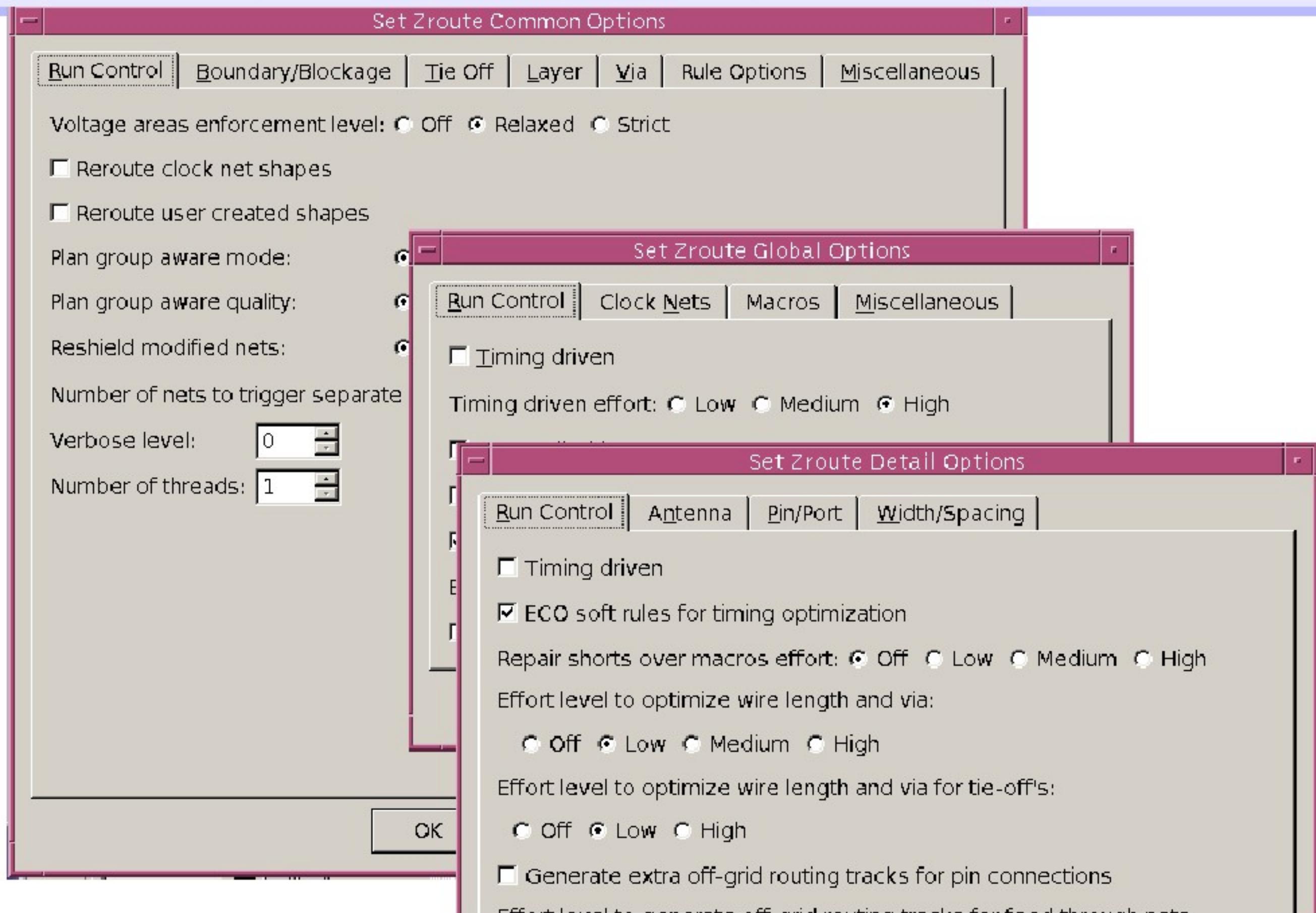
Pre-Route Checks

- Check design for routing stage readiness
- There should be no unexpected:
 - Ideal nets
 - High fanout nets
- Use `check_physical_design` to check a design's prerequisites for routing and report a list of violations



```
check_physical_design -stage pre_route_opt
all_ideal_nets
# Default high fanout threshold is 1000
# (default for variable high_fanout_net_threshold)
all_high_fanout -nets <-threshold #>
report_preferred_routing_direction
```

Perform pre-routing setup



Verify that Zroute is Enabled

- Check if Zroute is enabled

```
get_route_mode_options -zroute
```

- If false is returned → enable Zroute:

```
set_route_mode_options -zroute true
```

- OR: Use Zroute on/off switch in GUI route menu

Enabling Multithreading

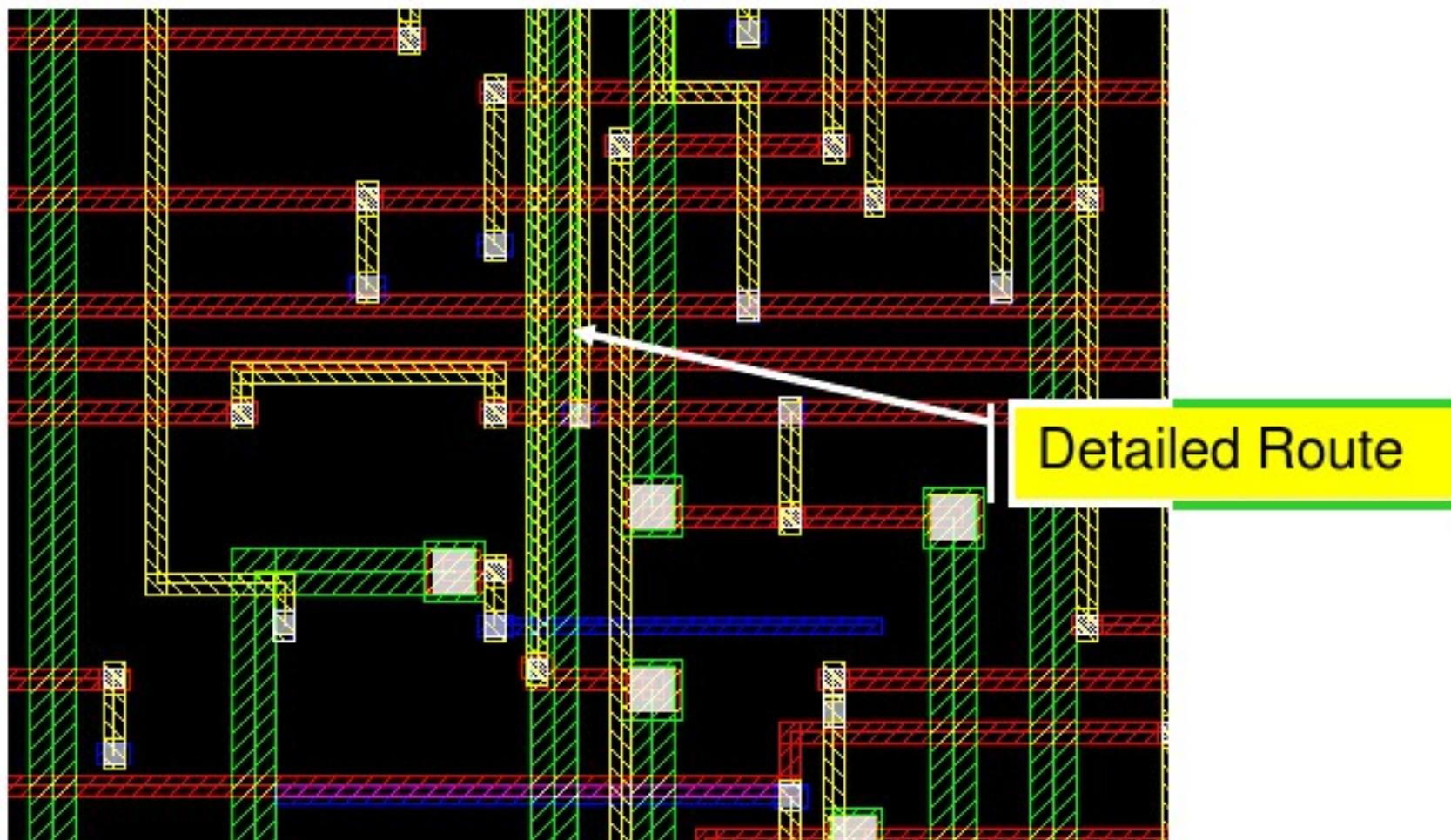
Zroute is designed as a multithreaded router

- Differs from distributed routing, which runs on multiple machines
- Multithreading can use all available cores, from multiple CPUs, on the same machine
- Specify the number of threads to be used as follows:

```
set_host_options -max_cores 8
```

- One IC Compiler license needed per four threads

Enable *Arnoldi* Delay Calculation Algorithm



Arnoldi provides more accurate net delay calculations from the extracted RC parasitics

- By default, Elmore is used

```
set_delay_calculation -arnoldi
```

Zroute Options

■ Zroute option setup commands

- *set_route_zrt_common_options*
- *set_route_zrt_global_options*
- *set_route_zrt_track_options*
- *set_route_zrt_detail_options*

■ Commands to report/query Zroute options

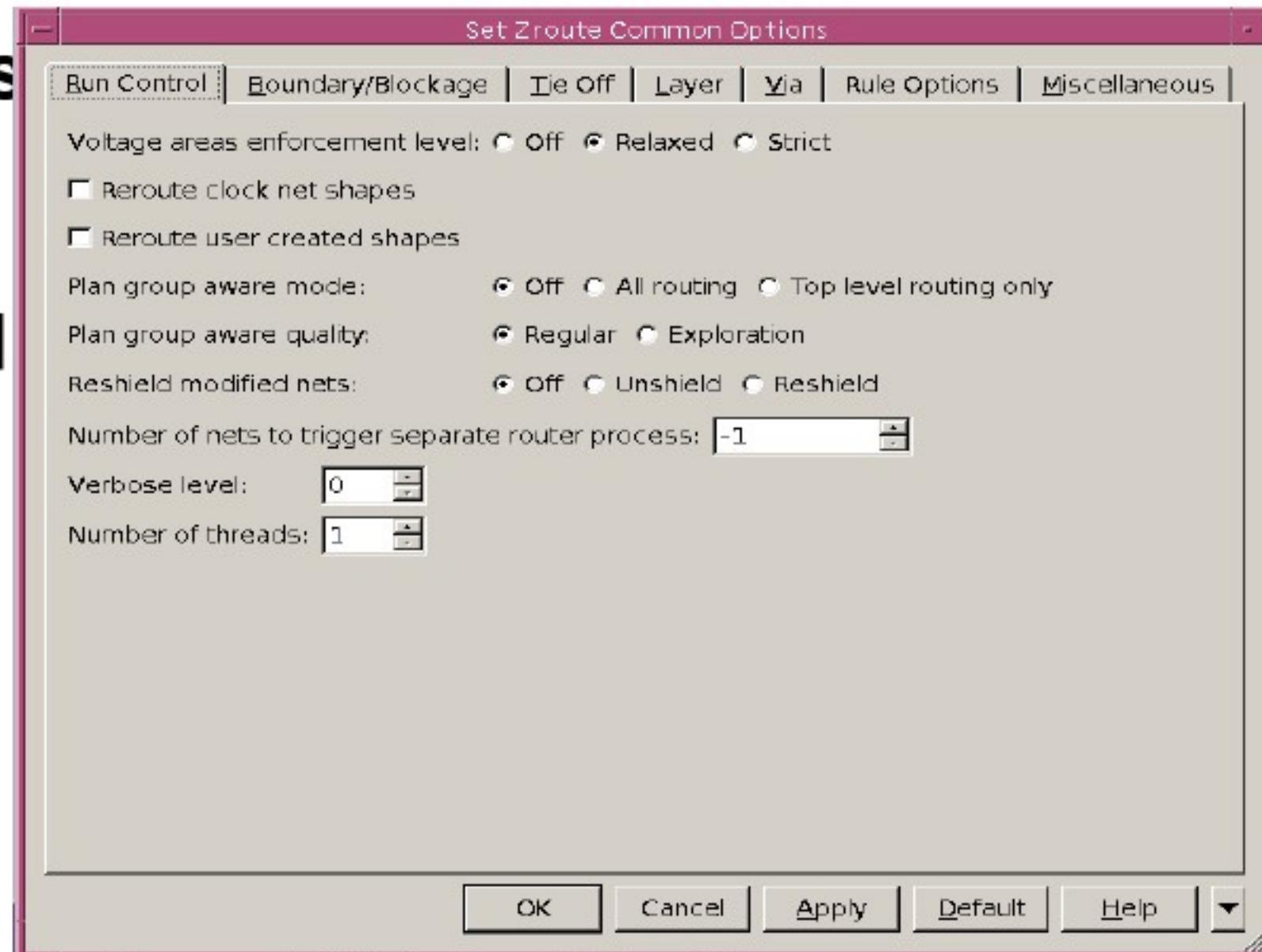
- *report_route_zrt_*_options*
- *get_route_zrt_*_options -name option_name*

* → common|detail|global|track

Set Common Route Options

- Common route options are used to control global routing, track assignment, and detail routing

- Run settings
- Boundaries
- Layer restrictions
- Vias



- *Redundant via insertion* option is recommended

```
set_route_zrt_common_options -default true  
set_route_zrt_common_options \  
-post_detail_route_redundant_via_insertion medium
```

Set Global Route Options

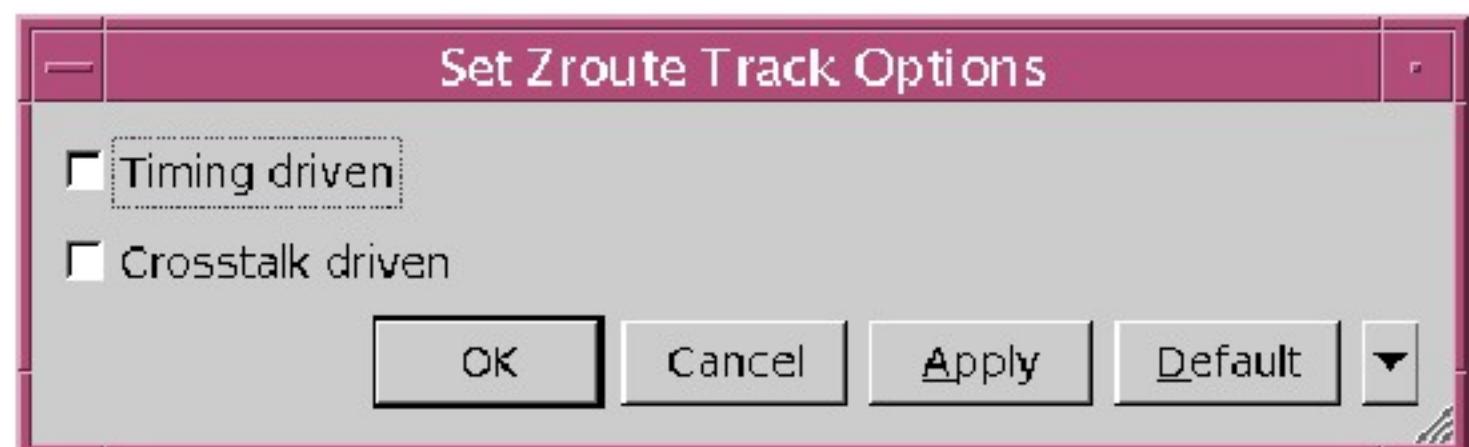
- Defines routing options that affect global routing only
 - Run control
 - Macro
 - Clock net routing
- Modify default settings if needed



```
set_route_zrt_global_options -default true  
set_route_zrt_global_options ...
```

Set Track Assignment Options

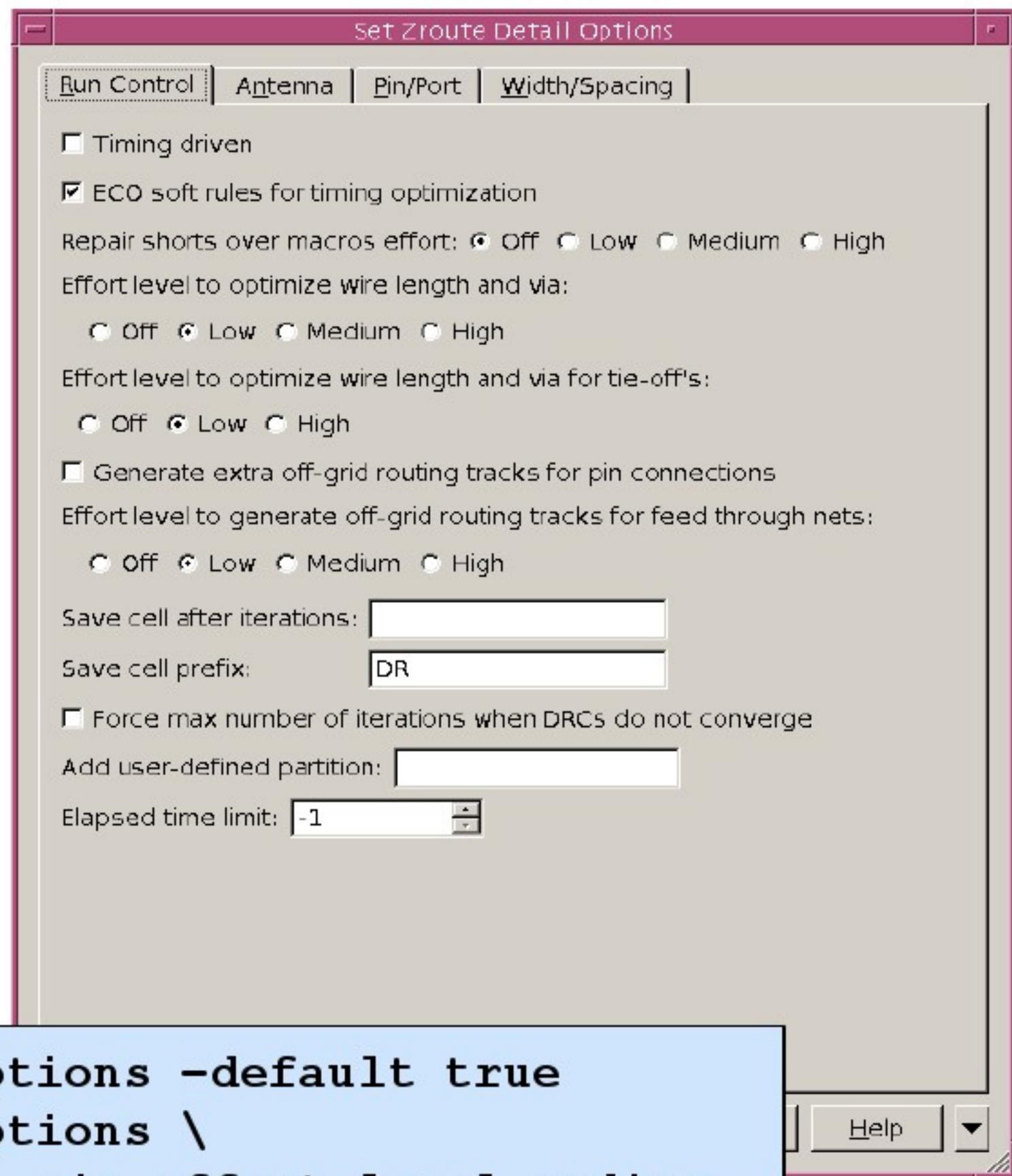
- **Defines routing options that affect track assignment**
 - Timing driven TA
 - Crosstalk driven TA
- **Modify default settings if needed**



```
set_route_zrt_track_options -default true  
set_route_zrt_track_options ...
```

Set Detail Route Options

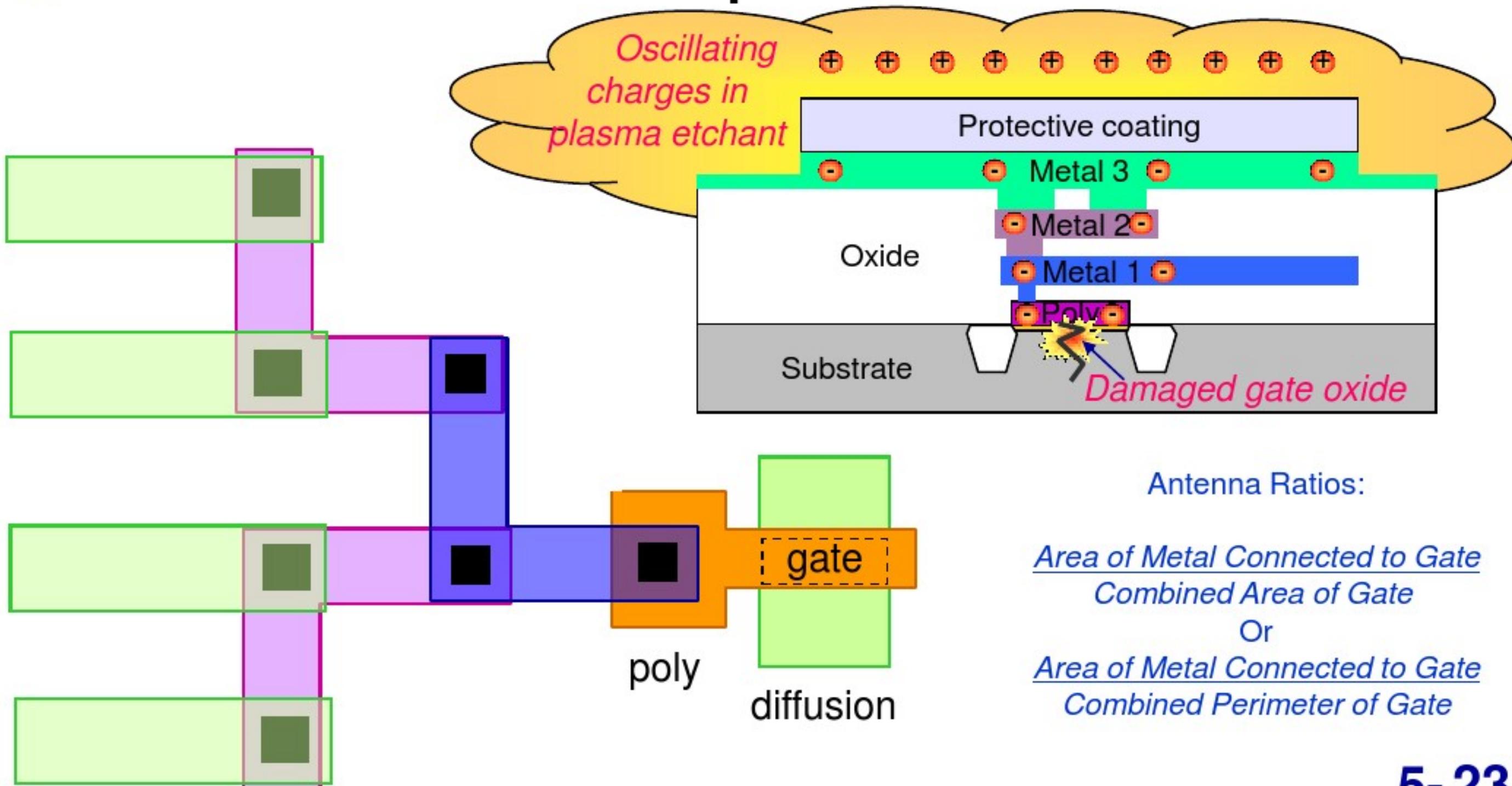
- **Controls detail route**
 - Antenna fixing
 - Pin/Port connection
 - Width/Spacing
 - Run controls
- **Via and wire optimization option is recommended when using redundant via insertion**



```
set_route_zrt_detail_options -default true  
set_route_zrt_detail_options \  
    -optimize_wire_via_effort_level medium
```

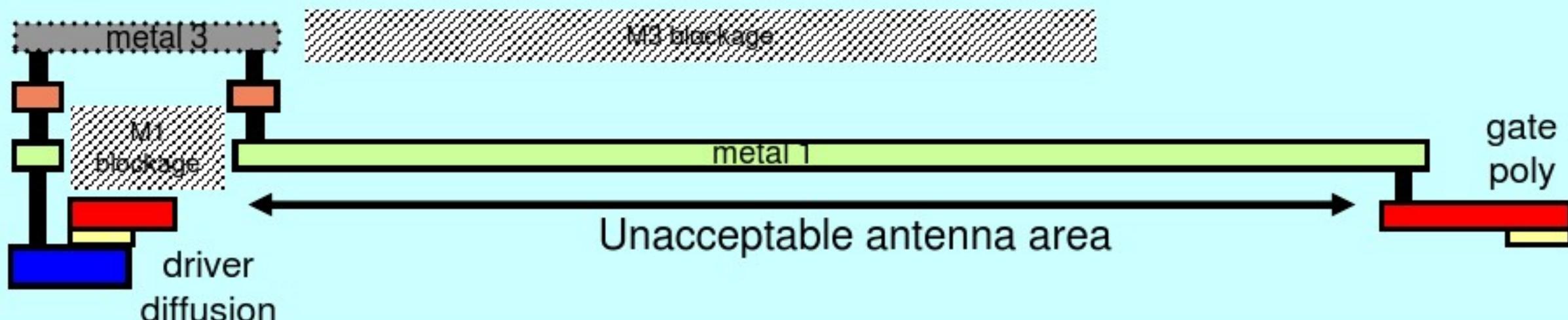
Antenna Violations

- As the total area of a wire increases during processing, the voltage stressing the gate oxide increases
- Antenna rules define acceptable total areas of wires

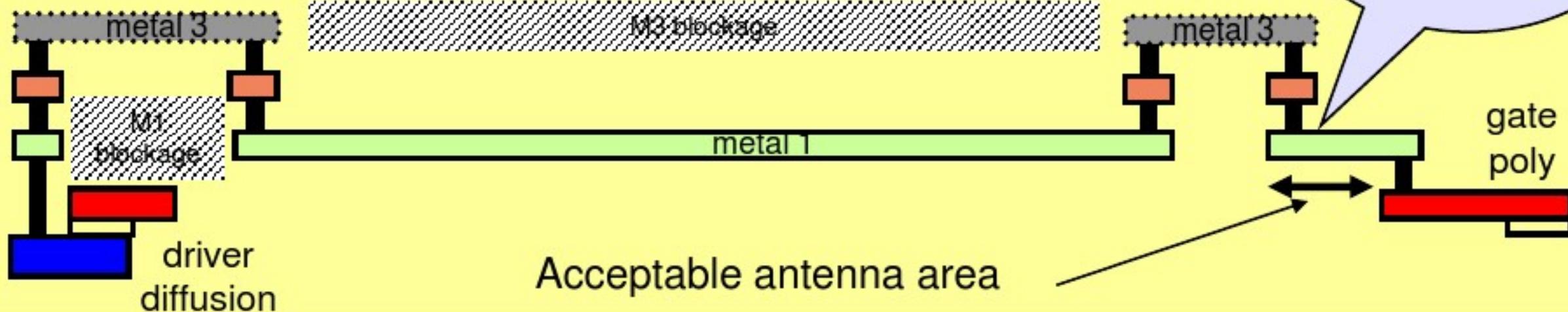


Fixing Antenna Violation by Layer Jumping

Before layer jumping



After layer jumping, to meet Antenna rules



Concurrent Antenna Fixing: Layer Jumping

- **Enable concurrent antenna fixing during signal detailed routing (optional)**

- Load antenna rules
- Enable antenna fixing during detailed route

```
source antenna_rules.tcl  
set_route_zrt_detail_options -antenna true
```

- **Concurrent antenna-fixing during the routing phase reduces overall turnaround time**

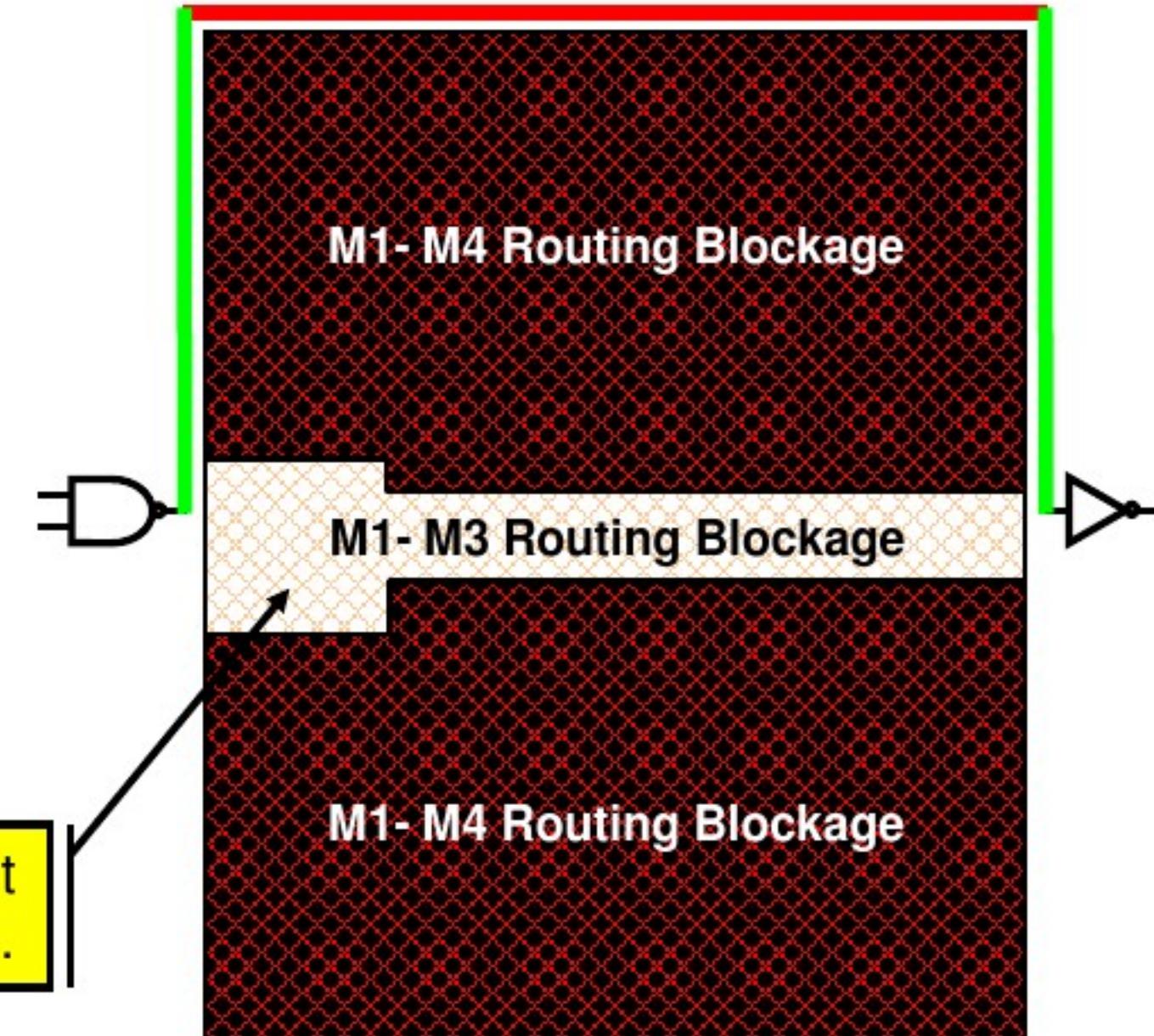
- No extra iterations for antenna fixing required during the chip finishing phase

Routing over Macros

By default IC Compiler will:

- Route over macros
- Not route where there is a routing blockage
- Not route through a narrow channel in the non-preferred routing direction

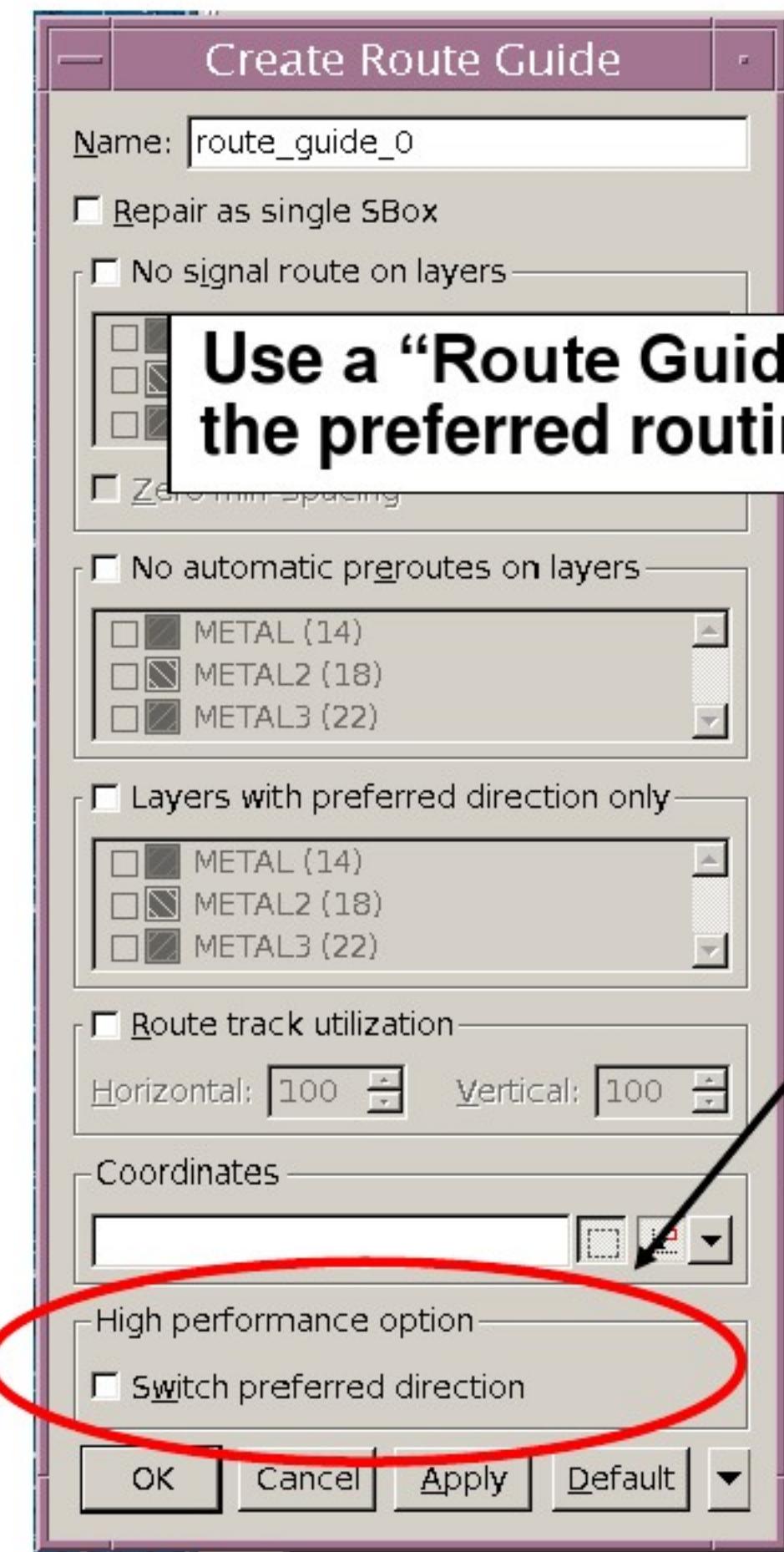
M4 has a horizontal routing channel but its preferred routing direction is vertical.



Macro with routing blockages

You need to change the preferred routing direction!

Change the Preferred Routing Direction



Use a “Route Guide” to change the preferred routing directions

M1- M4 Routing Blockage

M1- M4 Routing Blockage

Select this option to change the preferred routing direction for an area.

```
create_route_guide -name route_guide_0 \
-coordinates {{270 340} {491 485}} \
-switch_preferred_direction -no_snap
```

Define Routing Blockages

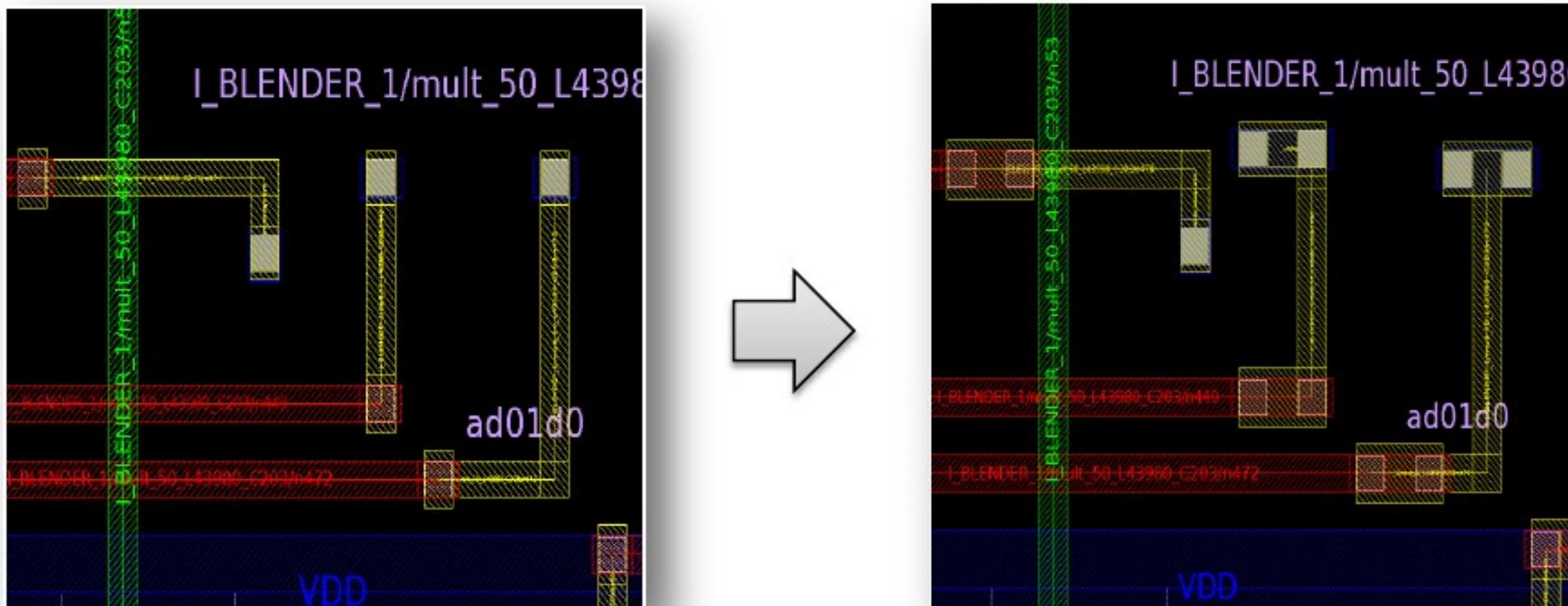
A routing blockage defines a region where no routing is allowed on a specific layer

- Affected layer must be specified using *blockage layers* instead of the techfile layer names (see notes)
- The region of the blockage is specified in one of two ways:
 - ◆ Use the `-bbox` option to specify the region for a rectangular routing blockage.
 - ◆ Use the `-boundary` option to specify the region for a rectilinear routing blockage.

```
# Create rectangular routing blockages on
# metall1 and via1 blockage layers
create_routing_blockage -bbox {30 100 120 340} \
-layers {metall1Blockage via1Blockage}
```

Redundant Via Insertion

- Replaces single-cut vias with multiple-cut via arrays or another single-cut via with a different contact code
- Redundant via insertion algorithm options:
 - Concurrent soft-rule-based redundant via insertion
 - Postroute redundant via insertion (discussed in next Unit)



Redundant Via Insertion: Setup

- Zroute reads default via definitions from the technology file
 - Via definitions are then used to generate an optimized via mapping table
- Check the default via mapping table:
 - If OK, proceed with redundant via insertion
 - If changes are required, define new redundant via rules

```
# Check default via mapping table:  
insert_zrt_redundant_vias -list_only  
  
# Optional definition of new redundant via rules  
define_zrt_redundant_vias -from_via {VIA23 VIA34} \  
-to_via {VIA23 VIA34} \  
-to_via_x_size {1 2} -to_via_y_size {2 1}
```

Redundant Via Insertion: Execution

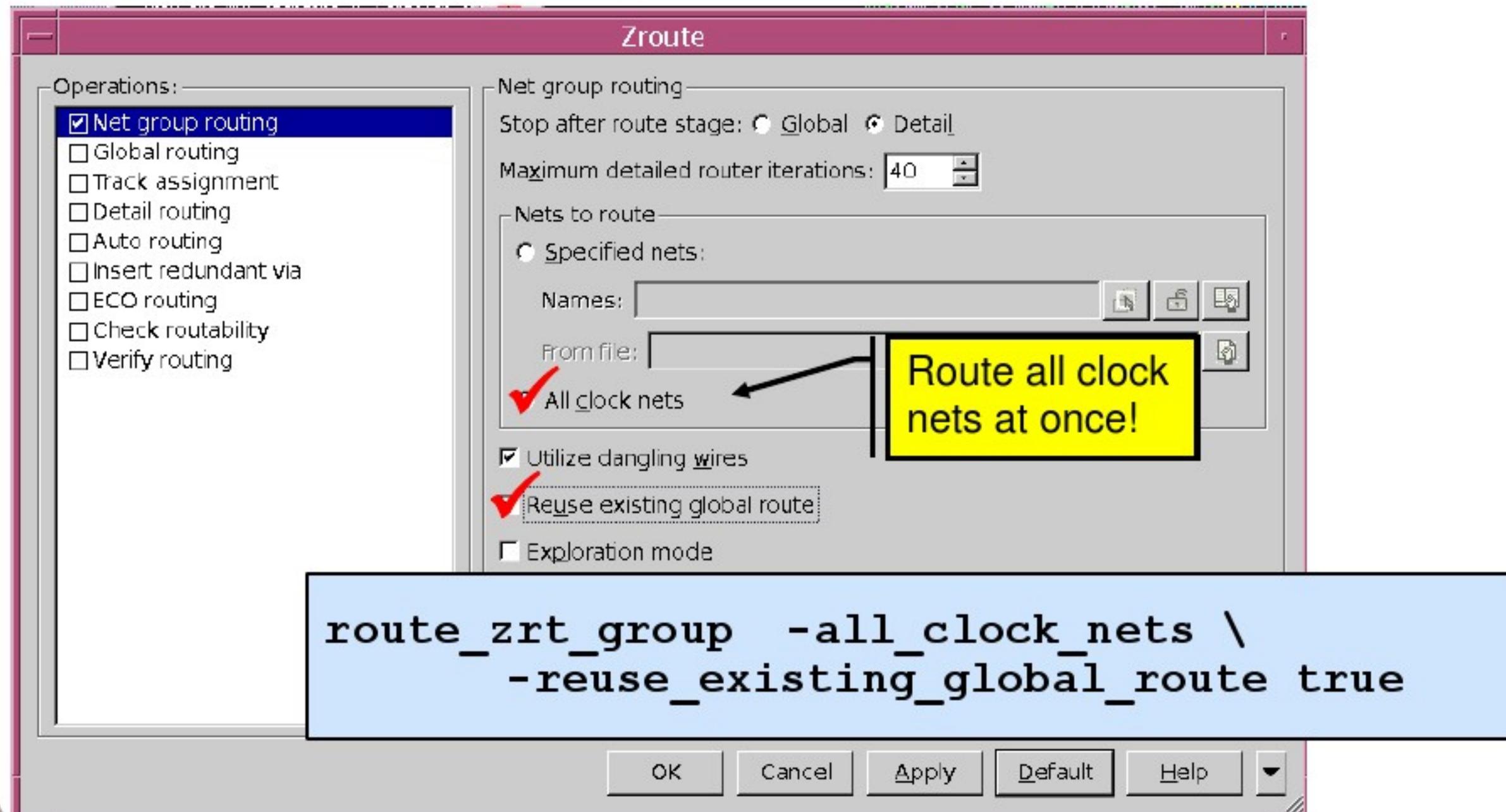
Redundant via insertion (RVI)

- Checks physical design rules to minimize DRC violations
- Happens during any detail routing operation

```
# Enable automatic insertion of redundant vias
# in subsequent routing steps
set_route_zrt_common_options \
    -post_detail_route_redundant_via_insertion medium
set_zrt_detail_route_options \
    -optimize_wire_via_effort_level medium
define_zrt_redundant_vias ...
route_opt -initial_route_only; # RVI happens here
...
route_opt -skip_initial_route; # RVI happens here
```

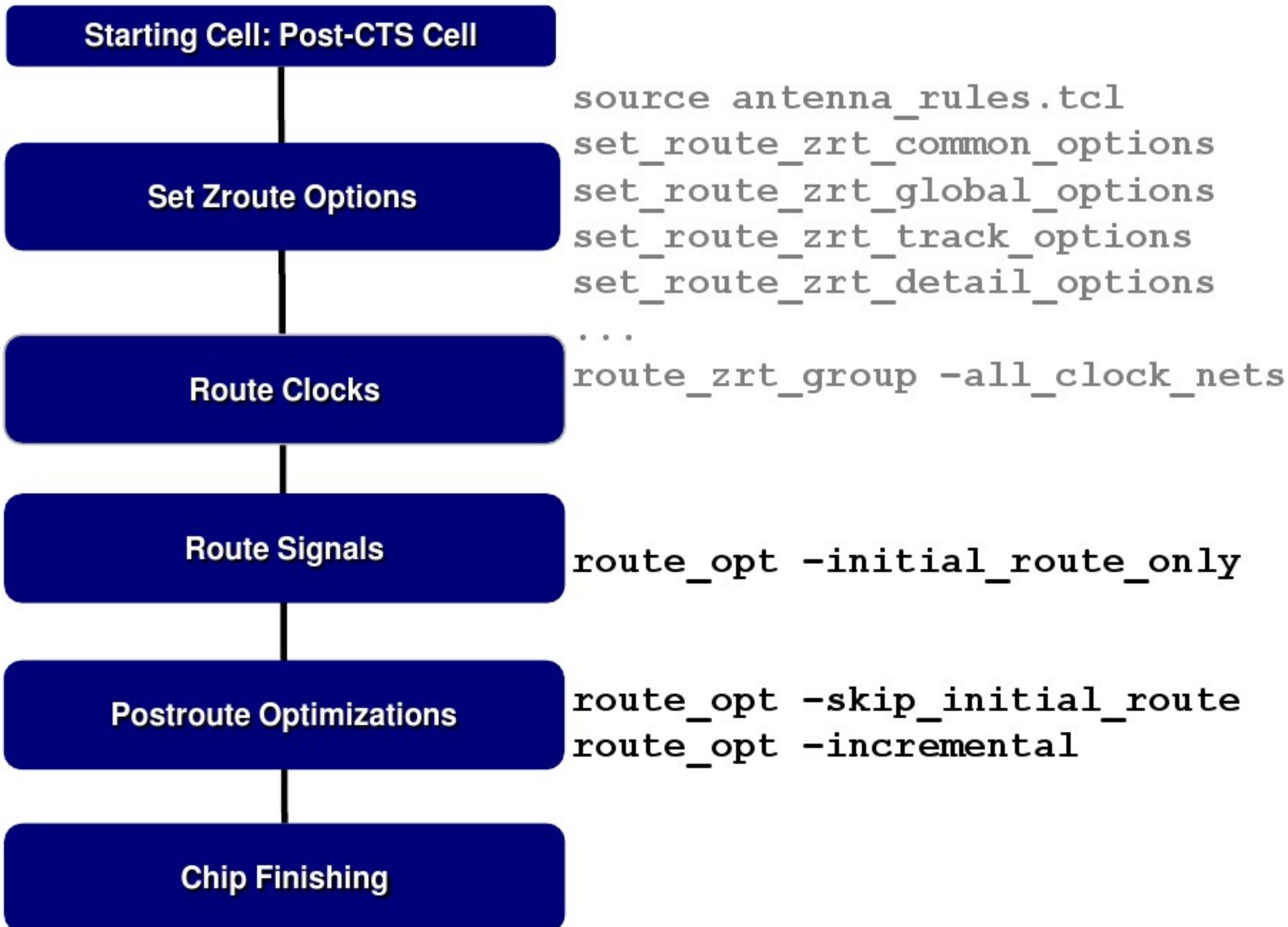
Route Clock Nets First

... if not already routed during clock_opt!



This will give the clocks free rein. If all nets were routed together, clocks would not have any particular priority over other nets.

Basic Zroute Flow



Core Routing: route_opt

route_opt

-effort low | medium | high

-stage global | track | detail

-power

-xtalk_reduction

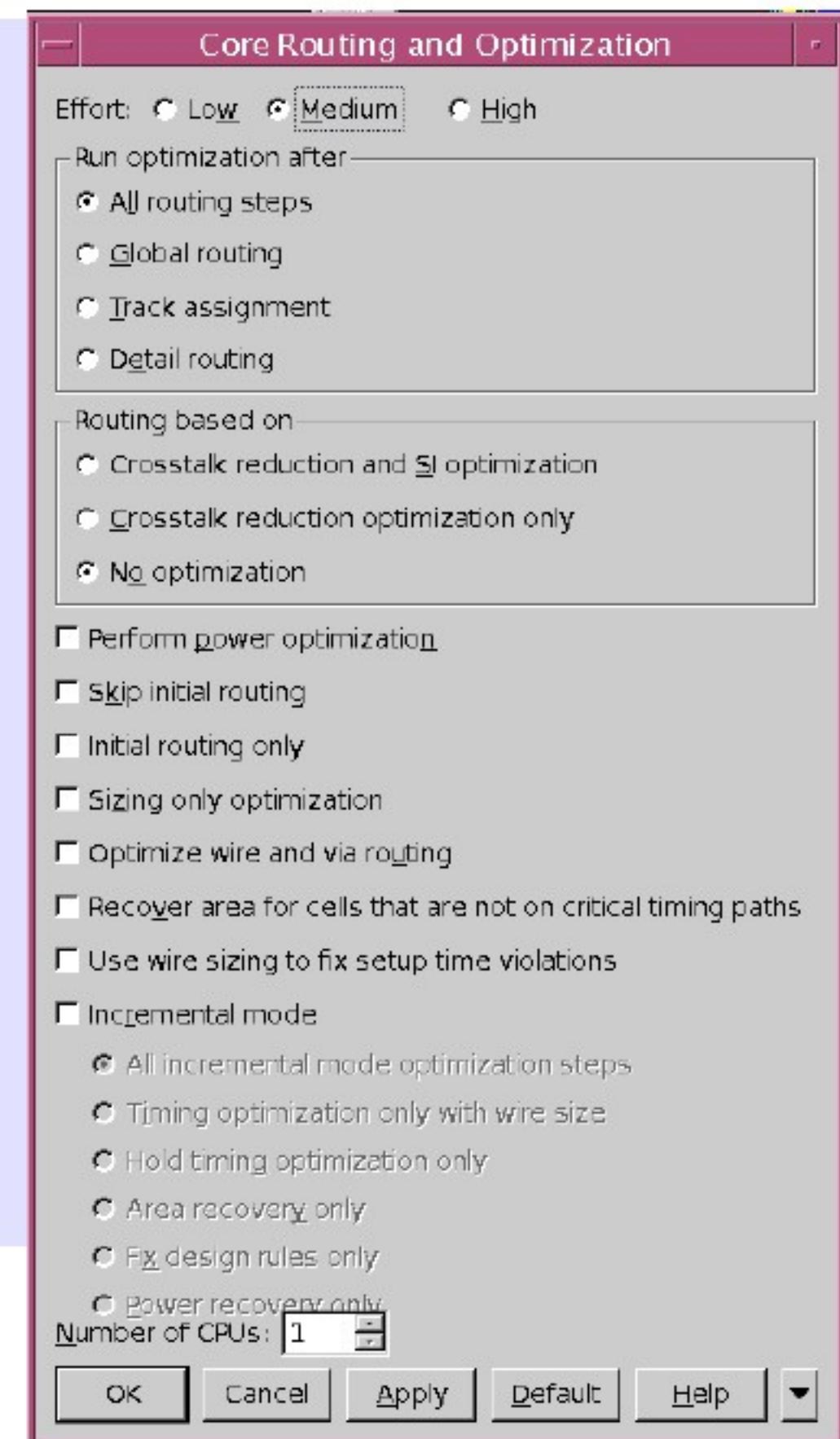
-initial_route_only

-skip_initial_route

-incremental

-area_recovery

...



First *route_opt*

- **Perform initial route only**

- Allows analysis (setup/hold timing, DRC, clock skew)
- Helps to determine post initial-route options

```
route_opt -initial_route_only
```

- **Initial routing entails *global routing*, *track assignment* and *detail routing***
- **All unrouted (signal) nets are now fully connected**
- **May have timing, max_tran/cap, and physical DRC violations**
 - Follow up with post initial-route **route_opt** optimizations

Post Route Optimization Examples

- Post initial-route full optimization

```
route_opt -skip_initial_route -effort medium -power
```

- If you have logical DRC violations

- Switch from default priority of timing over DRC

```
set_app_var routeopt_drc_over_timing true
```

```
route_opt -effort high -incremental -only_design_rule
```

- If additional specific optimization is needed:

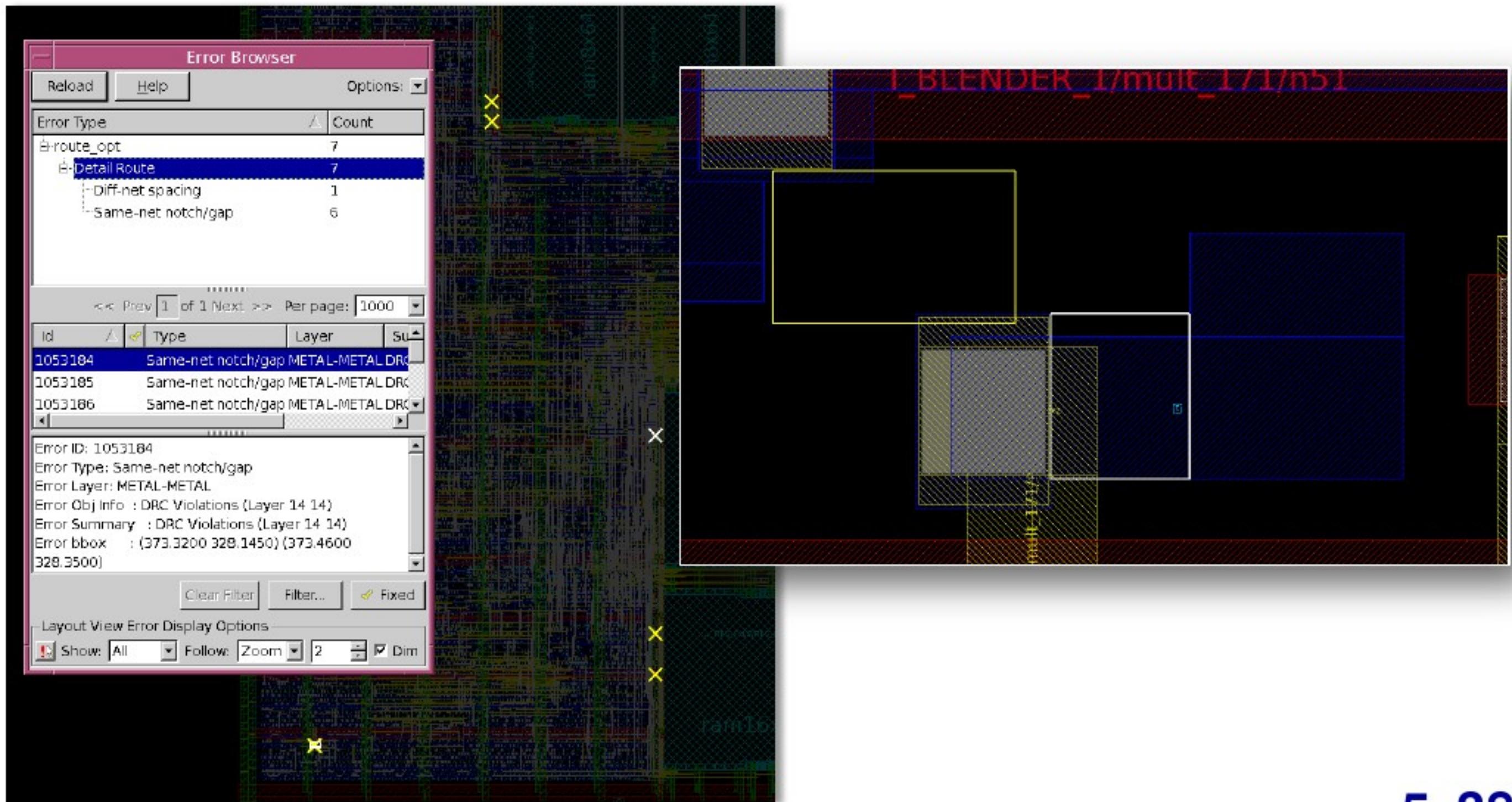
```
route_opt -size_only | -only_hold_time | \  
-only_wire_size | -wire_size
```

Getting Design Statistics

- After redundant via insertion, Zroute generates a conversion report:
 - Reports the optimized via conversion rate for each layer
 - ◆ The optimized via conversion rate includes both double vias and DFM-friendly bar vias, which have a single cut but a larger metal enclosure.
 - ◆ The distribution of optimized vias by weight for each layer
 - ◆ The overall double via conversion rate for the design
- The `report_design -physical` command gives a routing summary including double via conversion rates, global routing, track assignment, and detail routing

Check for Physical Design Rule Violations

```
verify_zrt_route; # Uses Zroute DRC engine  
route_zrt_detail -incremental true; # Fix DRCs
```

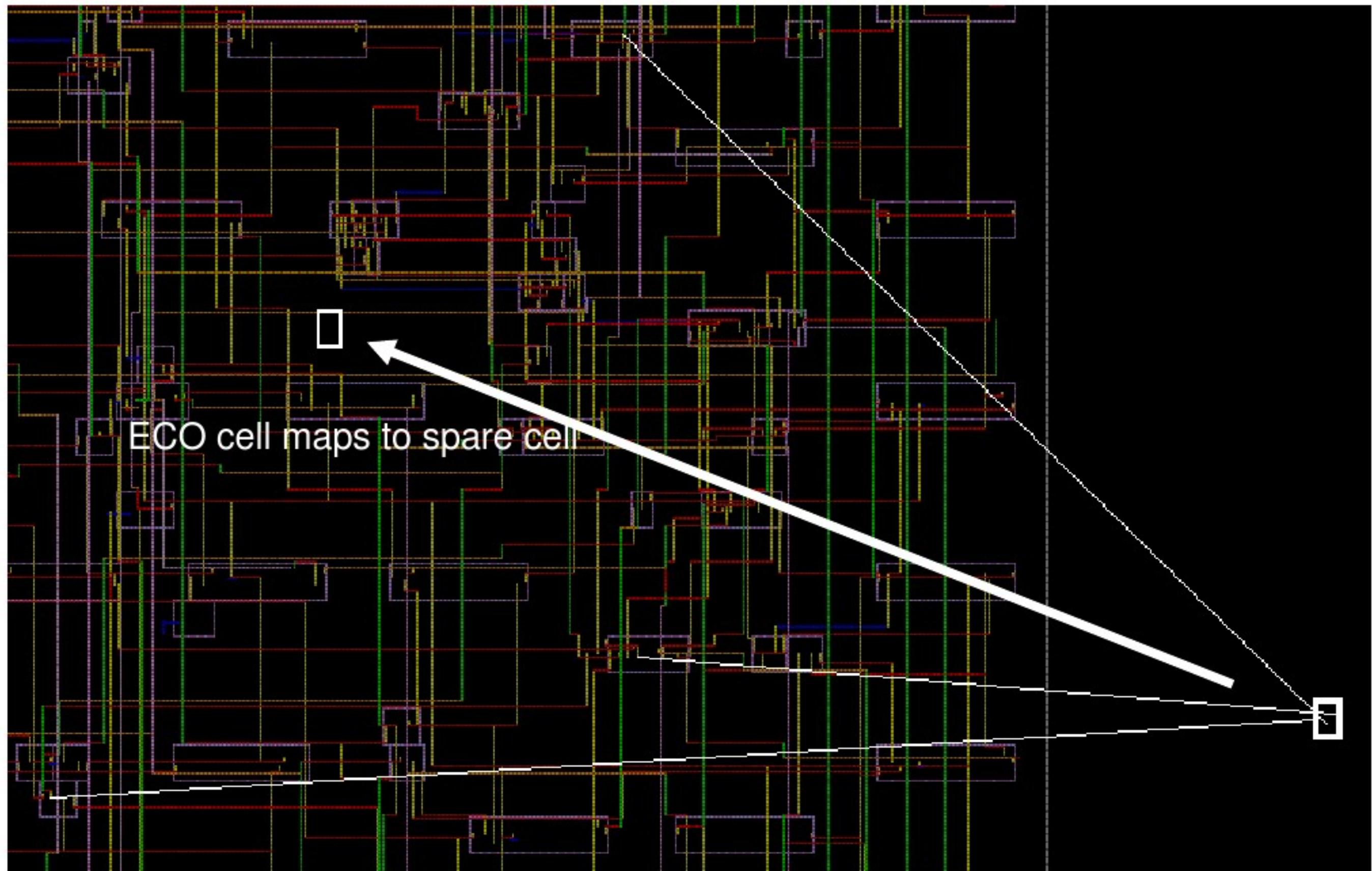


Verify Route: verify_zrt_route

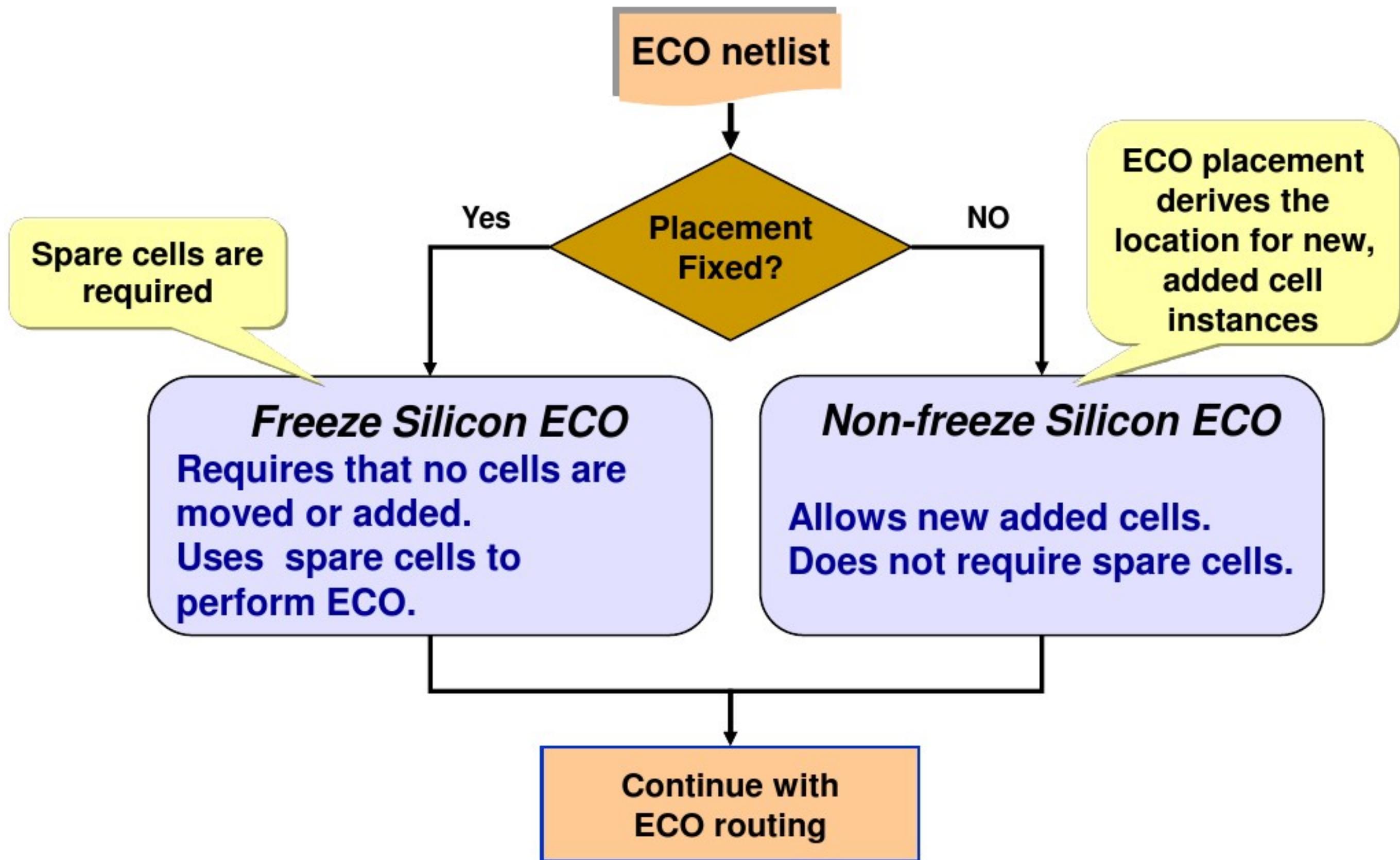
- **Checks signal and clock routing for**
 - Physical DRCs, opens, shorts and antenna violations
- **Does not check DRCs:**
 - Amongst pre-routed nets only (e.g. P/G grid structure)
 - On nets marked as type “user”
 - ◆ Wire shapes hand-created by the designer , not associated with any net (e.g. Logo, alignment marker, etc.)
- **Use verify_lvs to help debug opens**

```
verify_lvs -ignore_short -ignore_min_area
```

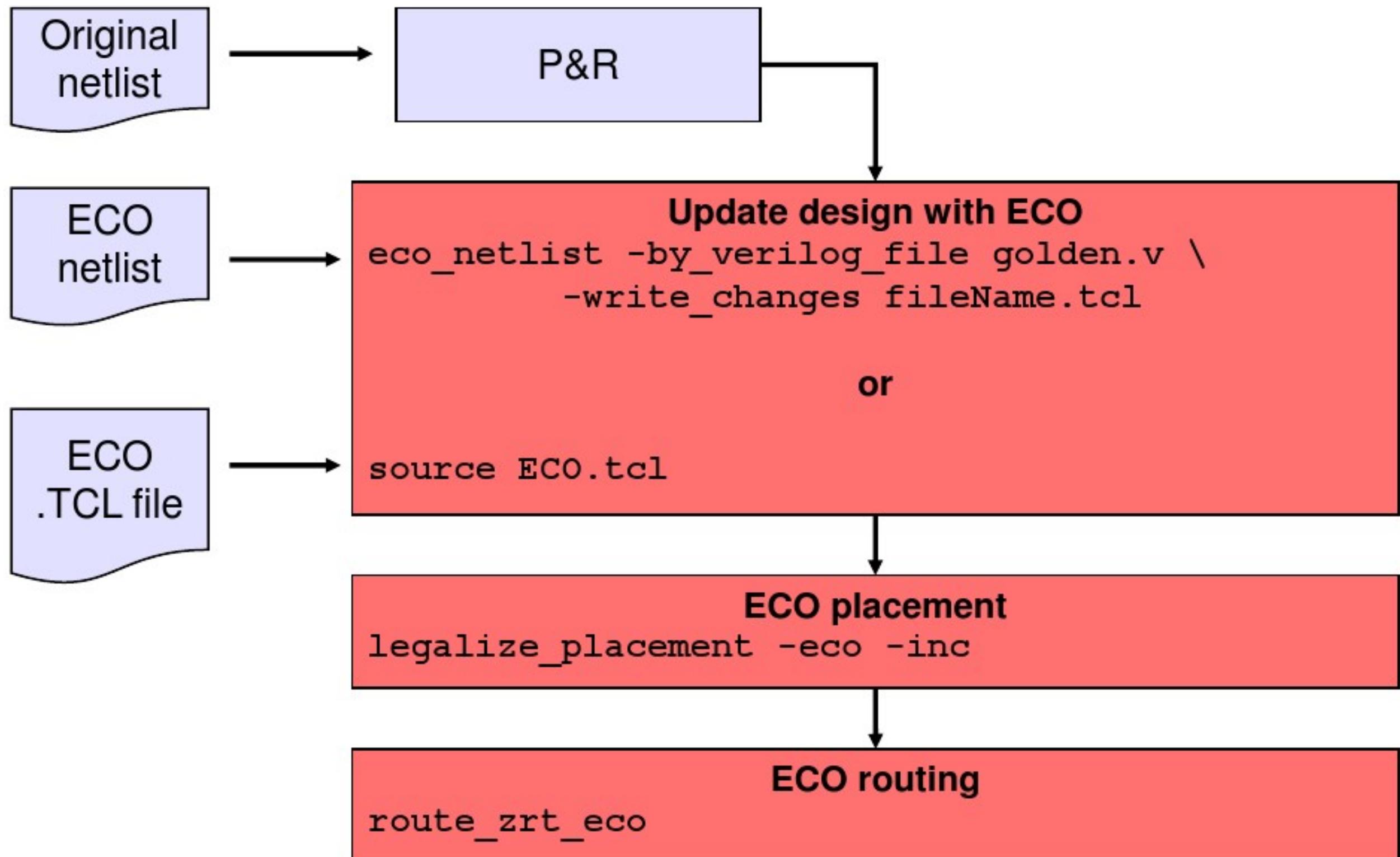
Unit Objectives: Perform functional ECOs



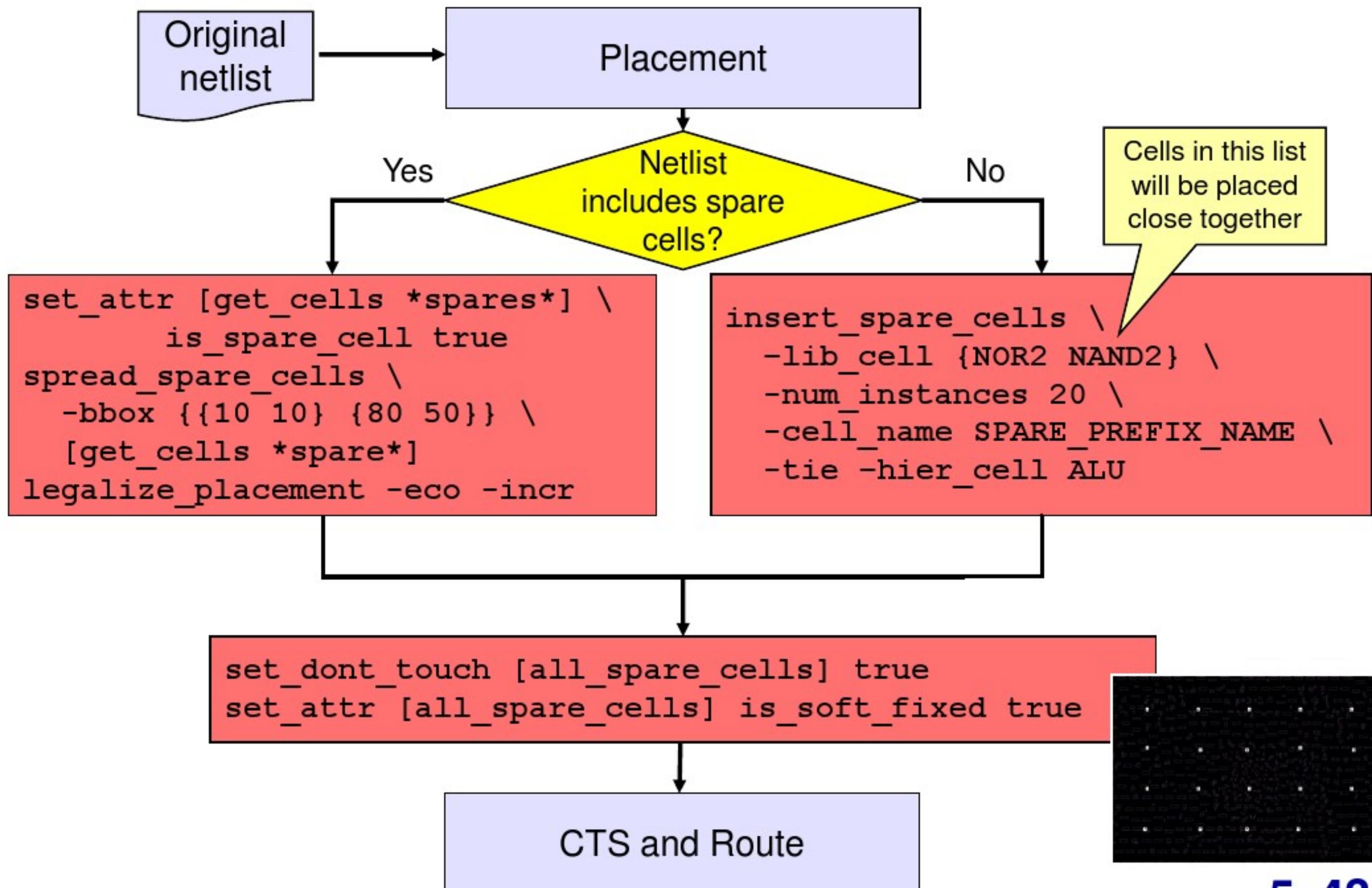
The Two Types of ECO Flows



Non-Freeze Silicon ECO



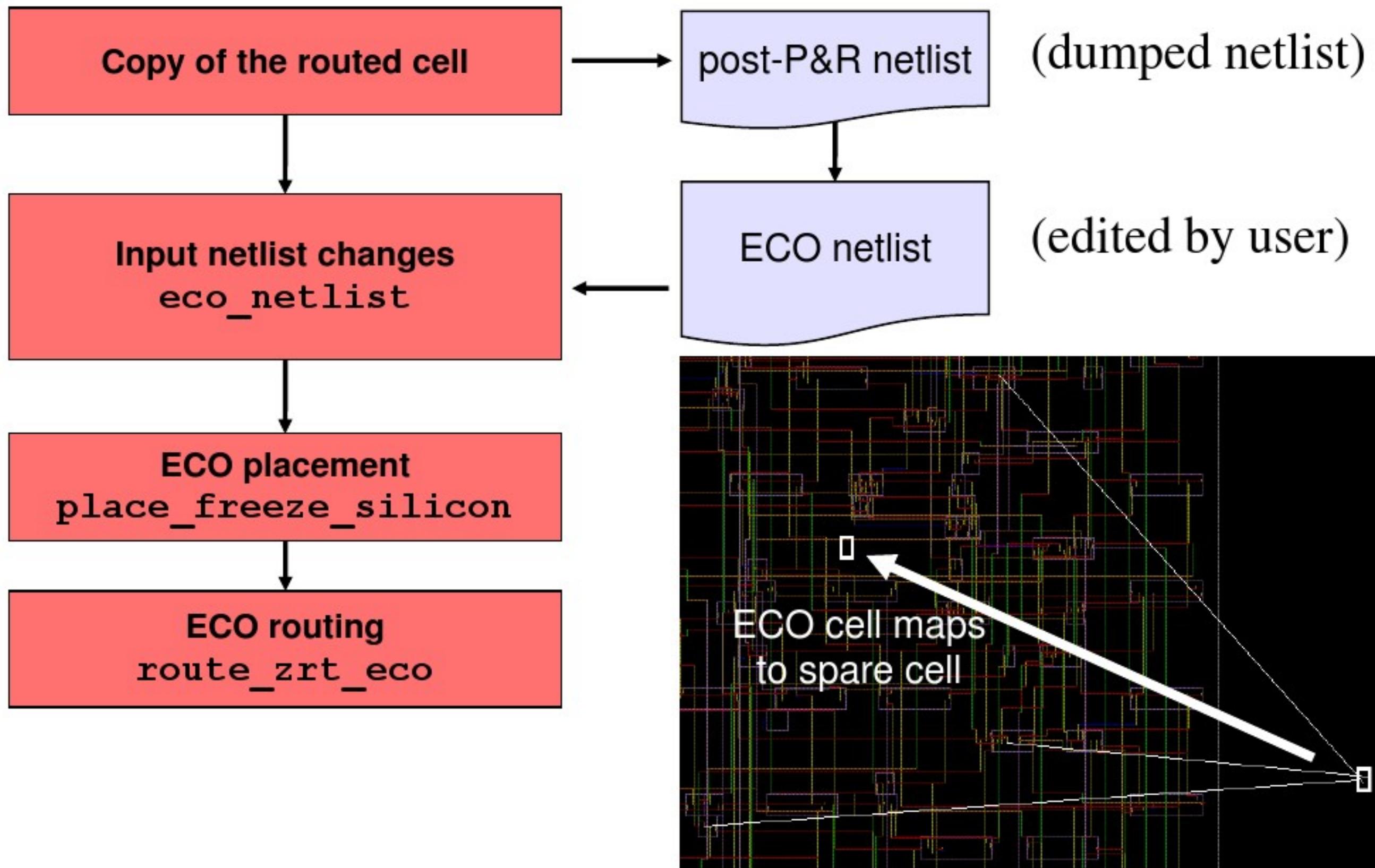
Freeze Silicon ECO Requires Spare Cells



Protecting Spare Cell

- Spare cells are *don't_touch* so ICC optimizations won't remove the unconnected cells
- Set the spare cells to SOFT FIXED once the spare cells are distributed
 - Use **set_attribute** to set the spare cells to SOFT FIXED
- Why set the spare cells to SOFT FIXED?
 - Detailed placer may fail if there are too many fixed cells
 - The soft-fixed attribute prevents incremental coarse placement from moving spare cells
 - The soft fixed cells can still be moved slightly and legalized by CTS and Routing optimizations

Freeze Silicon ECO: Metal Change Only



ECO Route: route_zrt_eco

```
route_zrt_eco
  -nets collection_of_nets
  -utilize_dangling_wires true | false
  -open_net_driven
  -max_detail_route_iterations int
```

- By default, the basic ECO routing command
 - Utilizes dangling wires
 - Runs global routing to connect broken nets
 - Runs track assignment to assign global wires
 - Runs detail routing to fix DRC violations
- Use ECO routing command for netlist changes or after manual changes

Unit Summary

You should now be able to:

- Explain all common Routing Operations
- Perform pre-routing checks and setup
- Route and optimize the design using *route_opt*
- Perform functional ECOs

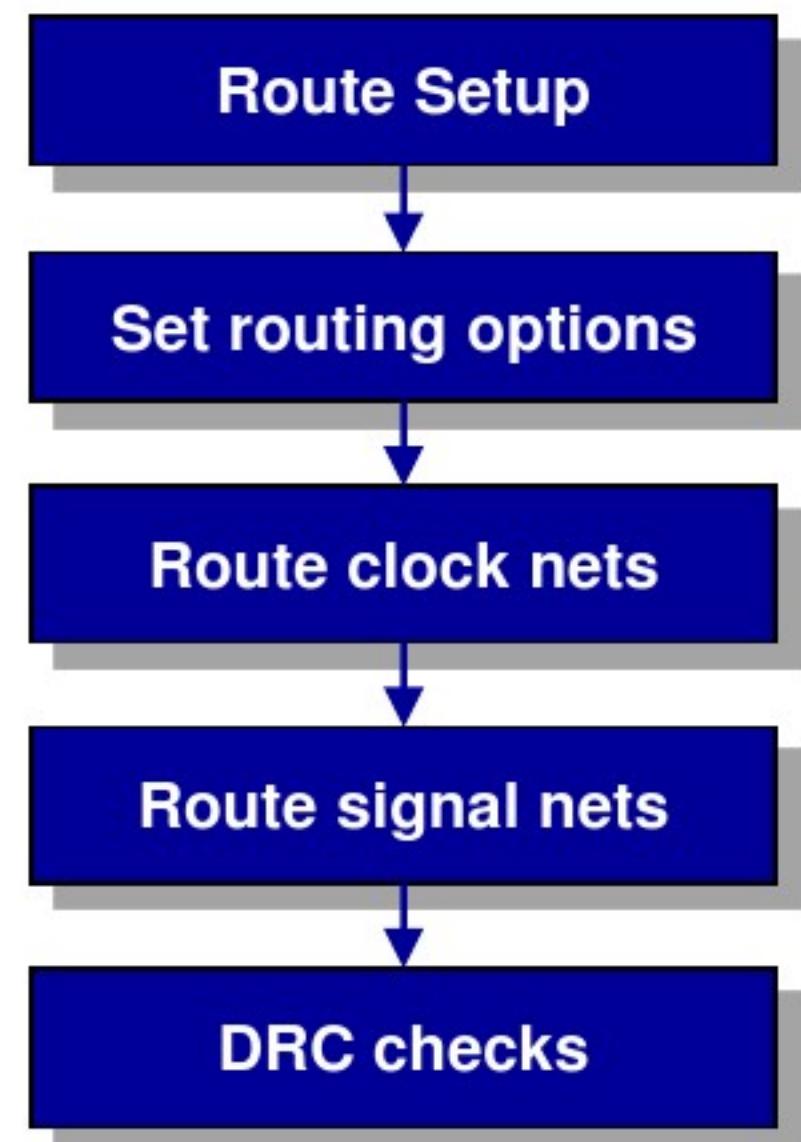


Lab 5: Routing



60 minutes

- Perform routing and related optimizations



Appendix

- A. Getting more information on Zroute**
- B. Zroute Versus Classic Router**
- C. Multi-threaded routing**
- D. Zroute support for pre-route engines**
- E. Saving intermediate cells**

A. More information on Zroute

- **External Zroute Web Page:**
 - <https://solvnet.synopsys.com/zroute>
- **SolvNet:**
 - Zroute SolvNet articles for version B-2008.09 and later
 - Search: “Zroute”
- **Parameter Translation Guide**
 - SolvNet 024478
- **User guide:**
 - Zroute is included in the IC Compiler documentation set
- **Man pages:**
 - Standard man pages and help

B. Classic Router Versus Zroute Commands

- Commands do not match 1:1 due to different routing architecture

Zroute	Classic router
route_zrt_global	route_global
route_zrt_track	route_track
route_zrt_detail	route_detail
route_zrt_detail -incremental true	route_search_repair
verify_zrt_route	verify_route
route_zrt_auto	route_auto
route_zrt_eco*	route_eco
insert_zrt_redundant_vias	insert_redundant_vias
n/a	optimize_wire_via
extract_zrt_hier_antenna_property	extract_hier_antenna_property

*Zroute must use eco route if there are opens!

B. Introducing Zroute



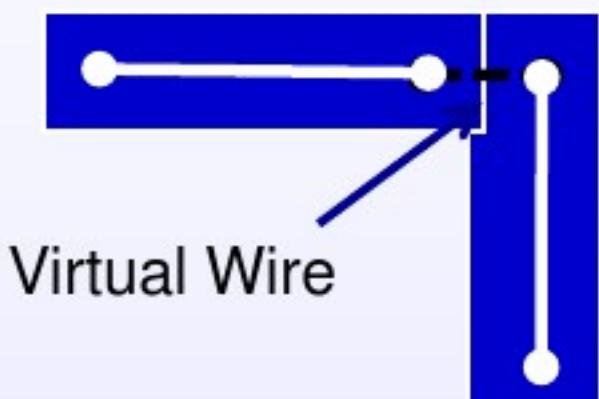
- 1 State of the art routing technology
- 2 Concurrent DFM optimizations
- 3 Multi-threaded throughout



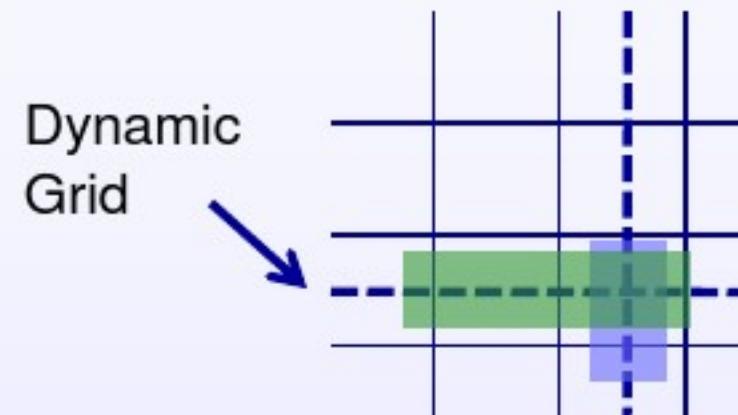
10X Speed-up, higher QoR, better manufacturability

B. State of the Art Routing Technology

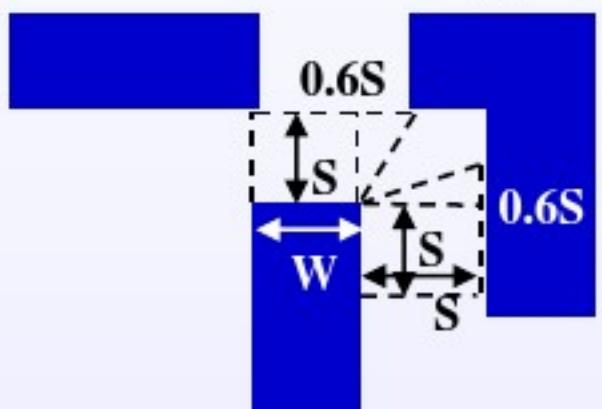
Realistic Connectivity



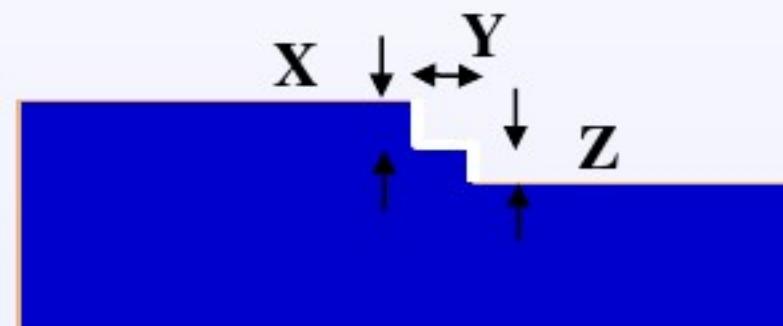
Dynamic Maze Grid



Advanced Design Rules



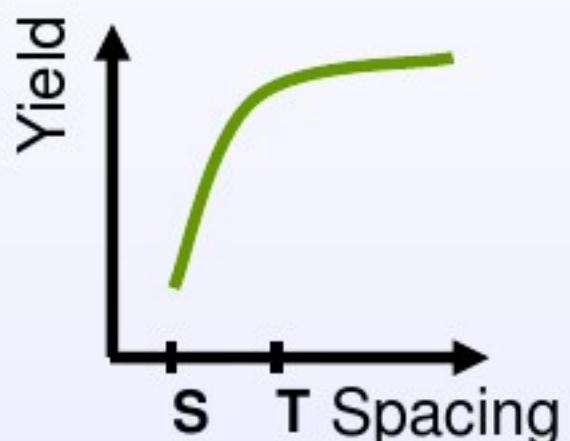
Polygon Manager



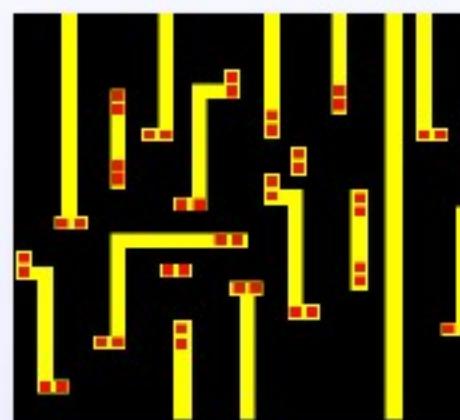
Faster runtimes, improved design closure

B. Concurrent DFM Optimizations

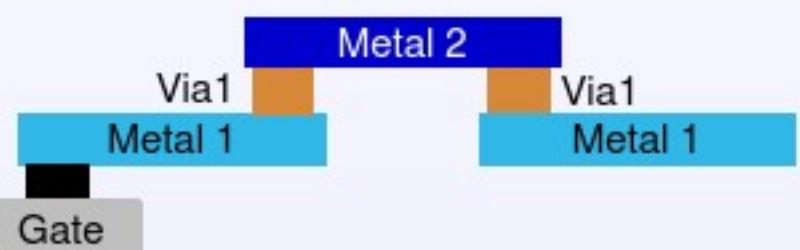
Soft Rules



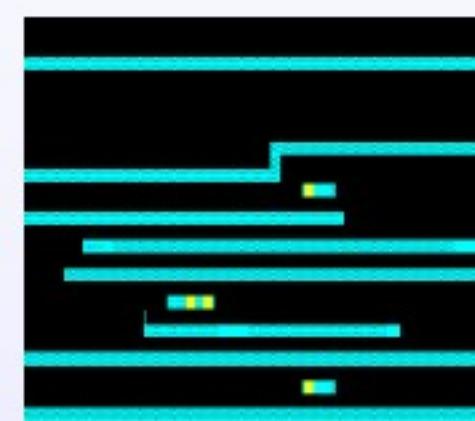
Redundant Vias



Antennas



Wire Spreading/Widening



Faster runtimes, improved manufacturability

B. Multi-threaded Throughout

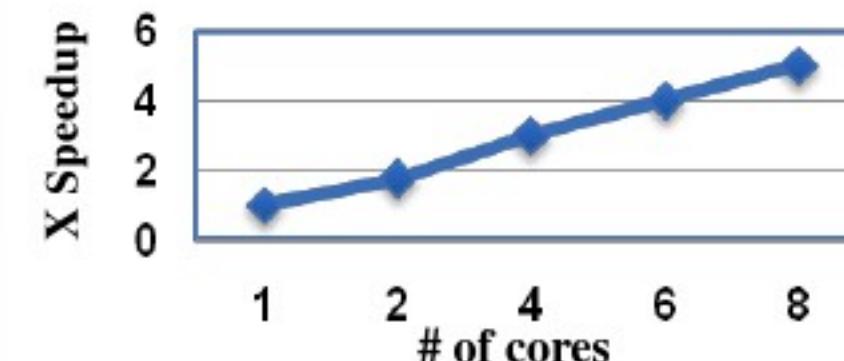
All Routing Steps

Global Routing

Track Assignment

Detail Route

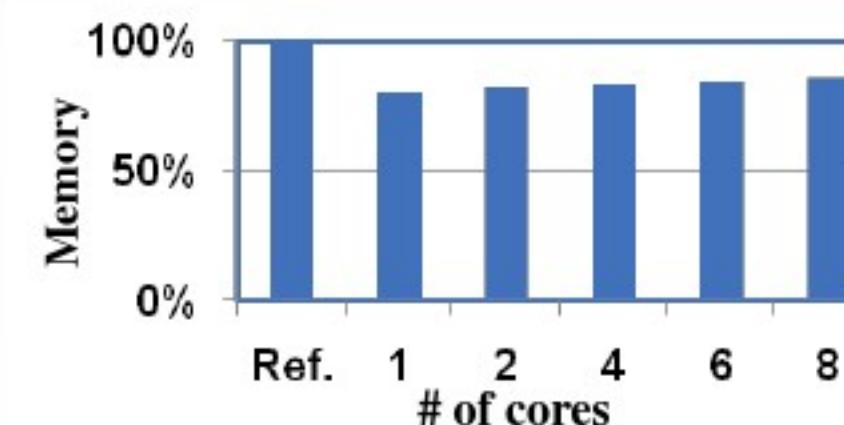
Scalability



Transparent

Native
Ready to go
Simple set-up

Larger Capacity

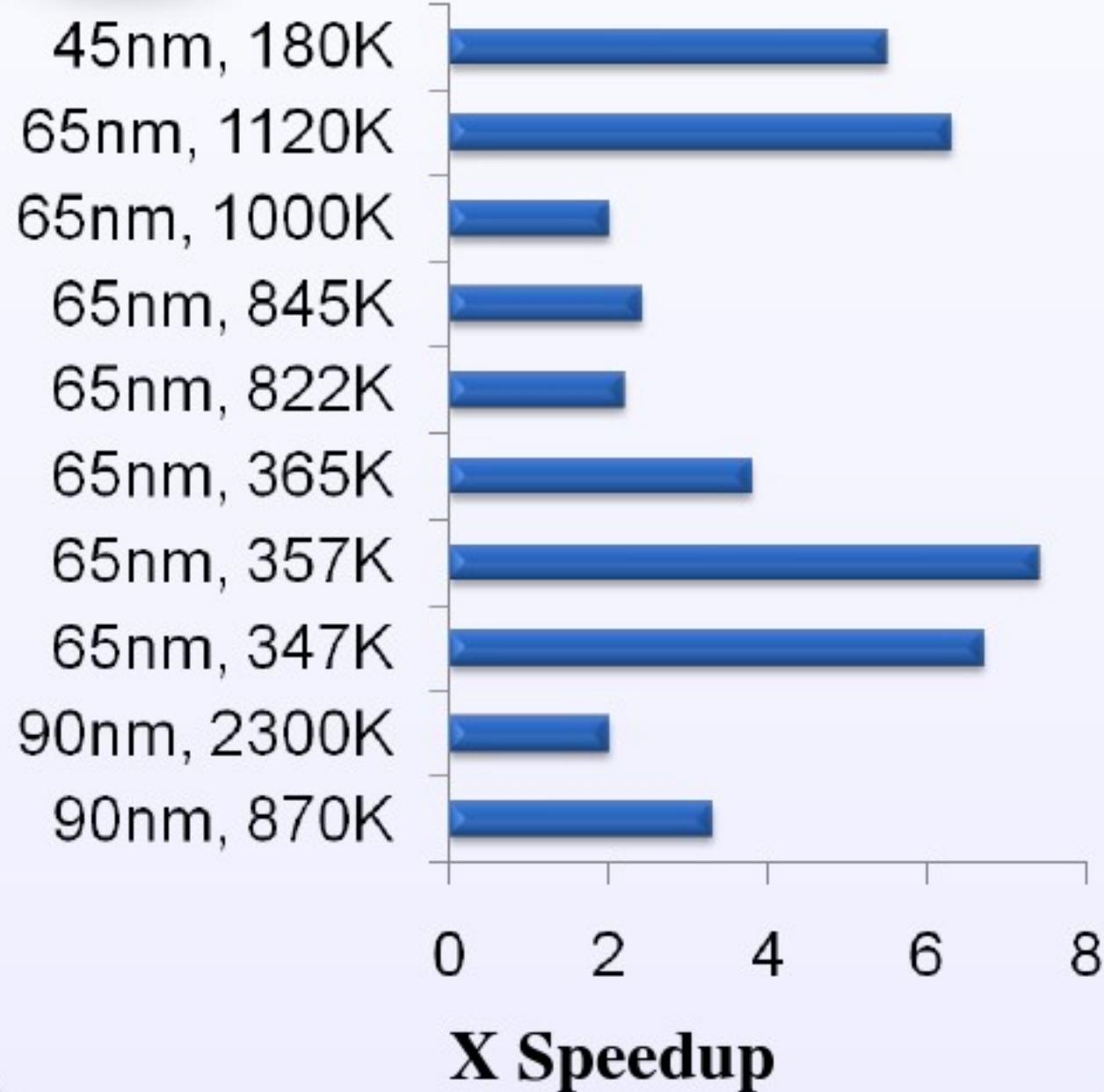


Faster runtimes, near linear scalability

B. 10X Speed-Up On Mainstream Hardware

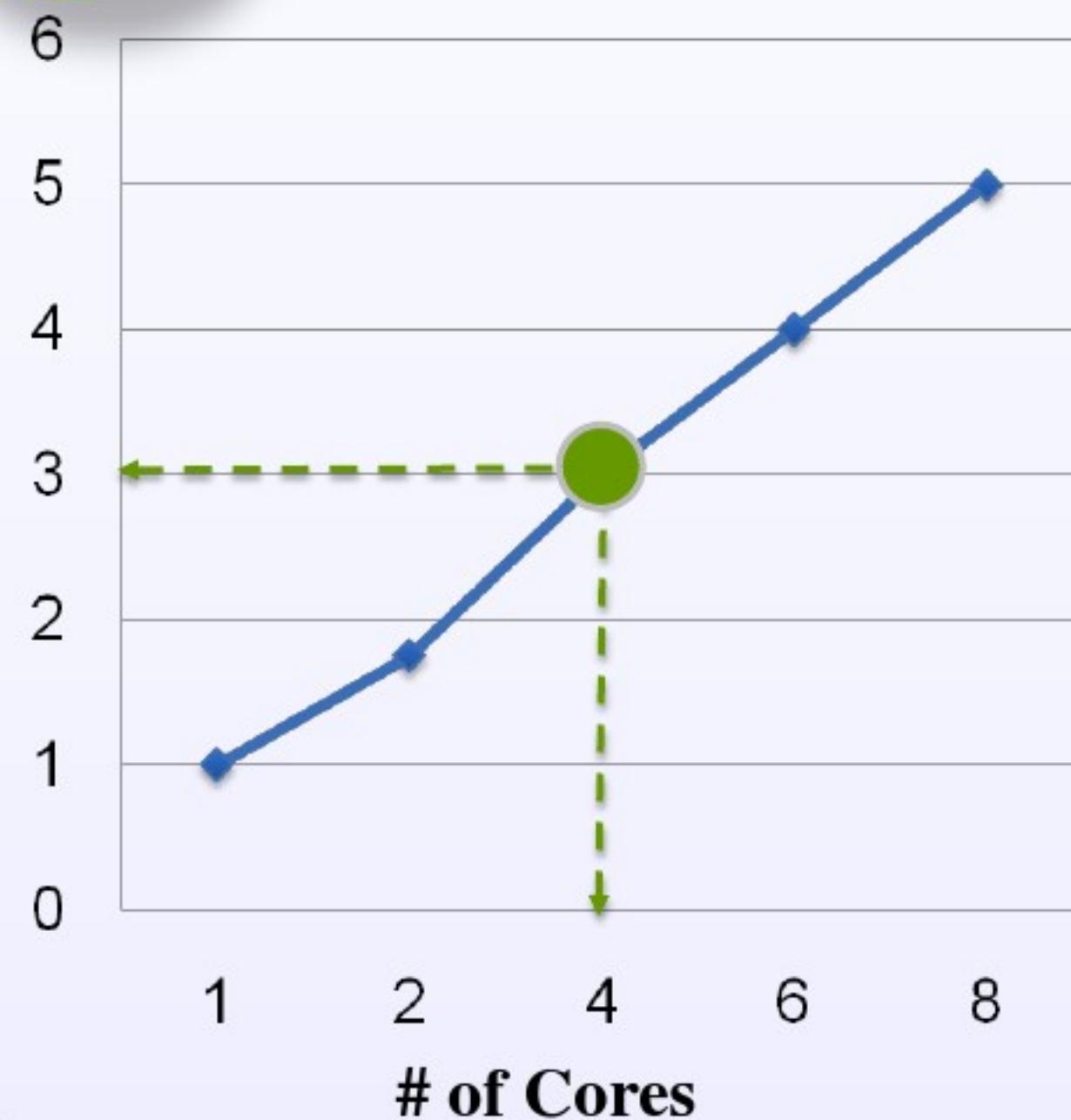
3-4X

Single-threaded



3X

Multi-threaded



B. Less Wire, fewer Vias

1-2%

10-15%

Wire Length

40nm, 225K



40nm, 60K

45nm, 170K

65nm, 845K

65nm, 593K

65nm, 545K

65nm, 356K

Improvement %

of Total Vias

45nm, 374K

65nm, 845K

65nm, 593K

65nm, 545K

65nm, 356K

80nm, 2777K

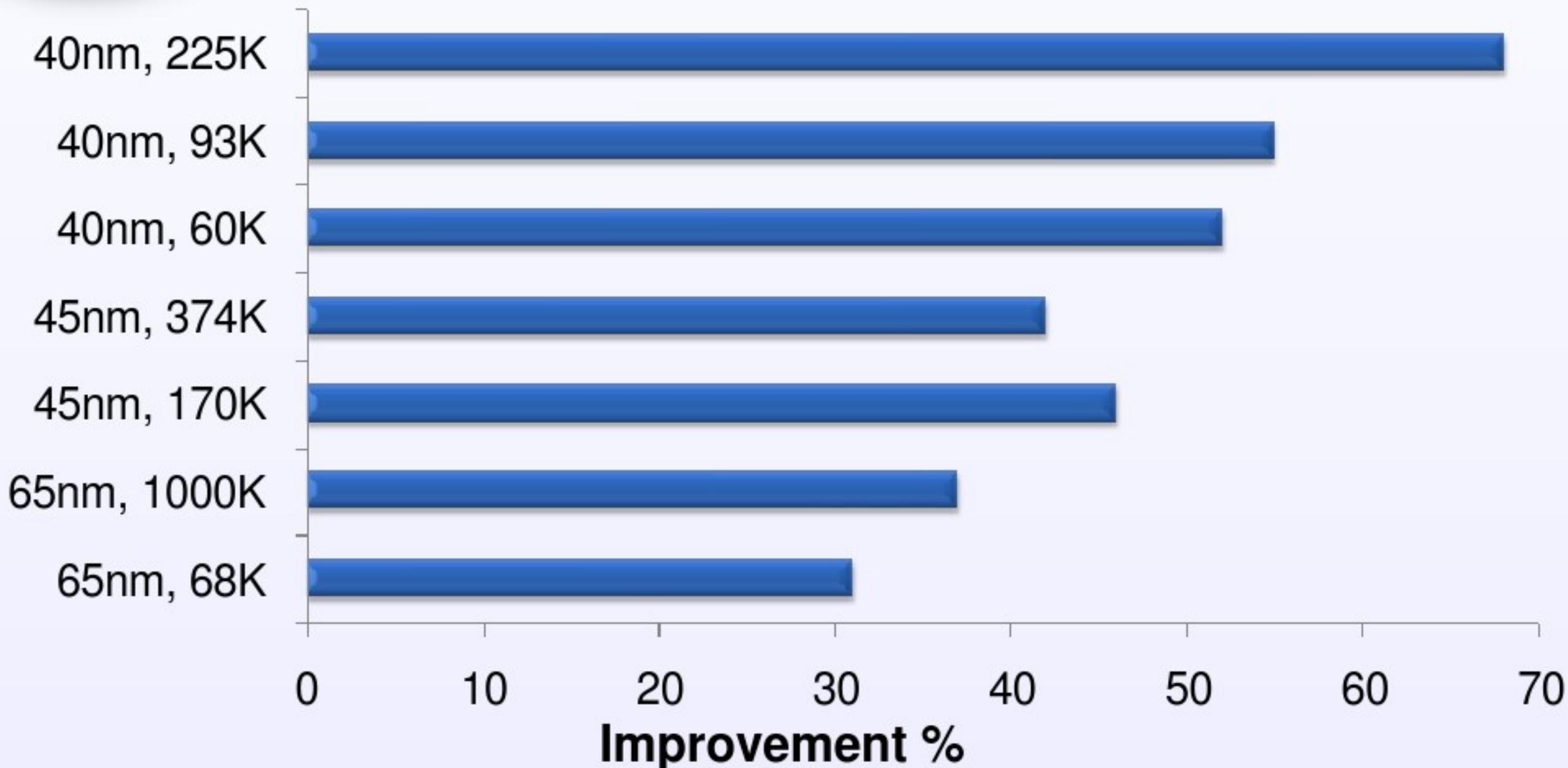
90nm, 2313K

Improvement %

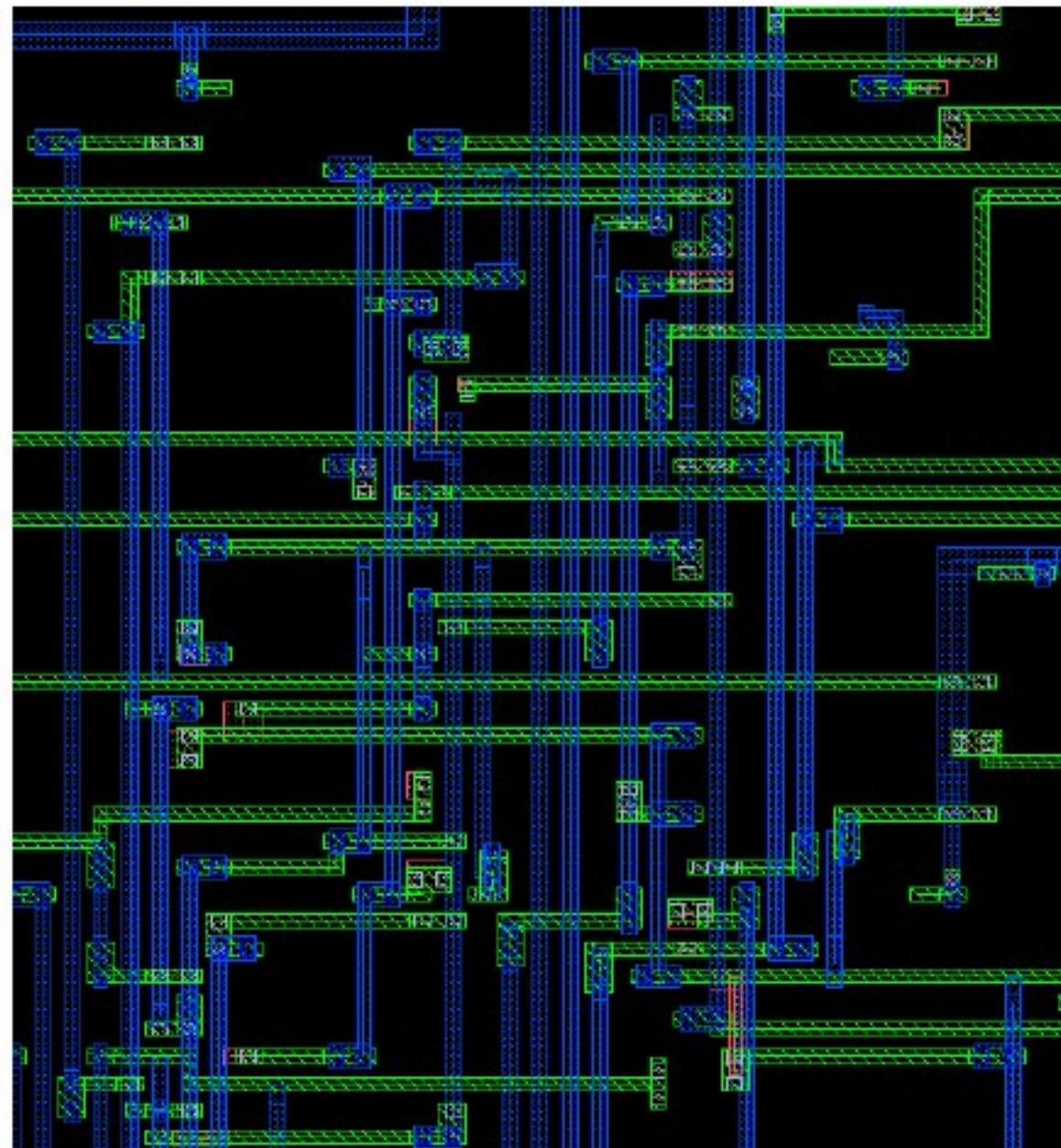
B. Fewer Non-Optimized (Single) Vias

30-50%

Remaining Non-Optimized Single Vias



B. Fewer Jogs/Notches, Better Lithography



Litho-Friendly Routing

No Scenic Routes



Fewer Jogs



Fewer Notches



C. Multi-Threaded Routing

- **Multi-threaded routing is supported – Distributed not**
 - Multi-threading: Multiple CPUs performing parallel algorithms on the same machine using a single process memory
 - Distributed: Partition design and send each partition to a separate CPU with separate memory image of routing task
- **Designed from the ground up as a multi-threaded router**
 - Most route engines run multi-threaded tasks
 - Great threadability – close to upper bounds without QoR loss
- **To enable multi-threading**

```
set_host_options -max_cores 2
```

C. Optimal Threading

- Maximum threads should equal the number of CPUs × the number of cores per CPU
 - Different IT departments count CPUs/slots differently, and different systems like LSF or GRD might interpret this differently
- Over-threading does not improve performance, and can degrade performance
- Using maximum threads and sharing resources with other processes might result in speedup (but not as much as expected) as long as shared resources do not start swapping memory

C. Finding Number of Cores

- Try to use **cpuinfo**

```
if {[file exists /proc/cpuinfo]} {  
    sh grep MHz /proc/cpuinfo}
```

- Example Report: each core reported once (4 core shown)

cpu MHz	: 2394.245

C. Split or Child Process

- **Splitting the process can lower peak memory used**
- **Zroute splits the process adaptively to reduce runtime.**
Process splitting is triggered if either of the following conditions are met:
 - The peak memory of the IC Compiler session is greater than 3 GB
 - The number of nets in the design is greater than 300,000
- **To control process splitting based on the number of nets in the design, set**

```
set_route_zrt_common_options  
    -child_process_net_threshold N
```

- **To turn off the splitting process by Zroute (and the classic router), use**

```
set_app_var physopt_enable_router_process false
```

D. Placer Congestion Support

- IC Compiler placer supports using global route congestion using Zroute

```
##Set Zroute mode  
set_route_mode_options -zroute true  
  
##Enable global router in place_opt  
set_app_var placer_enable_enhanced_router true  
  
##Run place_opt with congestion  
place_opt -congestion
```

-congestion option is needed

E. Saving Intermediate Cells (1)

- User can specify cells to be saved after specific iterations using `set_route_zrt_detail_options`, for example the following saves after iteration 1, 5, and 10:

- `-save_after_iterations` : {1 5 10}

- The name can be prefixed using:

- `-save_cell_prefix` : INIT_ZRT

- Option is persistent, so use the following to prevent save design during detail route portion of ECO routing:

```
set_route_zrt_detail_options -save_after_iterations ""
```

E. Saving Intermediate Cells (2)

- **Full route_opt (routing & post route optimizations together) automatically saves the design after two iterations (iteration 1) and before postroute optimization**
 - This allows you to view routing DRCs before route_opt finishes
 - -save_after_iterations : {1}
 - -save_cell_prefix : cell_name_INIT_RT
- **For route_opt -initial_route_only use the Zroute detail route options directly**
- **route_opt also has checkpointing, but it saves the above and other additional cells (see man page)**

```
set_app_var routeopt_checkpoint true
```