


# **IC Compiler 2010.03 Incremental Training**

## **Routing:**

**Split Net Support,  
Incremental Shielding,  
Redundant Via storage, and  
Controls for Diode Insertion**

# Agenda

- 
- Split Net Support for Preserving Routing Topology in Zroute
  - Incremental Shielding Flow
  - Persistent Storage of Redundant Via Mapping
  - User Controls for Concurrent Diode Insertion

# Split Net Support for Preserving Routing Topology in Zroute

- Overview

- When ECO routing is performed to complete the final connections in cases like buffer insertion on nets, Zroute might not follow the original routing topology.
- We now support a methodology to preserve the original routing topology in the ECO stage.

- UI

```
split_zrt_net -from_net {$long_net}  
-to_net {$long_net $n1 $n2 $n3}  
-cells {$c1 $c2 $c3}  
-breakpoints {{ix iy iz} {jx jy jz} {kx ky kz}}  
-max_breakpoint_to_cell_distance $dist
```

# Split Net Support for Preserving Routing Topology in Zroute

- User Benefit
  - Automatically breaks existing routing into parts and changes the net attributes associated with those routing shapes
- Flow Recommendation
  - Prerequisites
    - Net names to be split, repeater type to be inserted, repeater distance, and breakpoint intentions of net splitting
  - Manual `split_zrt_net` flow
    - `split_zrt_net -from_nets {REG_RAM_C_dout\[1\]} \`  
`-to_nets {eco_net} \`  
`-cells {eco_cell} \`  
`-breakpoints {{718960 494440 26}}`

# Split Net Support for Preserving Routing Topology in Zroute

- Flow Recommendation (cont.)

- Batch `split_zrt_net` flow

- `insert_buffer` used with `-on_route` generates `split_zrt_net_batch.tcl` batch file containing the required `split_zrt_net` information

- Example: `insert_buffer $net bufbd1 -on_route \`  
`-repeater_distance 100 \`  
`-new_cell_names rpt_insert \`  
`-new_net_names rpt_insert`

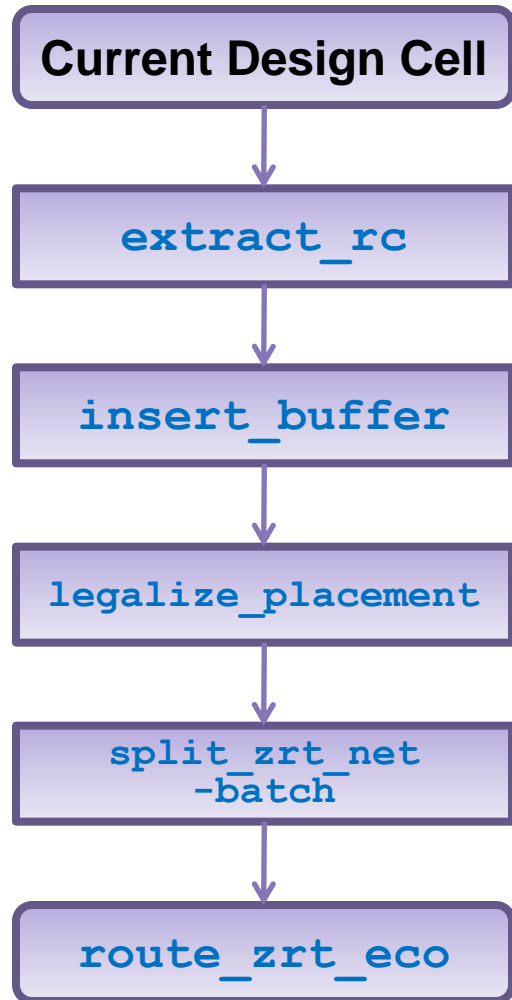
- `split_zrt_net -batch`

- Executes `split_zrt_net_batch.tcl` batch file generated by `insert_buffer` command

- Where in the flow can this feature be used

- After buffer insertion on bus nets

# split\_zrt\_net Flow




- Open design cell
- Perform RC extraction
  - Routed nets
  - Unrouted nets
- Insert buffer
  - Insert buffers and create `split_zrt_net.tcl` batch file
    - `-on_route` required
- Perform net splitting
  - Batch file
  - Manually created file
- Perform ECO routing

# Split Net Support for Preserving Routing Topology in Zroute

- Limitations
  - split\_zrt\_net\_batch.tcl file is not created correctly when you use `insert_buffer -location`
  - This will be fixed in an upcoming release

# Agenda

- Split Net Support for Preserving Routing Topology in Zroute
-  • Incremental Shielding Flow
- Persistent Storage of Redundant Via Mapping
- User Controls for Concurrent Diode Insertion

# Incremental Shielding Flow

- Overview

- The incremental unshielding and reshielding capability applies only to nets with a valid shielding net association that was created by `create_zrt_shield`.
- The goal of the incremental shielding flow is to provide an automatic way to unshield and reshield modified shielded nets.
- In 2009.06, incremental shielding support was limited to shielded nets with routing changes made by Zroute.
- The enhanced incremental shielding flow includes shielded nets modified outside of the Zroute environment, such cell sizing, buffer insertion, ECO, and route editing

# Incremental Shielding Flow

- Old UI
  - `set_route_zrt_detail_options`
  - `-reshield_rerouted_nets` is obsolete in 2010.03 and is replaced with `set_route_zrt_common_options`
  - `-reshield_modified_nets`
- New UI
  - `set_route_zrt_common_options`
  - `-reshield_modified_nets` automatically unshields or reshields modified shielded nets

# Incremental Shielding Flow

- Usage

```
- set_route_zrt_common_options \  
  -reshield_modified_nets  
  (off|unshield|reshield) Default:off
```

off: leaves shielding as is

unshield: removes shielding from modified shielded nets

reshield: removes shielding from modified shielded nets and then creates new shielding based on the new topology

# Incremental Shielding Flow

- Usage (Cont.)
  - Incremental shielding is automatically launched when any IC Compiler command that calls the Zroute detail router detects modified shielded nets, with the exception of `route_zrt_with_route_guidance`
  - If the nets specified in `create_zrt_shield -nets` do not include the modified shielded nets, the `reshield_modified_nets` directive is followed

# Incremental Shielding Flow

- Incremental Shielding in Flow

**CTS & Clock Routing**  
*(clock\_opt & route\_zrt\_group)*

**Clock Net Shielding**  
*(create\_zrt\_shield)*

**Signal Routing**  
*(route\_opt -skip\_initial\_route)*

**Postroute Optimization**  
*(route\_opt -incremental)*

**Netlist ECO & ECO Routing**  
*(eco\_netlist & route\_zrt\_eco)*

**Manual Timing / DRC Fixing**

**ECO Routing / Zroute DRC Fixing**  
*(route\_zrt\_eco & route\_zrt\_detail)*

Can be applied in different routing stages depending on the design needs

```
set_route_zrt_common_options  
-reshield_modified_nets  
reshield | unshield | off
```

Automatically unshield or reshield nets modified within the Zroute environment

Automatically unshield or reshield nets modified outside of the Zroute environment

- Net deleted from the netlist
- Net split by buffer insertion
- Net modified by routing editing
- Net modified by rerouting
- Net's nondefault routing rule modified

2010.03

# Incremental Shielding Flow


- User Benefits

- Full unshielding and reshielding support for shielded nets with any kind of routing or netlist change in IC Compiler
- Incremental shielding is done concurrently during routing

- Limitations

- Does not support routing or netlist changes made by other tools through `read_def`

# Agenda

- Split Net Support for Preserving Routing Topology in Zroute
- Incremental Shielding Flow
-  • Persistent Storage of Redundant Via Mapping
- User Controls for Concurrent Diode Insertion

# Persistent Storage of Redundant Via Mapping

- Overview
  - The `define_zrt_redundant_vias` command was enhanced to preserve the redundant via definitions in a design


- UI

```
define_zrt_redundant_vias
[-from_via {list_of_vias}]
[-to_via {list_of_vias}]
[-to_via_x_size {list_of_contact_numbers}]
[-to_via_y_size {list_of_contact_numbers}]
[-to_via_weights {list_of_weights}]
```

# Persistent Storage of Redundant-Via-Mapping

- User Benefit
  - Preserves the redundant via mapping file throughout the flow
  - The redundant via mapping applies during any routing or postroute optimization stage
- Flow recommendation
  - Define the Zroute redundant via map
  - Save to the Milkyway design to preserve the mapping
  - Start place and route or incremental routing

# Agenda

- Split Net Support for Preserving Routing Topology in Zroute
- Incremental Shielding Flow
- Persistent Storage of Redundant Via Mapping
-  • User Controls for Concurrent Diode Insertion

# User Controls for Concurrent Diode Insertion

- Overview
  - Prior to version 2010.03, Zroute fixed antenna violations by layer hopping or splitting followed by diode insertion. Diode insertion alone was not supported.
  - Zroute now supports the option of antenna fixing by diode insertion only and no layer hopping.

# User Controls for Concurrent Diode Insertion

- UI
  - A new Boolean option
    - `hop_layers_to_fix_antenna` is added to the `set_route_zrt_detail_options` command. Its default value is `true` meaning that layer hopping is still done by default to fix antenna violations.
  - To fix antenna violations by using only diode insertion, set the following options, then run Zroute:

```
set_route_zrt_detail_options \  
-hop_layers_to_fix_antenna false \  
-insert_diodes_during_routing true
```
  - Zroute will do antenna analysis and will fix the antenna violations by inserting diodes only

# Summary of Routing Enhancements

## Module 2

- How you can preserve the original routing topology in the ECO stage with `split_zrt_net`
- Incremental unshielding / re-shielding flow for changes made in IC Compiler
- You can now preserve the redundant via definitions in a design with the `define_zrt_redundant_vias` command
- You can fix antenna violations by using only diode insertion

# SYNOPSYS®

Predictable Success