Innovus ECO

ecoDesign flow

some limitation when dealing with gate array (or gate array filler) cells and low-power design

ecoDefIn flow

addresses the limitation of ecoDesign and gives users more control.

analyze_eco -> apply_patch -> optimize_patch ->write_eco_design

ecoSwapSpareCell

[-help]

instName

spareCellInstName

[-keepScan]

[-noScanTrace]

[-preservePin {list_of_spare_cell_pins}]

[-suffix string]

Swaps an unplaced or placed cell with a cell in the spare cell list.

You can specify this command only after you have run specifySpareGate.

This command can swap cells of the same cell type (same footprint and identical pin name). Cells of different sizes can also be swapped. The spare cell is connected according to the specified **global net connections**.

1) If the cell to be swapped is unplaced, it is mapped to the spare cell. **instName** is **deleted**, and its connection is transferred to the spare cell.

2) If the cell to be swapped is placed, it is swapped with the spare cell and is renamed to instNameSuffix if the -suffix option is used. If a suffix is not specified, the instName cell is renamed to spareCellInstName. The instName cell's connections are transferred to spareCellInstName. The input of **instName** is **tielo**, based on the global connection definition.

e.g.

```
#######Spare Gate Mapping file Generated by Conformal ECO Designer###########
ecoSwapSpareCell pixel_generator/vtgen/ver_gen/ECO2inst_1 spare_10/spr_gate025
ecoSwapSpareCell pixel_generator/vtgen/ver_gen/ECO2inst_2 spare_10/spr_gate021
ecoSwapSpareCell pixel_generator/vtgen/ver_gen/ECO2inst_3 spare_10/spr_gate023
```

addInst

[-help]

[-dontSnapToPlacementGrid]

[-moduleBased verilogModule]

[-physical]

-cell cellName

-inst instName

[-loc {x y} [-ori {R0 | R90 | R180 | R270 | MX | MX90 | MY | MY90}]]

[-place_status placementStatus]

Specifies the placement status of the added instance.

You can specify the following as the placement status:

placed: The status of the added instance is placed.

fixed: The status of the added instance is fixed.

unplaced: The status of the added instance is unplaced.

Default: placed

Note: This option is used along with the -loc option. It should not be used along with the -moduleBased option.

e.g.

```
#######Gate Array Mapping file Generated by Conformal ECO Designer###########
addInst -cell GOR2D2BWPLVT -inst pixel_generator/vtgen/ver_gen/ECO2inst_1 -loc 18.620 87.500 -ori R0
addInst -cell GND2D1BWPLVT -inst pixel_generator/vtgen/ver_gen/ECO2inst_2 -loc 12.320 84.980 -ori R0
addInst -cell GND2D1BWPLVT -inst pixel_generator/vtgen/ver_gen/ECO2inst_3 -loc 15.820 84.980 -ori R0
addInst -cell GND2D1BWPLVT -inst pixel_generator/vtgen/ver_gen/ECO2inst_4 -loc 16.520 79.940 -ori R0
```

ecoDesign

Takes an Innovus database and a modified netlist as input and performs ECO operations. It restores the design, examines the changes in the new netlist, and automatically implements the required changes with ecoPlace and ecoRoute.

```
sessionDirectoroy
design
newNetList
[-def defFile]
[-noEcoPlace]
[-noEcoRoute]
[-postMask]
[-fillerPrefix prefix | -noDeleteFiller]
[-modifyOnlyLayers MLb:MLt]
[-spareCells spareCellName]
[-suffix suffix]
[-tieCell cellNameList]
[-useGACells GACoreSite | -useGAFillerCells list_of_filler_cells ]
```

ecoPlace

setNanoRouteMode

ecoRoute

defComp

analyze_eco

Analyzes the ECO change in the Revised root module comparing to the Golden root module. The logic change is written to the **specified patch file,** which contains the Verilog module with the port names corresponding to the nets in the Golden design.

Only the logic cone under NONEQ points are analyzed by the command.

<patch_filename>

Specifies the name of the patch file.

-REPlace

Replaces the existing file.

-HIERarchical [-MODule <module name>

Analyzes all the nonequivalent modules reported by COMPARE ECO HIERARCHY.

If COMPARE ECO HIERARCHY has not been executed, the tool calls it implicitly.

If the patch file name contains %s, it will be replaced by the module name in each ECO hierarchy.

Otherwise, all the patches will output to the specified patch file.

-ECOPIN_dofile <FILE_NAME>

Creates a dofile that adds ECO pins when there are extra pins in the Revised design (as compared to Golden design) and deletes ECO pins when there are extra pins in the Golden design (as compared to Revised design).

apply_patch

Applies the ECO change specified in the patch module, generated by the ANALYZE ECO command, to the module under ECO. The patched module can be written out with the WRITE DESIGN command.

The patch generated by the ANALYZE ECO command can contain unmapped primitives.

-AUTO

Automatically reads in and applies all patches that were created with the ANALYZE ECO command in the current session.

-KEEPHierarchy

Specifies that the ECO changes will be put in a submodule. This is the default.

-KEEPFREED

Retains all freed instances and leaves input pins connected. Freed instance will **not be re-used** for mapping the patch.

-TIEFREED0

Retains all freed instances and applies a value of '0' (tie low) to the input pins of any freed instance.

-TIEFREED1

Retains all freed instances and applies a value of '1' (tie high) to the input pins of any freed instance.

-FREESCAN

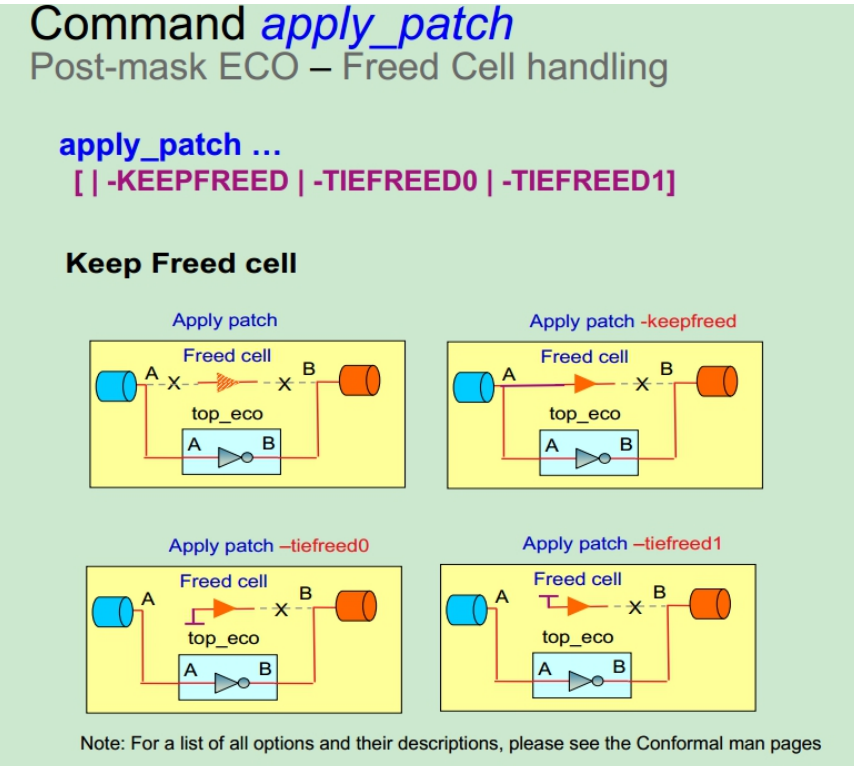
Free up the DFFs reachable only by scan chain.

-Golden

Applies to the Golden design. This is the default.

-Revised

Applies to the Revised design.



Command *apply_patch*

Post-mask ECO – Freed Cell handling

apply_patch …

[| -KEEPFREED | -TIEFREED0 | -TIEFREED1]

Keep Freed cell

Apply patch

Apply patch -keepfreed

Apply patch –tiefreed0

Apply patch –tiefreed1

Note: For a list of all options and their descriptions, please see the Conformal man pages

optimize_patch

Writes out a Genus script (cfm_eco_rc.tcl) in the working directory that will optimize the patches and execute the script.

After OPTIMIZE PATCH successfully completes, the ECO design will be in memory and can be written out. The optimized patches will be in the working directory.

Before running this command, you must run the APPLY PATCH -keephierarchy command.

report_eco_changes

Reports the ECO changes. To report the ECO change with this command, the ECOs must have been applied with the APPLY PATCH or OPTIMIZE ~~DESIGN~~ PATCH command.

call "report_eco_changes" after "apply_patch or optimize_patch"

-SUMmary

Shows a summary only. This is the default.

report_eco_changes == report_eco_changes -summary

-INNovus

Generates a script for the Innovus Implementation System

-FILE <filename>

Redirects the output file.

-SCRipt

Reports ECO changes and then generates a generic script that you can modify to use in different tools. For information on the commands within this script, refer to "Generic Script Format" in the Conformal ECO User Guide.

-TIE1 <cell_name>

Use the specified cell as the tie high cell.

-TIE0 <cell_name>

Use the specified cell as the tie low cell.

write_eco_design

Writes out the ECO netlist (note: does not write out non-ECO files) and attempts to reduce the number of text differences between the original netlist file and the ECO netlist file. This will reduce the number of differences reported by the UNIX diff command between the two files.

e.g.

```
TCL_SETUP> write_eco_design -newfile %s.hier.G3 -replace
```

Notes:

(a) %s retains the original name and %d allows numerical assignment to ECO instances and nets.

(b) **G3 netlist is stored in the same location as G1** ($WSDIR/LAB1/PNR1/)

add_spare_cell

Adds the **spare cells** or **freed cells** as the available cells for the OPTIMIZE PATCH command.
When using OPTIMIZE PATCH with the -usespare option, the spare cell count for each cell type is considered.
When using OPTIMIZE PATCH with the -def option, the spare cell instances are considered for **location-aware** optimization.

-DEFfile <filename>

Specifies the DEF filename (compressed gzip format is alo supported). This searches for the spare cell in the DEF file. By default, the spare cell is searched for in current hierarchy.

-FReedcell

Specifies that freed cells will be used for mapping.

-SParecell <cell_instance_name*> ...

Specifies the spare cells to be added. This accepts wildcards.

-GAFiller <cell_instance_name*> ...

Specifies the gate array filler to be added. This accepts wildcards.

delete_spare_cell

Deletes the spare cells or freed cells that were added with the add_spare_cell command. In essence, it enables discriminating use of certain spare gates for particular ECOs

e.g.

```
// Command: delete_spare_cell -freedcell chip/smb/cmdreg_*
// Command: delete_spare_cell -sparecell chip/spare/xff*
...
// Command: report_spare_cell
```

-FReedcell

Specifies the name(s) of the freed cell(s) to be deleted. This accepts wildcards.

-SParecell

Specifies the name(s) of the spare cell(s) to be deleted. This accepts wildcards.

-GAfiller

Specifies the name of the gate array cell(s) to be deleted. This accepts wildcards.

-CELLtype

Specifies the name(s) of library cell type(s) to be deleted. This accepts wildcards.

-ALL

Deletes all the cells.

optimize_patch

Writes out a Genus script (cfm_eco_rc.tcl) in the working directory that will optimize the patches and execute the script

-USESPARE

Attempts to implement the ECO by mapping to **spare** gates. This option requires an ECO GXL license

Enables spare gate mapping and/or gate array mapping

-GAlibcell <cell_name*> ...

Specifies the **gate array library cells**. This option requires an ECO GXL license.

-AVOID <cell_name>*

Avoids the specified library cells. This accepts wildcards.

When this option is used in combination with the -USESPARE option, only the cell quantities of those cell types **not** specified by -AVOID are available for mapping.

-USE <cell_name>*

Uses the specified library cells. This accepts wildcards.

When this option is used in combination with the

-USESPARE option, only the cell quantities of those cell types specified by -USE are used for mapping.

The order that you specify the -AVOID and -USE options is significant. For example:

"-avoid * -use NAND2 INV2"

avoids all the library cell types except NAND2 and INV2. If these options are specified in the following order:

"-use NAND2 INV2 -avoid *"

then no cell types will be available for mapping.

-LEF <filename> ...

Specifies the LEF file(s) (compressed gzip format is also supported). This calls Genus physical synthesis to perform location-aware spare-gate mapping.

This option requires an ECO GXL license.

-DEF <filename> ...

Specifies the DEF file(s) (compressed gzip format is also supported). This calls Genus physical synthesis to perform location-aware spare-gate mapping.

This option requires an ECO GXL license.

-MAPscript <filename>

Writes out the location aware spare gate mapping result in the form of an SoC Encounter TCL script. This option should be used with the **-DEF** option.

This option requires an ECO GXL license.

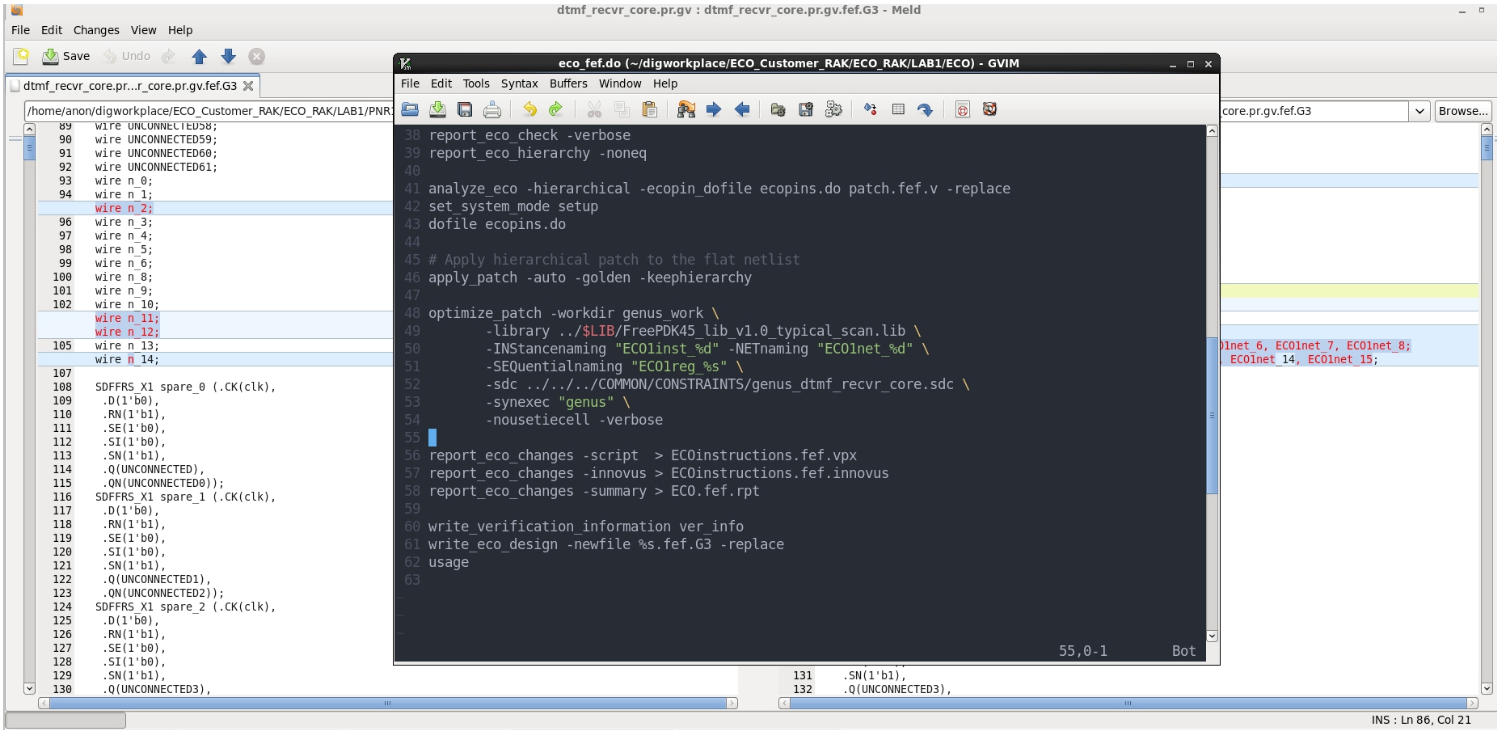
-CAPtable <filename>

Specifies the name of the CAP file. This option requires an ECO GXL license.

-QRCtech <filename>

Specifies the name of the **qrctech file**. This option requires an ECO GXL license.

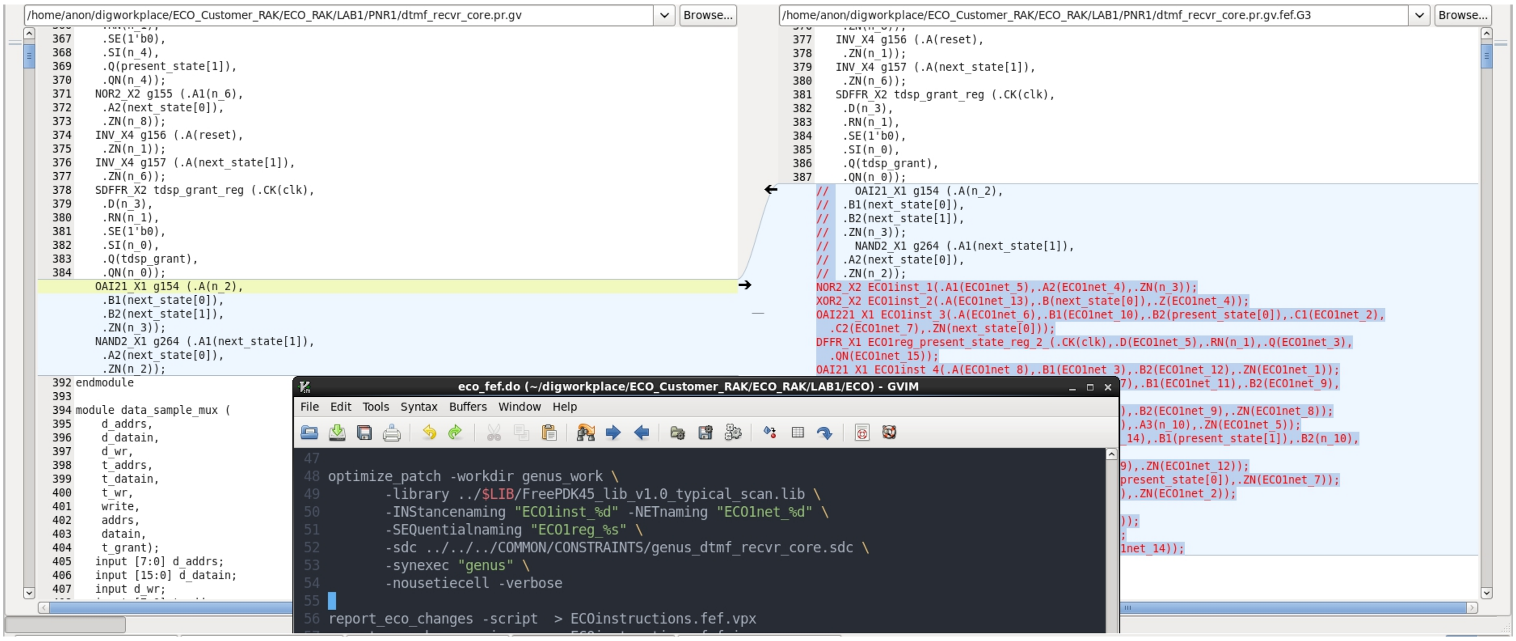
Note: qrctech files are supported in Genus, but not in RTL Compiler (RC).


dtmf_recvr_core.pr.gv : dtmf_recvr_core.pr.gv.fef.G3 - Meld
File Edit Changes View Help
Save Undo
dtmf_recvr_core.pr...r_core.pr.gv.fef.G3
/home/anon/digworkplace/ECO_Customer_RAK/ECO_RAK/LAB1/PNR1
core.pr.gv.fef.G3
Browse...
eco_fef.do (~/digworkplace/ECO_Customer_RAK/ECO_RAK/LAB1/ECO) - GVIM
File Edit Tools Syntax Buffers Window Help
55,0-1 Bot
INS : Ln 86, Col 21

```
wire UNCONNECTED58;
wire UNCONNECTED59;
wire UNCONNECTED60;
wire UNCONNECTED61;
wire n_0;
wire n_1;
wire n_2;
wire n_3;
wire n_4;
wire n_5;
wire n_6;
wire n_8;
wire n_9;
wire n_10;
wire n_11;
wire n_12;
wire n_13;
wire n_14;

SDFFRS_X1 spare_0 (.CK(clk),
 .D(1'b0),
 .RN(1'b1),
 .SE(1'b0),
 .SI(1'b0),
 .SN(1'b1),
 .Q(UNCONNECTED),
 .QN(UNCONNECTED0));
SDFFRS_X1 spare_1 (.CK(clk),
 .D(1'b0),
 .RN(1'b1),
 .SE(1'b0),
 .SI(1'b0),
 .SN(1'b1),
 .Q(UNCONNECTED1),
 .QN(UNCONNECTED2));
SDFFRS_X1 spare_2 (.CK(clk),
 .D(1'b0),
 .RN(1'b1),
 .SE(1'b0),
 .SI(1'b0),
 .SN(1'b1),
 .Q(UNCONNECTED3),
```

```
report_eco_check -verbose
report_eco_hierarchy -noneq

analyze_eco -hierarchical -ecopin_dofile ecopins.do patch.fef.v -replace
set_system_mode setup
dofile ecopins.do

# Apply hierarchical patch to the flat netlist
apply_patch -auto -golden -keephierarchy

optimize_patch -workdir genus_work \
        -library ../$LIB/FreePDK45_lib_v1.0_typical_scan.lib \
        -INStancenaming "ECO1inst_%d" -NETnaming "ECO1net_%d" \
        -SEQuentialnaming "ECO1reg_%s" \
        -sdc ../../../COMMON/CONSTRAINTS/genus_dtmf_recvr_core.sdc \
        -synexec "genus" \
        -nousetiecell -verbose

report_eco_changes -script  > ECOinstructions.fef.vpx
report_eco_changes -innovus > ECOinstructions.fef.innovus
report_eco_changes -summary > ECO.fef.rpt

write_verification_information ver_info
write_eco_design -newfile %s.fef.G3 -replace
usage
```

```
)1net_6, ECO1net_7, ECO1net_8;
, ECO1net_14, ECO1net_15;
 .SN(1'b1),
 .Q(UNCONNECTED3),
```


/home/anon/digworkplace/ECO_Customer_RAK/ECO_RAK/LAB1/PNR1/dtmf_recvr_core.pr.gv
Browse...
/home/anon/digworkplace/ECO_Customer_RAK/ECO_RAK/LAB1/PNR1/dtmf_recvr_core.pr.gv.fef.G3
Browse...
eco_fef.do (~/digworkplace/ECO_Customer_RAK/ECO_RAK/LAB1/ECO) - GVIM
File Edit Tools Syntax Buffers Window Help

```
 .SE(1'b0),
 .SI(n_4),
 .Q(present_state[1]),
 .QN(n_4));
NOR2_X2 g155 (.A1(n_6),
 .A2(next_state[0]),
 .ZN(n_8));
INV_X4 g156 (.A(reset),
 .ZN(n_1));
INV_X4 g157 (.A(next_state[1]),
 .ZN(n_6));
SDFFR_X2 tdsp_grant_reg (.CK(clk),
 .D(n_3),
 .RN(n_1),
 .SE(1'b0),
 .SI(n_0),
 .Q(tdsp_grant),
 .QN(n_0));
OAI21_X1 g154 (.A(n_2),
 .B1(next_state[0]),
 .B2(next_state[1]),
 .ZN(n_3));
NAND2_X1 g264 (.A1(next_state[1]),
 .A2(next_state[0]),
 .ZN(n_2));
endmodule

module data_sample_mux (
   d_addrs,
   d_datain,
   d_wr,
   t_addrs,
   t_datain,
   t_wr,
   write,
   addrs,
   datain,
   t_grant);
   input [7:0] d_addrs;
   input [15:0] d_datain;
   input d_wr;
```

```
INV_X4 g156 (.A(reset),
 .ZN(n_1));
INV_X4 g157 (.A(next_state[1]),
 .ZN(n_6));
SDFFR_X2 tdsp_grant_reg (.CK(clk),
 .D(n_3),
 .RN(n_1),
 .SE(1'b0),
 .SI(n_0),
 .Q(tdsp_grant),
 .QN(n_0));
//   OAI21_X1 g154 (.A(n_2),
//  .B1(next_state[0]),
//  .B2(next_state[1]),
//  .ZN(n_3));
//   NAND2_X1 g264 (.A1(next_state[1]),
//  .A2(next_state[0]),
//  .ZN(n_2));
NOR2_X2 ECO1inst_1(.A1(ECO1net_5),.A2(ECO1net_4),.ZN(n_3));
XOR2_X2 ECO1inst_2(.A(ECO1net_13),.B(next_state[0]),.Z(ECO1net_4));
OAI221_X1 ECO1inst_3(.A(ECO1net_6),.B1(ECO1net_10),.B2(present_state[0]),.C1(ECO1net_2),
  .C2(ECO1net_7),.ZN(next_state[0]));
DFFR_X1 ECO1reg_present_state_reg_2_(.CK(clk),.D(ECO1net_5),.RN(n_1),.Q(ECO1net_3),
  .QN(ECO1net_15));
OAI21_X1 ECO1inst_4(.A(ECO1net_8),.B1(ECO1net_3),.B2(ECO1net_12),.ZN(ECO1net_1));
7),.B1(ECO1net_11),.B2(ECO1net_9),

),.B2(ECO1net_9),.ZN(ECO1net_8));
),.A3(n_10),.ZN(ECO1net_5));
_14),.B1(present_state[1]),.B2(n_10),

9),.ZN(ECO1net_12));
present_state[0]),.ZN(ECO1net_7));
),.ZN(ECO1net_2));

));
;
1net_14));
```

```
optimize_patch -workdir genus_work \
        -library ../$LIB/FreePDK45_lib_v1.0_typical_scan.lib \
        -INStancenaming "ECO1inst_%d" -NETnaming "ECO1net_%d" \
        -SEQuentialnaming "ECO1reg_%s" \
        -sdc ../../../COMMON/CONSTRAINTS/genus_dtmf_recvr_core.sdc \
        -synexec "genus" \
        -nousetiecell -verbose

report_eco_changes -script  > ECOinstructions.fef.vpx
```

ECO instructions Set

To generate ECO instructions set with **native format**

// Command: optimize_patch -workdir …

// Command: report eco changes -script > ECOinstructions.vpx

ECOinstructions.vpx:

set_root_module arb

# DISCONNECT PINS : 25

disconnect_pin {present_state_reg[0]/D} ;# net:next_state[0]_43 , dir:in

disconnect_pin {p7498A/B1} ;# net:next_state[0]_43_1 , dir:in

# DEL INSTANCES : 5

delete_instance {p7181D} ;# OAI22_X1

delete_instance {p7143A} ;# OAI21_X1

```
# ADD INSTANCE : 19
add_instance {ECO1inst_1} {OAI21_X1}
add_instance {ECO1inst_2} {AND2_X2}
```

To generate ECO instructions set with Innovus format

```
// Command: optimize_patch -workdir ...
// Command: report eco changes -innovus > ECO_instructions.innovus
```

ECO_instructions.innovus:

```
# DISCONNECT PINS : 24
detachTerm -moduleBased arb {p7556A} {A2}
# DEL INSTANCES : 5
deleteInst -moduleBased arb {p7181D}
deleteInst -moduleBased arb {p7143A}
# ADD INSTANCE : 19
addInst -moduleBased arb -cell OAI21_X1 -inst {ECO1inst_1}
addInst -moduleBased arb -cell AND2_X2 -inst {ECO1inst_2}
# ADD NET : 18
addNet -moduleBased arb {ECO1net_1}
```

Understanding content and usage of ECO script files generated from "report_eco_changes -innovus" and "optimize_patch -mapscript" commands

Understanding content and usa...

command entry mode: vpxmode, tclmode

1) vpxmode

Returns Conformal to the default VPX **command entry mode**.

The commands in this reference manual are the VPX commands. Refer to the Conformal Extended Checks User Guide to learn more about Conformal Tcl commands.

When issuing this command from the Tcl command interpreter, you must type this in lowercase. For example:

```
TCL> vpxmode
```

In VPX mode, you can save report data to files using the redirection command. For example, the following command saves the gate report data to a file named gate.out:

```
SETUP> report gate -type dff > gate.out
```

2) tclmode

Switches the **command entry mode** from VPX to Tcl. VPX mode is the default command mode.

There are two types of Tcl mode commands: Native and Conformal.

To start the Conformal software in Tcl mode without executing any initialization script, run the following command at a UNIX system prompt:

```
UNIX% lec -verify -tclmode
```

In the Tcl command entry mode, you can save report data to files using the redirection command. For example, the following command saves the gate report data to a file named gate.out:

```
TCL_SETUP> report_gate -type dff > gate.out
```

system mode

set_system_mode

Switches **system modes** between the Setup mode and the Verify mode.

1) In Setup mode, you can read in the design and set all the necessary constraints and environment variables.

2) In Verify mode, the Conformal Extended Checks software runs the proof and diagnosis.

Use the REPORT ENVIRONMENT command to display the current system mode.

break

Interrupt script execution and return to the **command prompt.** Use this command in a **dofile script**. This has no effect if you type this command at the command prompt.

This command is not available if you launched the tool using the ccd-t command.

This command is valid in vpxmode, you can call “vpx break” in tclmode

write_hier_compare_dofile

Writes out a hierarchical dofile script that verifies the two hierarchical designs starting from the lower-level modules and progressing to the top root module. All modules not output to the hierarchical dofile script will be flattened into their parent modules for comparison.

<filename>

Specifies the name of the dofile script that verifies design hierarchy.

-ECO_aware

Recognizes ECO-related changes. This option recognizes ports that would otherwise be ignored for non-ECO comparisons, thus facilitating the correct comparison between the Golden and Revised design.

-ECOPIN_dofile <filename>

Writes out a dofile for adding ECO pins to the Golden design as compared to the Revised design.

-FORmat

Specifies the format (VPX or Tcl) of the commands in the generated hierarchical dofile.

Auto

Determines the format automatically, based on the mode in which the WRITE HIERARCHICAL DOFILE command was specified. This is the **default**.

VPX

VPX format. The hierarchical dofile will contain VPX commands.

TCL

Tcl format. The hierarchical dofile will contain Tcl commands.

-Usage

Executes the USAGE command after each comparison and at the end of the hierarchical comparison.

-VERBOSE

Provides additional information when writing out the hierarchical dofile script.

usage

Displays the total CPU run time and current memory use since you started Conformal.

flatten

When using this command with Conformal ECO, you should use the -nolibrary option to prevent flattening to the primitive level.

Removes all hierarchy on a specified module or for all modules in the database. If you do not specify one or all modules, Conformal flattens the root module by default. Thus, this command expands all of the gate primitive or transistor primitive devices into the cell that is being flattened.

To ensure that the hierarchy matches between the designs, it is recommended that you match the hierarchy between the Golden and Revised designs (FLATTEN -MATCHHierarchy) before you use UNIQUIFY. Note that in the ECO flow, the module and instance names should not be changed in the golden G1 netlist. Therefore, the FLATTEN and UNIQUIFY commands should only be applied to the revised G2 netlist in that particular use case.

-NOLibrary

Flattens all modules in the designs.

-MATCHHierarchy

To make the hierarchy match between the designs, this option flattens hierarchical modules in the Golden or Revised design.

-INSTancename: Flattens a module only when there is no matching module and no matching instance of the same name in the complementing design. This is the default.

-NOINSTancename: Flattens a module only if there is no matching module in the complementing design.

uniquify

Makes the specified module, which has multiple instances, unique. This command lets you remedy the "incompatible" instantiations warnings during hierarchical script generation. If Conformal does not make the modules unique, they are not included in the hierarchical dofile.

-ALL

Makes all modules, within the given defaults, in the specified design unique.

-Library

Makes all modules in designs and libraries unique. This is the **default**.

-NOLibrary

Makes all modules in designs unique.

-USE_RENaming_rules

Considers renaming rules for instances.

-Golden

The specified modules are in the Golden design. This is the default.

-Revised

The specified modules are in the Revised design.

When adding renaming rules, the software renames the Golden design instances to be same as in the Revised design, so running a subsequent UNIQUIFY command with this option will make the Golden modules that have matching instance names in the Revised design unique.

dofile

Executes a set of commands contained in a specified file. If there is an error while the Conformal software is executing the dofile script, it terminates the dofile execution and returns to the system mode prompt.

resolve

Ungroups a module in the Golden or Revised design hierarchy. Resolving or ungrouping is the process of eliminating a module and promoting its content up one level of the hierarchy.

report_environment

Displays global designs, system and lowpower option settings. The default is to report all global settings.

-LP

Reports environment related to lowpower options.

-NON_DEFAULT

Reports environment that is not the default values.

set_root_module

After you read in a design, Conformal Extended Checks treats the top module as the root module by default. You can manually reset a root module with the SET ROOT MODULE command.

Use this command to focus on specific parts of a design for verification and debugging.

Use the REPORT ENVIRONMENT command to display the settings for the design's root module.

<module_name>

This module is the root. This assignment overrides the automatic root module assignment Conformal makes when you use the READ DESIGN command.

-Golden

Assigns the root module name for the Golden design. This is the default.

-Revised

Assigns the root module name for the Revised design.

-Both

Assigns the root module name for both the Golden and Revised designs.e.g.

```
TCL_SETUP> report_environment -NON_DEFAULT
   Golden:
      Root module                  : dtmf_recvr_core (*)
      Inverted pin extension   : "_BAR" (*)
   Revised:
      Root module                  : dtmf_recvr_core (*)
      Inverted pin extension   : "_BAR" (*)
   Logfile                              : ./LOGS/log_g1g2.lec.20.20-s200 (*)
   Flatten model:
      Sequential constant        : ON (*)
      Gated clock                     : ON (*)
   Mapping method:
      Z_NAMEMAP_EXCLUDE                     : UNCONNECTED%d$ UNCONNECTED_HIER_Z%d$ (*)
   0
   CPU time        : 11.22     seconds
   Elapse time   : 4946       seconds
   Memory usage : 197.38    M bytes
TCL_SETUP> set_root_module arb -both
   0
   CPU time        : 11.77     seconds
```

```
    Elapse time   : 5331      seconds
    Memory usage : 197.38   M bytes
TCL_SETUP> report_environment -NON_DEFAULT
    Golden:
      Root module                  : arb (*)
      Inverted pin extension   : "_BAR" (*)
    Revised:
      Root module                  : arb (*)
      Inverted pin extension   : "_BAR" (*)
    Logfile                         : ./LOGS/log_g1g2.lec.20.20-s200 (*)
    Flatten model:
      Sequential constant        : ON (*)
      Gated clock                  : ON (*)
    Mapping method:
      Z_NAMEMAP_EXCLUDE                      : UNCONNECTED%d$ UNCONNECTED_HIER_Z%d$ (*)
```

add_compared_points

Adds mapped points to the compare list. You can add compare points for all mapped points, or for a list of the gate ID numbers, instance paths, or pin paths.

If you add a compare point to the Golden design, the Conformal software also adds its mapped compare point from the Revised design. Alternately, if you add a compare point to the Revised design, the software also adds its mapped compare point in the Golden design.

compare

Starts the equivalency checking comparison between the Golden and Revised designs on the added compared points. During the comparison, the following information is displayed:

Progress percentile number, which displays the completion rate

Running count, which displays the number of key points that have been compared along with the total number of non-equivalent key points

Innovus

suspend

Suspends your script and returns to the Innovus prompt.

The suspend command gives you more flexibility in managing and debugging your scripts. You can insert multiple suspend commands in a script. The suspend command can also be used in nested scripts. When the script reaches a point where you have inserted the suspend command, it stops and restores access to the Innovus prompt. You can then type any command required for debugging.

Whenever you want to resume your script, just type resume at the Innovus prompt.

resume

Resumes a suspended script from the point where it had stopped.

Use the suspend and resume commands to debug your scripts interactively. The suspend command stops a script and restores access to the Innovus prompt. You can then type any command required for debugging. Whenever you want to resume the suspended script, type resume at the Innovus prompt.

setFillerMode

Controls certain aspects of how the software adds filler cells. The mode setting is persistent and saved along with the database by the saveDesign command.

Use getFillerMode to return the current settings for the setFillerMode command.

The setFillerMode parameters affect the behavior of the following commands:

addFiller

deleteFiller

Note: Parameters specified by the addFiller or deleteFiller command supersede those specified by the setFillerMode command

-core {{list_of_cells1}

Specifies the filler cells to add with addFiller. Enclose the filler cell names in quotation marks (") or curly braces ({ }). You can specify multiple cell lists at one time, using curly braces ({ }) to separate each one.

Default: "" (empty string).

getFillerMode

Returns the information about setFillerMode parameters in the Innovus log file and in the Innovus console. Returns the following information.

Parameter name

Current value

Type (Boolean, string, and so on)

Whether the current value was set by user

If you do not specify a parameter, the software displays values for all of the setFillerMode mode parameters.

applyGlobalNets

Applies or restores the global net connectivity rules to the design and creates the necessary connections between instances and these global nets.

clearGlobalNets

Resets the logical power net connection. It also clears all the tie-high and tie-low nets logical power net connection.

e.g.

```
globalNetConnect VDD! -pin VDD -override
globalNetConnect VSS! -pin VSS -override
clearGlobalNet
```

ecoDefIn

Restores physical information from an old design and compares this information with the design in memory. This command uses the specified DEF file in the following ways for pre-mask and post-mask ECO:

1) Used in **pre-mask mode** (**without the -postMask parameter**) this command behaves as follows:

New cells and nets

Cells that exist in memory but not in the DEF file become unplaced.

Deleted cells or nets

Cells and nets that exist in the DEF file but not in memory are discarded.

Modified nets

Nets found both in the DEF file and in memory, but whose connections are different, are marked as "modified" for further processing during ecoRoute.

Matched cells/nets

Nets and cells that exist both in the DEF file and in memory, and have the same connections, are placed as described in the DEF file. This includes the soft matching capability used in defIn.

2) In **post-mask mode (with the -postMask parameter)**, the software can **only change** nets, not cells. This command behaves as follows:

New added cells and nets

Cells that exist in memory but not in the DEF file become unplaced. Later, these cells will be mapped to spare cells when you use ecoPlace -useSpareCells.

Deleted cells

Cells and instances that exist in DEF but not in memory are **added to a spare cell list in Innovus**: They are not discarded. Deleted cells are renamed with their original name and the **suffix** you specify. The default suffix is **_SPARE**. For example, cellA is renamed cellA_SPARE. The software retains deleted nets and their routing, which ecoRoute reroutes later.

./LAB2/PNR2/ecoDefIn.rpt:DELINST TDSP_CORE_INST/EXECUTE_INST/g26843 NOR2_X2 (converted to spare cell TDSP_CORE_INST/EXECUTE_INST/g26843_**SPARE**)
./LAB2/PNR2/ecoDefIn.rpt:DELINST TDSP_CORE_INST/EXECUTE_INST/g26883 NAND3_X1 (converted to spare cell TDSP_CORE_INST/EXECUTE_INST/g26883_**SPARE**)
./LAB2/PNR2/ecoDefIn.rpt:DELINST TDSP_CORE_INST/EXECUTE_INST/g26924 NOR4_X1 (converted to spare cell TDSP_CORE_INST/EXECUTE_INST/g26924_**SPARE**)

Deleted nets

Nets that exist in DEF but not in memory are kept and processed later when you use the ecoRoute command.

Modified nets

Same behavior as in the pre-mask mode.

Matched cells and nets

Same behavior as in the pre-mask mode.

Physical cells

Cells that exist in the DEF file but not in memory, which are marked with +SOURCE DIST in the DEF file, are restored.

If you specify -useGACells, this implies a **post-mask mode**. This option behaves as follows:

For cells that match the specified GA core site:

New added GA cell that exists in memory but not in the DEF file are left unplaced.

New GA cells that exist in the DEF file but not in memory are deleted.

For cells that do not match the specified GA core site, the software treats them as standard cells as described in -postMask mode

If you specify a report filename, this command reports the following information:

New cells and nets found in memory but not in the DEF file

Deleted cells and nets found in DEF but not in memory

Nets whose connection does not match

Cells and nets that match in Innovus and DEF

checkPlace

```
TCL_SETUP> add_spare_cell -def ../PNR1/dtmf_recvr_core.tapeout.def -spare *spare*
// Note: 378 cells added
0
TCL_SETUP> add_spare_cell -freedcell
// Note: 3 cells added
TCL_SETUP> report_spare_cell
    (1) (F) TDSP_CORE_INST/EXECUTE_INST/g26843 <NOR2_X2>
    (2) (F) TDSP_CORE_INST/EXECUTE_INST/g26883 <NAND3_X1>
    (3) (F) TDSP_CORE_INST/EXECUTE_INST/g26924 <NOR4_X1>
    (4) (S) ARB_INST/spare_0 <SDFFRS_X1> (554420, 215600)
    (5) (S) ARB_INST/spare_1 <SDFFRS_X1> (564300, 344400)
    (6) (S) ARB_INST/spare_2 <SDFFRS_X1> (554800, 218400)
    (7) (S) ARB_INST/spare_3 <SDFFRS_X1> (376960, 201600)
    (8) (S) ARB_INST/spare_4 <SDFFRS_X1> (382660, 243600)
    (9) (S) ARB_INST/spare_5 <SDFFRS_X1> (394440, 204400)
  ...
```

(1), (2), (3) is freedcell

refinePlace

You do not need a fully placed design to run refinePlace. The refinePlace command only makes changes if the placement is not legal, such as for instance overlap, incorrect orientation, constraint violations, and placement/routing blockage violations

The refinePlace command is called automatically at the end of each major super command such as place_opt_design, optDesign, and ccopt_design to ensure that the final placement is violation-free. If any instances are moved manually, refinePlace can be run to legalize any cell placements that may have resulted in violations due to this manual movement.

refinePlace only operates on the cells which have the 'placed' placement status.

To place unplaced instances, use the ecoPlace command.

-eco {true|false}

Activates ECO mode for the post-route design.

Default: false

Note: This option is controlled by

setPlaceMode

-place_detail_eco_max_distance maxDistance

Specifies max distance, in microns, in the horizontal or vertical direction for the refinePlace ECO mode.

For example, if the original location of the instance is at (x0 y0) and the specified eco max distance is M, the search window will be (x0-M, yo-M) (x0+M y0+M). The maximum movement should be less than 2M.

Default: 0

Minimum: 0.000000

Maximum: 9999.000000

If -place_detail_eco_max_distance is not defined by the user, and the user does refinePlace -eco, the value of -place_detail_eco_max_distance is internally set to 10 Standard Cell height units.

Note: This option will impact both refinePlace -eco and refinePlace -inst.

-place_detail_eco_priority_insts {placed|fixed|eco}

placed: Treats current instances as placeable but with higher priority.

eco: refinePlace will give higher priority to ECO instances rather than placed instances. But legalization can move both kinds of placed and ECO instances.

fixed: all placed instances will be treated as fixed. So, they will not be moved for legalization

Note: All instances inserted/sized with the eco* commands have placed status(default), so refinePlace will work on them

But addInst inserts the instances at the location (0,0) with "unplaced" status.

So, either use "addInst <...> -loc {x y } -status placed" or after addInst, update the status to "placed" and move eco instances into the eco area.

Finally, refinePlace -eco will reinsert fillers after legalization.