CECO Sharing

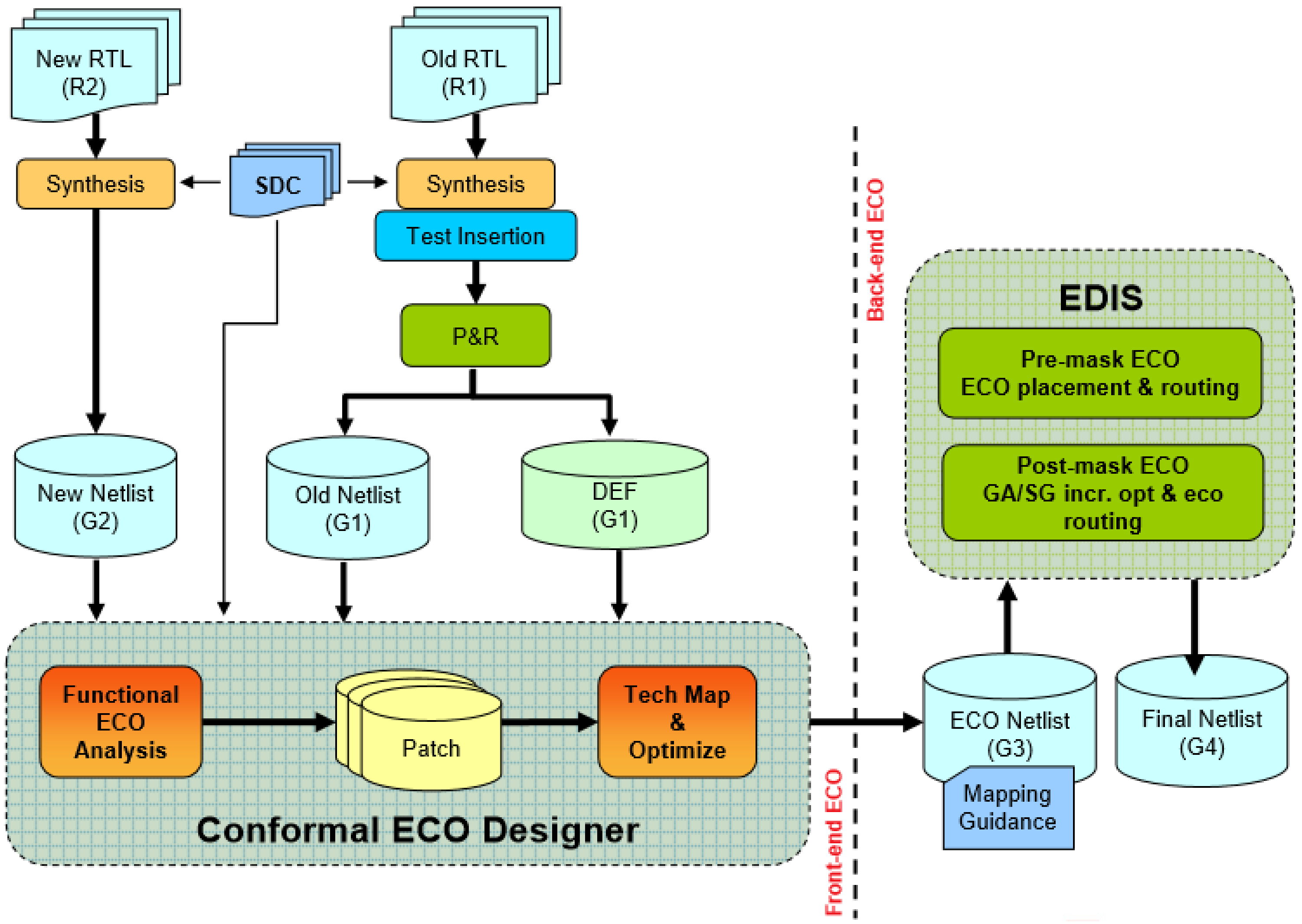
Lin Yu-Hsuan (Adam)

Singapore

Why we need CECO?

- Manual ECO is hard to correlate between RTL behavior code and Postlayout Gate-level netlist.
  - CECO use ECO netlist instead of ECO RTL to let two design get similar.
- Scope would be limited (< 1000)
  - CECO can deal with every difference.
- Even ECO script writing instead of pure manual work is still time wasting.
  - Just few commands need in CECO
- Still need to find a way to validate the ECO is correct
  - After append the ECO change to target netlist, CECO will do LEC run directly.

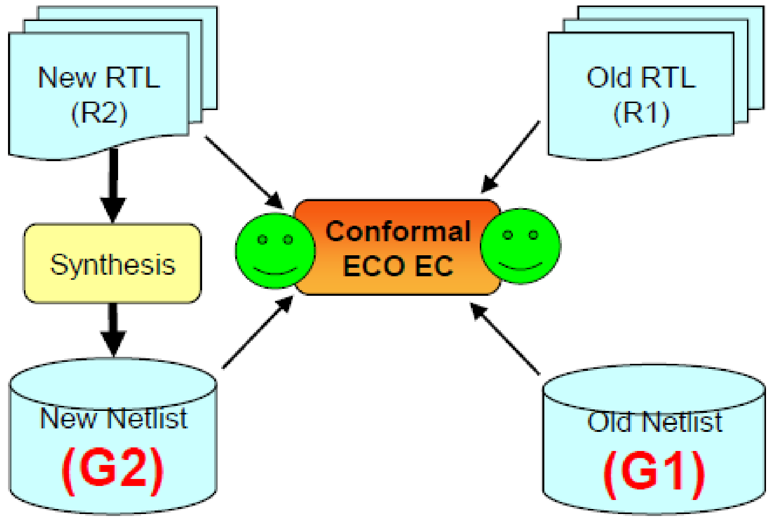
Conformal ECO Solution



Functional ECO Flow: Step 1

Create gate netlist from New RTL and identify changes

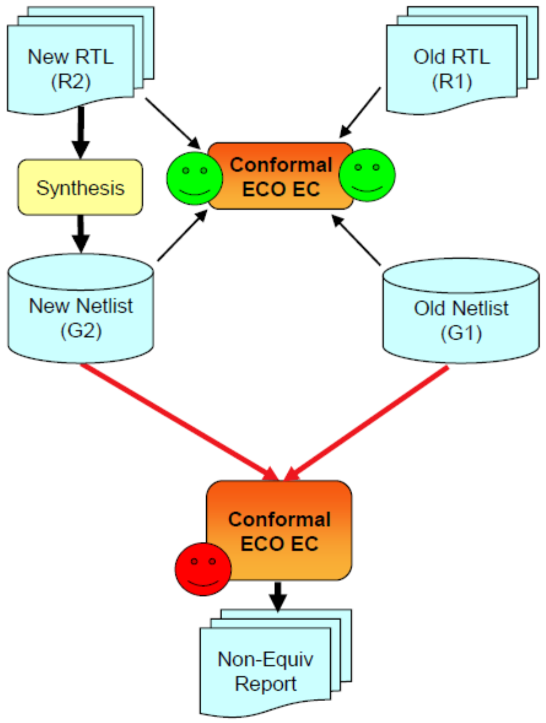
- Compare old RTL (R1) to post-layout old netlist(G1)
  - R1=G1
  - Should be EQ
- Synthesize the new netlist(G2)
  - Structurally similar netlist
- Compare new RTL (R2) to new netlist(G2)
  - R2=G2
  - Should be EQ



Functional ECO Flow: Step 2

Create a path file which contains the different G1 between G2

- Compare old RTL (R1) to new RTL (R2)
  - Identify non-equivalent modules/logic cones
  - R2-R1 = ***ΔR***
- Compare old netlist(G1) to new netlist(G2)
  - G2-G1=***ΔG***
  - Verify change with ***ΔG***
  - prepare for patch generation



Encounter Conformal ECO Flow

- Why use G2 instead R2
  - Better patch quality
    - Don’t care cone, Datapath, ICG
  - Uniquify
    - G2 don’t pay effort on uniquify
  - R2 create word level primitives that can not be optimized
    - CD, VDW_WMUX, VDW_ADD ….

- Use “check eco setup” after switch into LEC mode to double check is there any setup issue.

Template for CECO

```
set log file logfile.$LEC_VERSION –replace
read design –file G1.v -verilog –golden
read design -file G2.v -verilog –revised
add pin constraint 0 scan_en –gold     // Scan chain control
set flatten model –gated_clock // Gated clock control
set eco option -flat
set system mode lec
analyze hier –eco_aware
add compare point –all
compare
compare eco hier
analyze eco patch.v -replace –hier –ecopin ecopin.do
set system mode setup dofile ecopoin.do
apply patch -auto optimize patch -workdir rc_work -library typical_scan.lib \
-synexec "genus -legacy_ui“ -sdc core.sdc -verbose
write eco design -newfile %s.cfmECO -replace
```

$\}$ Standard LEC comparison

ADD ECO PIN

- ADD ECo Pin

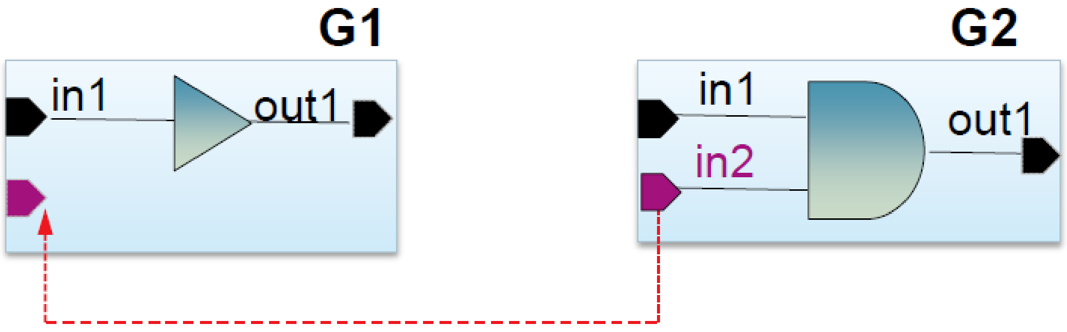
  <module_name>

  <pin_name> | <bus_name <size>> …

  [-INput | -OUTput | -IO]

  [-FORce]

  [-Golden | -Revised] (Setup mode)



```
Command: add eco pin g1_top in2 –input –golden
Command: set system mode lec
```

ADD ECO PIN

```
// Command: set system mode lec
// Mapping key points ...
// Warning: Primary input dc1 in Revised have no
correspondence(s) in Golden
```

```
// Command: analyze eco patch.v -rep
// Grouping
// Note: 3 group(s) added
// Error: Patch reaches unmapped pin 'dc1
'// Note: Use command 'add eco pin' to create new port on Golden
// Note: 0 library cell(s) is in the patch
// Note: 193 primitive(s) are in the patch
```

```
// Command: apply patch –auto -keeph
// Error: Cannot find net dc1 in module test_1 (pin:test_1_eco.dc1)
```

Report Mismatch Pin

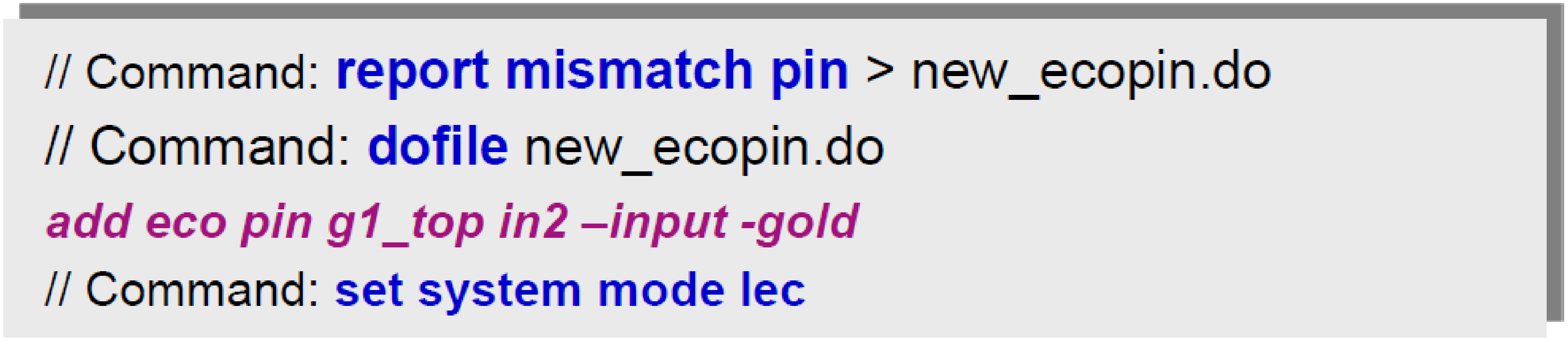
REPort Mismatch Pin

[-Gold <module_name>]

[-Revised <module_name>]

[-DELeteextra[-KEEPFreed]]

(Setup Mode)



```
// Command: report mismatch pin > new_ecopin.do
// Command: dofile new_ecopin.do
add eco pin g1_top in2 –input -gold
// Command: set system mode lec
```

Always review ECO pin matching ECO version

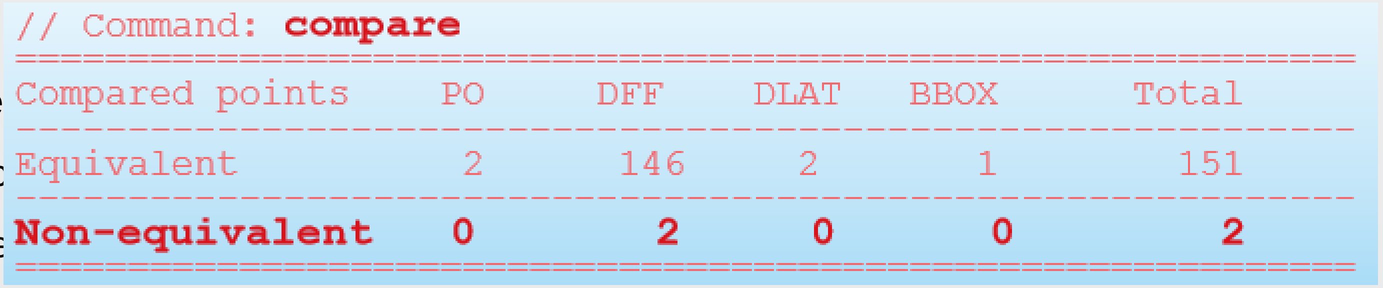
Template for CECO

```
set log file logfile.$LEC_VERSION –replace

read design –file G1.v -verilog –golden

read design -file G2.v -verilog –revised

add pin constraint 0 scan_en –gold     // Scan chain control

set flatten model –gated_clock // Gated clock control

set eco option -flat

set system mode lec

analyze hier –eco_aware

add compare point –all

compare

compare eco hier

analyze eco patch.v -replace –hier –ecopin ecopin.do

set system mode setup dofile ecopoin.do

apply patch -auto optimize patch -workdir rc_work -library typical_scan.lib \

-synexec "genus -legacy_ui“ -sdc core.sdc -verbose

write eco design -newfile %s.cfmECO -replace
```

- Matching module instances
- Generate boundary information
- Gathering information for **compare eco hier**

Template for CECO

```
set log file logfile.$LEC_VERSION –replace
read design –file G1.v -verilog –golden
read design -file G2.v -verilog –revised
add pin constraint 0 scan_en –gold      // Scan chain control
set flatten model –gated_clock // Gated clock control
set eco option -flat
set system mode lec
analyze hier –eco_aware
add compare point –all
compare
compare eco hie
analyze eco patc
set system mode
apply patch -auto optimize patch -workdir rc_work -library typical_scan.lib \
-synexec "genus -legacy_ui“ -sdc core.sdc -verbose
write eco design -newfile %s.cfmECO -replace
```

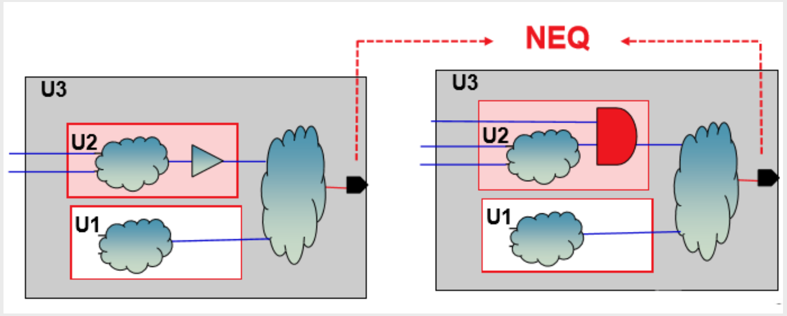


```
// Command: compare
==========================================================================
Compared points      PO       DFF      DLAT     BBOX        Total
--------------------------------------------------------------------------
Equivalent            2       146       2         1          151
--------------------------------------------------------------------------
Non-equivalent        0         2       0         0            2
==========================================================================
```

Template for CECO

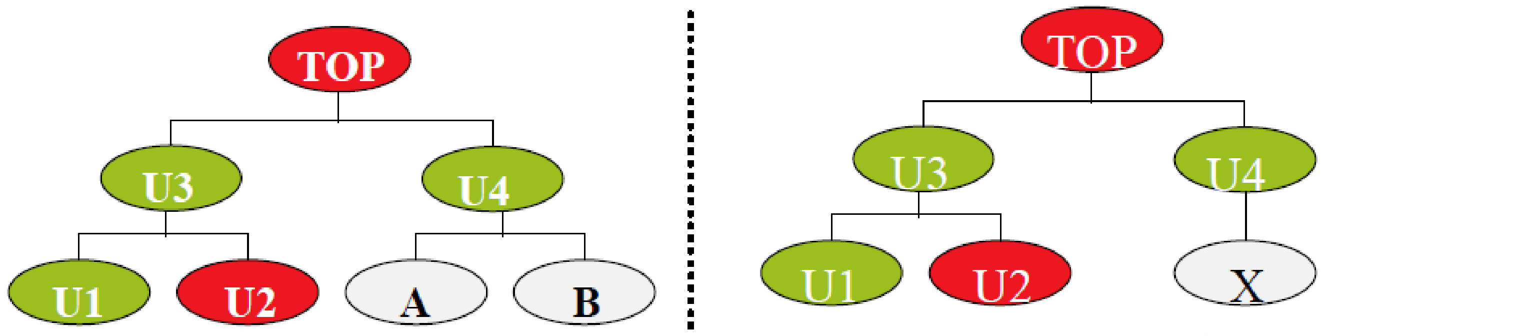
```
set log file logfile.$LEC_VERSION –replace
read design –file G1.v -verilog –golden
read design -file G2.v -verilog –revised
add pin constraint 0 scan_en –gold     // Scan chain control
set flatten model –gated_clock // Gated clock control
set eco option -flat
set system mode lec
analyze hier –eco_aware
add compare point –all
compare
compare eco hier
analyze eco patch.v -replace –hier –ecopin ecopin.do
set system mode setup dofile ecopoin.do
apply patch -auto optimize patch -workdir rc_work -library
-synexec "genus -legacy_ui“ -sdc core.sdc -verbose
write eco design -newfile %s.cfmECO -replace
```

Compare eco hier
takes the flattened compare results, along with the boundary information from ANALYZE HIER and breaks down the nonequivalent points to their associated submodules.



Compare ECO Hierarchy

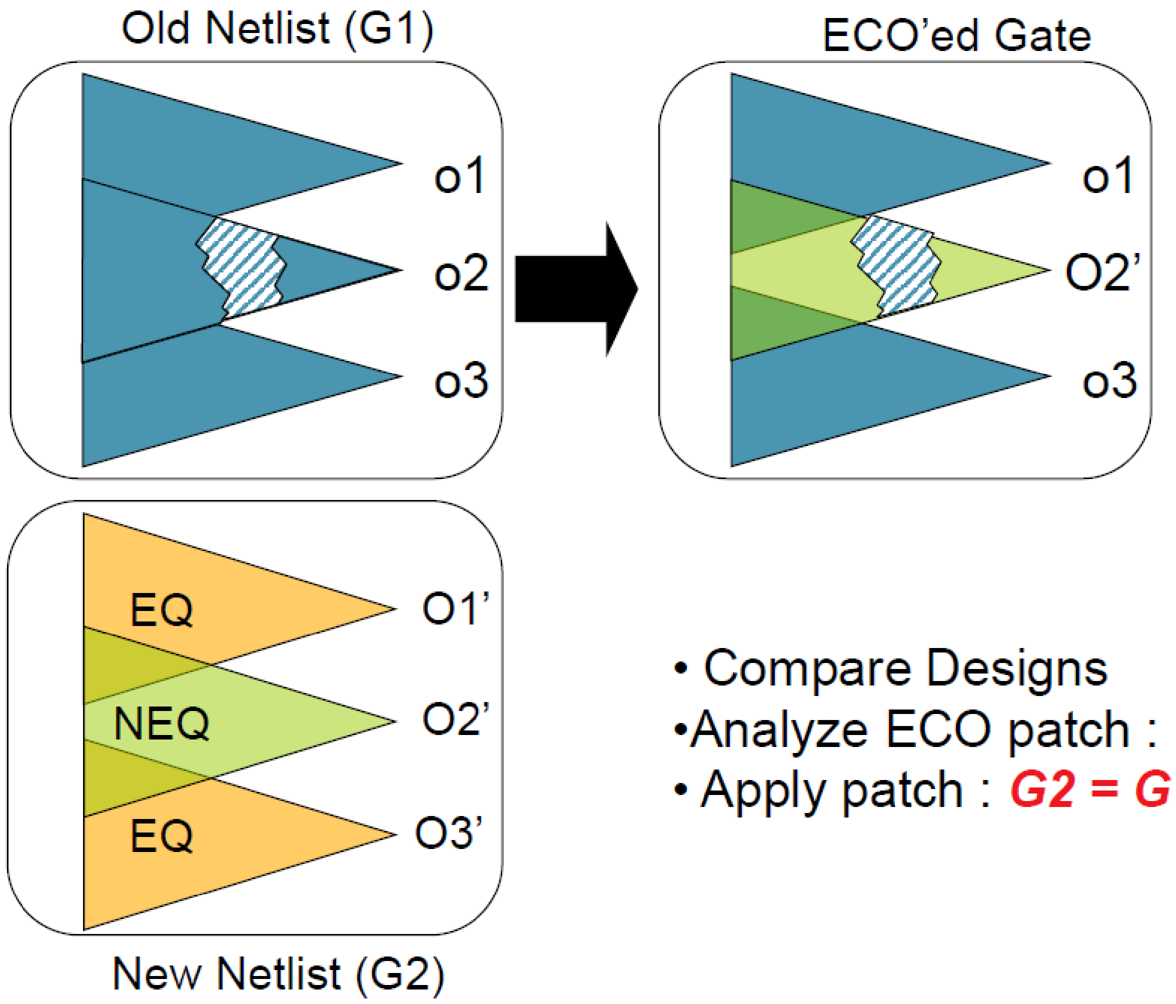
```
// Command: compare
=================================================================
Compared points      PO       DFF      Total
-----------------------------------------------------------------
Equivalent            2        146      151
-----------------------------------------------------------------
Non-equivalent        0          2        2
=================================================================
// Command: compare eco hierarchy -verbose
// Analyzing Non-equivalent logic cones
(G) MODULE U2 INSTANCE TOP/U3/U2
(R) MODULE U2  INSTANCE TOP/U3/U2
........
(G) MODULE TOP   INSTANCE TOP
(R) MODULE TOP   INSTANCE TOP
.......
Total non-equivalent modules = 2
```



Template for CECO

```
set log file logfile.$LEC_VERSION –replace
read design –file G1.v -verilog –golden
read design -file G2.v -verilog –revised
add pin constraint 0 scan_en –gold     // Scan chain control
set flatten model –gated_clock // Gated clock control
set eco option -flat
set system mode lec
analyze hier –eco_aware
add compare point –all
compare
compare eco hier
analyze eco patch.v -replace –hier –ecopin ecopin.do
set system mode setup dofile ecopoin.do
apply patch -auto ; optimize patch -workdir rc_work -library typical_scan.lib \
-synexec "genus -legacy_ui“ -sdc core.sdc -verbose
write eco design -newfile %s.cfmECO -replace
```

Patch Generation



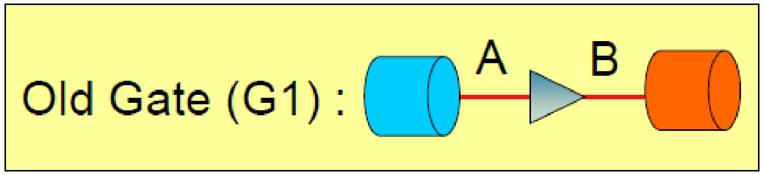
- Compare Designs
- Analyze ECO patch : ***G2-G1 = ΔG***
- Apply patch : ***G2 = G1+ΔG***

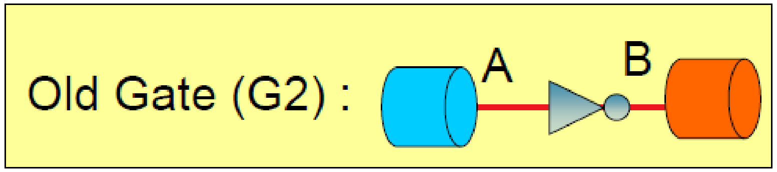
cādence®

Apply Patch

Old RTL (R1) : assign B = A;

ECO RTL (R2) : assign B = ~A;

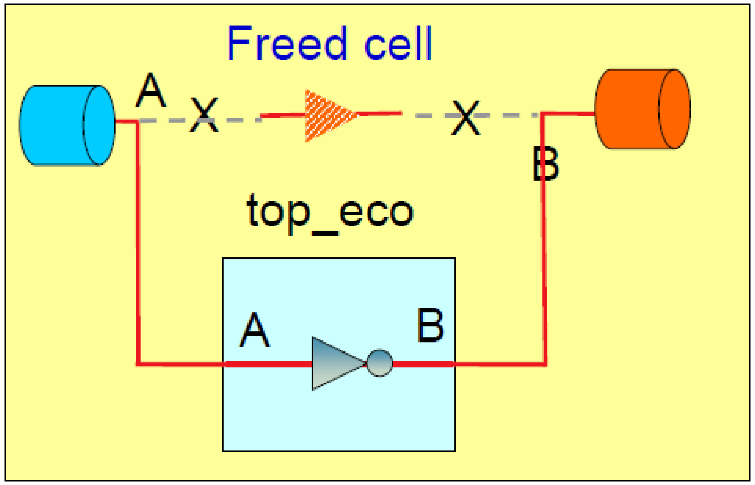




Patch:

```
Module top_eco(.A(A), .B(B));
Input A;
Output B;
INV  GATE1(.A(A), .B(B));
endmodule
```

ECO Netlist (G3)



Analyze ECO

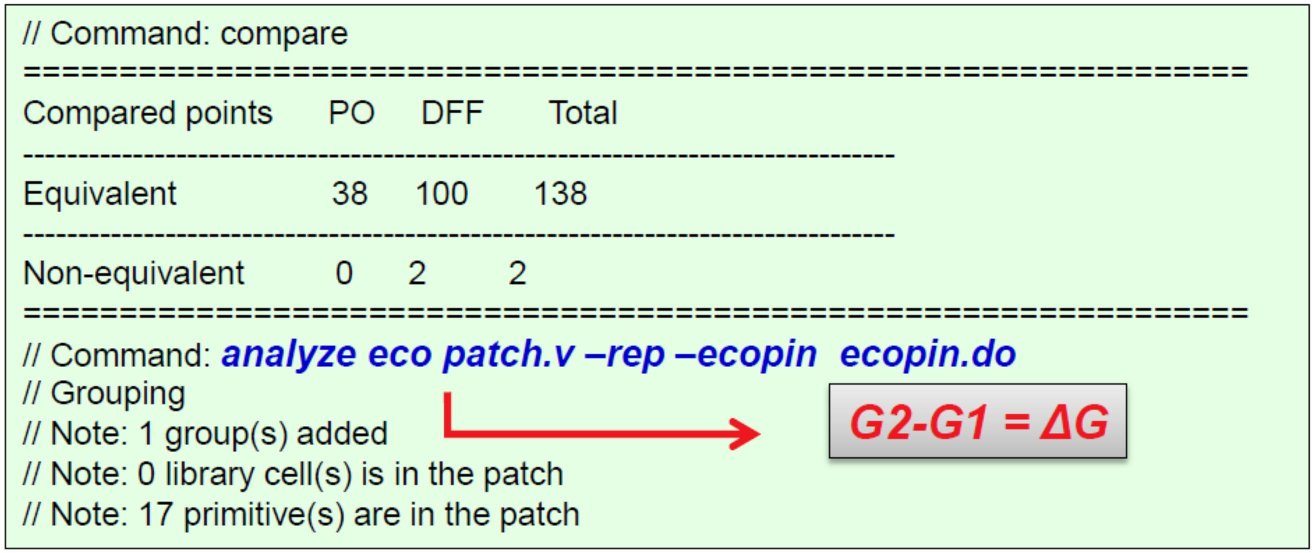
ANAlyze Eco

<path_filename>

[-REPlace]

[-ECOPIN <ecopin_dofile>]

(*LEC* mode)



```
// Command: compare
==============================================================
Compared points      PO     DFF      Total
----------------------------------------------------------------
Equivalent           38     100      138
----------------------------------------------------------------
Non-equivalent       0      2        2
==============================================================
// Command: analyze eco patch.v –rep –ecopin  ecopin.do
// Grouping
// Note: 1 group(s) added
// Note: 0 library cell(s) is in the patch
// Note: 17 primitive(s) are in the patch
```

Analyze eco –hier-ecopin

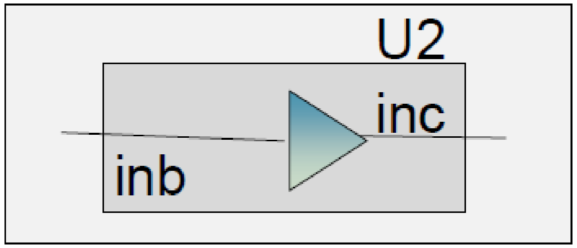
- Leverage information of '**analyze hier–eco**'
- Reached module pins in Neqcone in revised can not be found in golden will be generated.

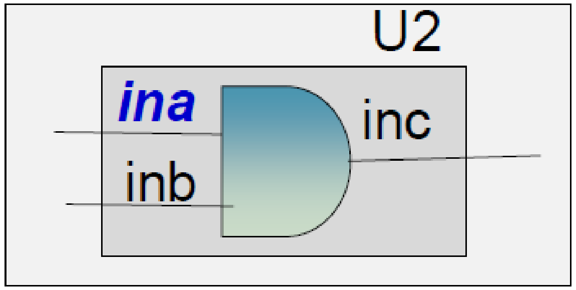
patch.v

```
module U2_eco(ina, inb, outc);
input ina, inb;
output outc;
and (outc, ina, inb);
endmodule

moudle TOP_eco(x, y,)
input x;
output y;
inv (y,x);
endmodule
```

ecopin.do

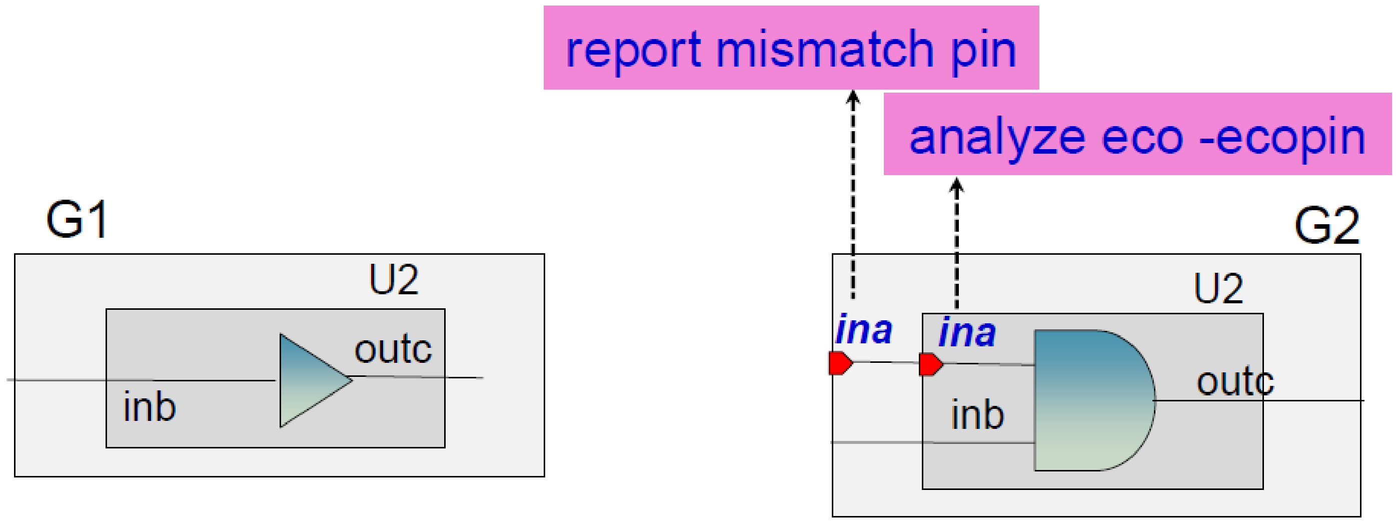
```
add eco pin U2 –in ina –force
```





ECO Pin vs Mismatch pin

- Remember Mismatch pin ??
- What is the difference of **report mismatch pin** and **analyze eco –ecopin ??**
- **Report mismatch pin** only report ROOT Module ECO Pin
- **Analyze eco –ecopin** only report block level ECO Pin



APPly Patch

APPly PAtch

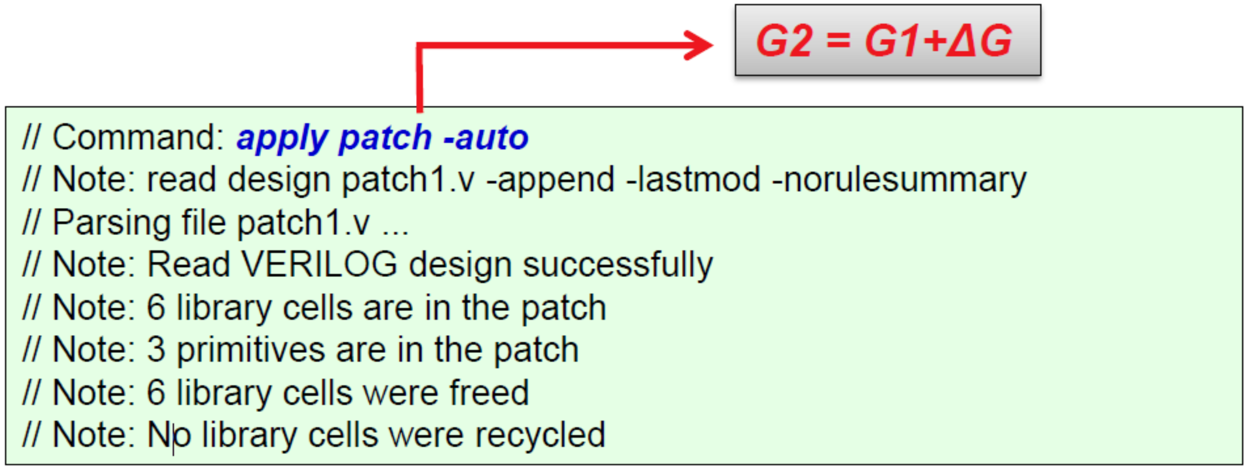
{[-auto] |<module_under_ECO_namepatch_module_name> }

[| -KEEPFREED | -TIEFREED0 | -TIEFREED1]

[-FREEDSCAN] [-stitchscan] [-shiftenable<pin> <0|1>]

[-FREEDNOSCAN][-Golden | -Revised]

(Setup Mode)



G2 = G1+ΔG

```
// Command: apply patch -auto
// Note: read design patch1.v -append -lastmod -norulesummary
// Parsing file patch1.v ...
// Note: Read VERILOG design successfully
// Note: 6 library cells are in the patch
// Note: 3 primitives are in the patch
// Note: 6 library cells were freed
// Note: Np library cells were recycled
```

Limitation of CECO

- Design hierarchy
  - With some aggressive optimization, it would eliminate design hierarchy or create some extra module pin
  - It lead ECO faild
- Garbage in, garbage out.
  - CECO would generate ECO for each non-EQ point.
  - If there are some manual setting error, it would cause huge number false non-EQ.

cādence®