

Translation of Innovus Legacy UI Commands into Common UI (CUI) Commands

Product Version Innovus 21.12
February 2022

Copyright Statement

© 2022 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence and the Cadence logo are registered trademarks of Cadence Design Systems, Inc. All others are the property of their respective holders.

Contents

Purpose	4
Overview.....	4
High-Level Flow Steps.....	4
Example.....	5
Script Name: init.tcl	5
Script Name: place.tcl	7
Script Name: cts.tcl	8
Script Name: postcts_hold.tcl.....	10
Script Name: route.tcl.....	10
Script Name: postroute.tcl.....	12
Script Name: signoff.tcl	13
Script Name: view_definition.tcl	15
Support	17
Feedback	17

Purpose

The purpose of this application note is to demonstrate translation of Innovus legacy UI commands into Common UI (CUI) commands using the real example design targeted for 16nm process node.

Overview

The typical block-level Innovus tool implementation flow consists of a set of required inputs, flow settings, and flow steps to generate a set of outputs for block-level signoff verification. The example scripts given in this application notes expect that the inputs to the flow are available in the form of gate-level Verilog netlist, scan chain in def format, floorplan and power plan file in def format, timing constraints in sdc format, and a view_definition file. The scripts in the flow step include legacy commands (commented out with # and colored blue) followed by their equivalent CUI commands. For some legacy commands, if the CUI mapping command is not available at the time of writing this application note, the legacy command is encapsulated using the “eval_legacy” CUI command.

High-Level Flow Steps

At high level, for simplicity, the following flow steps are identified:

1. Design initialization and initial checks:
Script name: init.tcl
2. Std cell placement and pre-cts timing optimization:
Script name: place.tcl
3. Clock Tree Synthesis with concurrent clock and datapath timing optimization:
Script name: cts.tcl
4. Pre-route hold fixing with propagated clocks:
Script name: postcts_hold.tcl
5. Routing with timing and SI:
Script name: route.tcl
6. SI-aware postroute timing optimization for setup and hold:
Script name: postroute.tcl
7. Running signoff static timing analysis using signoff parasitics extraction:
Script name: signoff.tcl

Example

The Following example shows how to run the CUI scripts. All these scripts can be run using Innovus 18.10-p002_1.

The example view_definition file in CUI format is also given at the end of this document for completeness.

```
innovus -stylus -init init.tcl      -log LOG/init.log
innovus -stylus -init place.tcl     -log LOG/place.log
innovus -stylus -init cts.tcl       -log LOG/cts.log
innovus -stylus -init postcts_hold.tcl -log LOG/postcts_hold.log
innovus -stylus -init route.tcl     -log LOG/route.log
innovus -stylus -init postroute.tcl  -log LOG/postroute.log
innovus -stylus -init signoff.tcl   -log LOG/signoff.log
```

Script Name: init.tcl

```
# setMultiCpuUsage -localCpu 16
set_multi_cpu_usage -local_cpu 16

read_mmmc      ../input/view_definition.tcl
read_physical  -lef { \
../libs/lef/xyz.tlef_mod \
../libs/lef/xyz_par.lef \
../libs/lef/NDR_clock.lef \
../libs/lef/std_cells.lef}
read_netlist   -top abc \
../inout/abc_mapped.v

# set init_gnd_net VSS
# set init_pwr_net VDD
set_db init_ground_nets VSS
set_db init_power_nets VDD
init_design

enable_metrics -on
push_snapshot_stack

read_def      ../input/abc_fp_and_pg.def
# defIn      ../input/abc.scan.def
read_def      ../input/abc.scan.def
# checkPlace
check_place
# set_global report_timing_format {instance cell arc fanout load
slew delay user_derate incr_delay arrival}
```

```
set_db timing_report_fields {instance cell arc fanout load
transition delay user_derate incr_delay arrival}
# setDesignMode -process 16
set_db design_process_node 16
# setMaxRouteLayer M9
set_db design_top_routing_layer M9
# setAnalysisMode -analysisType onChipVariation -cpr both
set_db timing_analysis_type ocv
set_db timing_analysis_cpr both

# globalNetConnect VDD -type pgpin -pin VDD -inst *
# globalNetConnect VSS -type pgpin -pin VSS -inst *
# globalNetConnect VDD -type tiehi
# globalNetConnect VSS -type tielo
connect_global_net VDD -type pgpin -pin VDD -inst *
connect_global_net VSS -type pgpin -pin VSS -inst *
connect_global_net VDD -type tiehi
connect_global_net VSS -type tielo

# set_verify_drc_mode -disable_rules out_of_die
set_db check_drc_disable_rules out_of_die
# verify_drc -report RPT/init_step_DRC.rpt
mkdir RPT
check_drc -out_file RPT/init_step_DRC.rpt
# verify_connectivity -net VDD -no_antenna -report \
# RPT/init_step_VDD_connectivity.rpt
check_connectivity -nets VDD -ignore_dangling_wires \
-out_file RPT/init_step_VDD_connectivity.rpt
# verify_connectivity -net VSS -no_antenna -report \
# RPT/init_step_VSS_connectivity.rpt
check_connectivity -nets VSS -ignore_dangling_wires \
-out_file RPT/init_step_VSS_connectivity.rpt
# clearDrc
delete_drc_markers
# timeDesign -preplace -prefix preplace -outDir RPT
time_design -pre_place -report_prefix preplace -report_dir RPT
# checkDesign -all
# check_design -type all
check_legacy_design -all
check_timing -verbose > RPT/check_timing.rpt
pop_snapshot_stack
create_snapshot -name init -categories "power check design flow
hold route clock setup"

# saveDesign DBS/init.enc -compress
write_db DBS/init.dat
```

```
# saveNetlist DBS/LEC/init.v.gz
write_netlist DBS/LEC/init.v.gz
# reportVtInstCount -area
report_threshold_instance_count -area

set top_cell [get_db designs .name]
report_metric -format vivid -out_file RPT/${top_cell}_metric.html
write_metric -format json -out_file RPT/${top_cell}_metric.json
exit
```

Script Name: place.tcl

```
# setMultiCpuUsage -localCpu 16
set_multi_cpu_usage -local_cpu 16
# restoreDesign DBS/init.enc.dat abc
get_common_ui_map restoreDesign
read_db DBS/init.dat

enable_metric -on
push_snapshot_stack

# setPlaceMode -place_global_place_io_pins false
set_db place_global_place_io_pins false

# setOptMode -usefulSkew true
set_db opt_useful_skew true

# setOptMode -fixFanoutLoad true
set_db opt_fix_fanout_load true

# place_opt_design -out_dir RPT -prefix place
place_opt_design -report_dir RPT -report_prefix place

# setTieHiLoMode -cell {{TIEL TIEH}}
set_db add_tieoffs_cells {TIEL TIEH}

# setDontUse {TIEL TIEH} false
get_db base_cells TIE*
set_db [get_db base_cells TIE*] .dont_use true
# addTieHiLo
add_tieoffs
# setDontUse {TIEL TIEH} true
set_db [get_db base_cells TIE*] .dont_use false

pop_snapshot_stack
enable_metric
```

```
create_snapshot -name place -categories "power check design flow  
hold route clock setup"
```

```
# saveDesign DBS/place.enc -compress  
write_db DBS/place.dat  
# saveNetlist DBS/LEC/place.v.gz  
write_netlist DBS/LEC/place.v.gz  
# reportVtInstCount -area  
report_threshold_instance_count -area  
  
set top_cell [get_db designs .name]  
report_metric -format vivid -out_file RPT/${top_cell}_metric.html  
write_metric -format json -out_file RPT/${top_cell}_metric.json  
exit
```

Script Name: cts.tcl

```
# setMultiCpuUsage -localCpu 16  
set_multi_cpu_usage -local_cpu 16  
# restoreDesign DBS/place.enc.dat abc  
read_db DBS/place.dat  
  
enable_metric -on  
push_snapshot_stack  
  
# setNanoRouteMode -drouteUseMultiCutViaEffort high \  
# -routeWithLithoDriven false  
set_db route_design_detail_use_multi_cut_via_effort high  
# following is default  
# set_db route_design_with_litho_driven false  
get_db route_design_with_litho_driven  
  
# setOptMode -usefulSkewCCOpt medium  
set_db opt_useful_skew_ccopt medium  
  
# set_ccopt_mode -cts_buffer_cells {DCCKBD8 DCCKBD10 DCCKBD12  
DCCKBD16}  
# set_ccopt_mode -cts_inverter_cells {DCCKND8 DCCKND10 DCCKND12  
DCCKND16}  
# set_ccopt_mode -cts_clock_gating_cells {CKLHQD2 CKLHQD4  
CKLHQD6 CKLHQD8 CKLHQD10 CKLHQD12 CKLHQD16 CKLNQD2 CKLNQD4  
CKLNQD6 CKLNQD8 CKLNQD10 CKLNQD12 CKLNQD16}  
set_db cts_buffer_cells "DCCKBD8 \  
DCCKBD10 DCCKBD12 DCCKBD16"  
set_db cts_inverter_cells "DCCKND8 \  
DCCKND10 DCCKND12 DCCKND16"
```

```
set_db cts_clock_gating_cells "CKLHQD2 ]
CKLHQD4 CKLHQD6 CKLHQD8 \
CKLHQD10 CKLHQD12 CKLHQD16 \
CKLNQD2 CKLNQD4 CKLNQD6 \
CKLNQD8 CKLNQD10 CKLNQD12 \
CKLNQD16"

# set_ccopt_mode -cts_use_inverters true
set_db cts_use_inverters true

create_route_type -name leaf_rt -top_preferred_layer M4 -
bottom_preferred_layer M3

# create_route_type -name trunk_rt -top_preferred_layer M9 \
# -bottom_preferred_layer M8 -non_default_rule NDR_clock \
# -shield_net VSS
create_route_type -name trunk_rt -top_preferred_layer M9 \
-bottom_preferred_layer M8 -route_rule NDR_clock -shield_net VSS

# set_ccopt_property route_type -net_type leaf leaf_rt
set_db cts_route_type_leaf leaf_rt
# set_ccopt_property route_type -net_type trunk trunk_rt
set_db cts_route_type_trunk trunk_rt

# setNanoRouteMode -routeBottomShieldLayer M5
set_db route_design_bottom_shield_layer M5
# set_ccopt_property -update_io_latency true
# following is default
# set_db cts_update_io_latency true
get_db cts_update_clock_latency

# create_ccopt_clock_tree_spec
create_clock_tree_spec
# ccopt_design -outDir RPT -prefix cts
ccopt_design -report_dir RPT -report_prefix cts

# report_ccopt_clock_trees -file RPT/rclk_full.rpt
# report_ccopt_skew_groups -file RPT/rskg_full.rpt
report_clock_trees -out_file RPT/rclk_full.rpt
report_skew_groups -out_file RPT/rskg_full.rpt

pop_snapshot_stack
enable_metric
create_snapshot -name cts -categories "power check design flow
hold route clock setup"
```

```
write_db DBS/cts.dat
write_netlist DBS/LEC/cts.v.gz
report_threshold_instance_count -area

set top_cell [get_db designs .name]
report_metric -format vivid -out_file RPT/${top_cell}_metric.html
write_metric -format json -out_file RPT/${top_cell}_metric.json
exit
```

Script Name: postcts_hold.tcl

```
# setMultiCpuUsage -localCpu 16
set_multi_cpu_usage -local_cpu 16
# restoreDesign DBS/cts.enc.dat abc
read_db DBS/cts.dat

enable_metric -on
push_snapshot_stack

# setOptMode -fixHoldAllowSetupTnsDegrade false \
# -ignorePathGroupsForHold default
set_db opt_fix_hold_allow_setup_tns_degradation false
set_db opt_fix_hold_ignore_path_groups default
# optDesign -postCTS -hold -outDir RPT -prefix postcts_hold
opt_design -post_cts -hold -report_dir RPT -report_prefix \
postcts_hold

pop_snapshot_stack
enable_metric
create_snapshot -name postcts_hold -categories design

write_db DBS/postcts_hold.dat
write_netlist DBS/LEC/postcts_hold.v.gz
report_threshold_instance_count -area

set top_cell [get_db designs .name]
report_metric -format vivid -out_file RPT/${top_cell}_metric.html
write_metric -format json -out_file RPT/${top_cell}_metric.json
exit
```

Script Name: route.tcl

```
# setMultiCpuUsage -localCpu 16
set_multi_cpu_usage -local_cpu 16
# restoreDesign DBS/postcts_hold.enc.dat abc
read_db DBS/postcts_hold.dat
```

Translation of Innovus Legacy UI Commands into CUI Commands

```
enable_metric -on
push_snapshot_stack

# setDelayCalMode -siAware true -engine aae
set_db delaycal_enable_si true

# setNanoRouteMode -drouteUseMultiCutViaEffort high \
# -routeWithLithoDriven false -routeAntennaCellName \
# ANTENNA -routeInsertAntennaDiode true
set_db route_design_detail_use_multi_cut_via_effort high
set_db route_design_antenna_cell_name ANTENNA
set_db route_design_antenna_diode_insertion true
# set_db route_design_with_litho_driven false

# setFillerMode -core {FILL64 FILL32 FILL16 FILL8 FILL4 FILL2 \
# FILL1} -corePrefix FILL
set_db add_fillers_cells "FILL64 FILL32 FILL16 FILL8 FILL4 \
FILL2 FILL1"
set_db add_fillers_prefix FILL

# addFiller
add_fillers
# setNanoRouteMode -droutePostRouteSpreadWire false
set_db route_design_detail_post_route_spread_wire false

# setDelayCalMode -equivalent_waveform_model_type vivo
# setDelayCalMode -equivalent_waveform_model_propagation true
set_db delaycal_equivalent_waveform_model propagation
set_db delaycal_ewm_type simulation
# routeDesign
route_design

# setExtractRCMode -engine postRoute -effortLevel high
set_db extract_rc_engine post_route
set_db extract_rc_effort_level high

pop_snapshot_stack
enable_metric
create_snapshot -name route -categories "power check design flow
hold route clock setup"

write_db DBS/route.dat
write_netlist DBS/LEC/route.v.gz
report_threshold_instance_count -area

set top_cell [get_db designs .name]
```

```
report_metric -format vivid -out_file RPT/${top_cell}_metric.html
write_metric -format json -out_file RPT/${top_cell}_metric.json
exit
```

Script Name: postroute.tcl

```
# setMultiCpuUsage -localCpu 16
set_multi_cpu_usage -local_cpu 16
# restoreDesign DBS/route.enc.dat abc
read_db DBS/route.dat

enable_metric -on
push_snapshot_stack

# optDesign -postRoute -outDir RPT -prefix postroute \
# -setup -hold
opt_design -post_route -report_dir RPT -report_prefix \
postroute -setup -hold

pop_snapshot_stack
enable_metric
create_snapshot -name postroute -categories design

write_db DBS/postroute.dat
write_netlist DBS/LEC/postroute.v.gz
report_threshold_instance_count -area

set top_cell [get_db designs .name]
report_metric -format vivid -out_file RPT/${top_cell}_metric.html
write_metric -format json -out_file RPT/${top_cell}_metric.json
exit
```

Script Name: signoff.tcl

```
# setMultiCpuUsage -localCpu 16
set_multi_cpu_usage -local_cpu 16
# restoreDesign DBS/postroute.enc.dat abc
read_db DBS/postroute.dat

enable_metrics -on
push_snapshot_stack

# setExtractRCMode -coupled true -effortLevel signoff
set_db extract_rc_coupled
set_db extract_rc_effort_level signoff

set vars(active_rc_corners) [list]
foreach view [concat [all_analysis_views -type setup]
[all_analysis_views -type hold]] {
    set corner [get_delay_corner [get_analysis_view $view -
delay_corner] \
    -rc_corner]
    if {[lsearch $vars(active_rc_corners) $corner] == -1 } {
        lappend vars(active_rc_corners) $corner
    }
}

puts "ACTIVE RC CORNER LIST: $vars(active_rc_corners)"
set empty_corners [list]
foreach corner $vars(active_rc_corners) {
    if {[file exists [get_rc_corner $corner -qrc_tech]]} {
        lappend empty_corners $corner
    }
}

if {[llength $empty_corners] == 0} {
# setExtractRCMode -engine postRoute -effortLevel signoff \
# -coupled true
    set_db extract_rc_engine post_route
    set_db extract_rc_effort_level signoff
    set_db extract_rc_coupled true
} else {
    puts "CAN'T RUN SIGNOFF EXTRACTION BECAUSE qx_tech_file IS
NOT DEFINED FOR these corners: $empty_corners"
# setExtractRCMode -engine postRoute -effortLevel low \
# -coupled true
    set_db extract_rc_engine post_route
    set_db extract_rc_effort_level low
    set_db extract_rc_coupled true
}
```

```
}
# extractRC
extract_rc

foreach corner $vars(active_rc_corners) {
# rcOut -rc_corner $corner -spef $corner.spef.gz
write_parasitics -rc_corner $corner -spef_file $corner.spef.gz
}

# timeDesign -prefix signoff -signoff -reportOnly -outDir RPT
time_design -report_prefix signoff -sign_off -report_only \
-report_dir RPT
# timeDesign -prefix signoff -signoff -reportOnly -hold -outDir
RPT
time_design -report_prefix signoff -sign_off -report_only \
-hold -report_dir RPT

# summaryReport -outDir RPT
report_summary -out_dir RPT

# verifyConnectivity -noAntenna
check_connectivity -ignore_dangling_wires -out_file \
RPT/signoff_step_connectivity.rpt

# verify_drc
check_drc -out_file RPT/signoff_step_DRC.rpt

# verifyMetalDensity
check_metal_density -report RPT/signoff_step_metal_density.rpt

# verifyProcessAntenna
check_process_antenna -out_file\
RPT/signoff_step_process_antenna.rpt

pop_snapshot_stack
create_snapshot -name signoff -categories "power check design flow hold route
clock setup"
# saveDesign DBS/signoff.enc -compress
write_db DBS/signoff.dat
# saveNetlist DBS/LEC/signoff.v.gz
write_netlist DBS/LEC/signoff.v.gz
set top_cell [get_db designs .name]
report_metric -format vivid -out_file RPT/${top_cell}_metric.html
write_metric -format json -out_file RPT/${top_cell}_metric.json
exit
```

Script Name: view_definition.tcl

```
create_rc_corner -name cwcwt_25 \  
  -T 25 \  
  -qrc_tech ../../libs/qrcTech/cwcwt/qrcTechFile \  
  -pre_route_res 1.00 \  
  -pre_route_cap 1.00 \  
  -pre_route_clock_res 0.00 \  
  -pre_route_clock_cap 0.00 \  
  -post_route_res "1.00 1.00 1.00" \  
  -post_route_cap "1.00 1.00 1.00" \  
  -post_route_clock_res "0.00 0.00 0.00" \  
  -post_route_clock_cap "0.00 0.00 0.00" \  
  -post_route_cross_cap "1.00 1.00 1.00"  
  
create_rc_corner -name cbc_b_0 \  
  -T 0 \  
  -qrc_tech ../../libs/qrcTech/cbc_b/qrcTechFile \  
  -pre_route_res 1.00 \  
  -pre_route_cap 1.00 \  
  -pre_route_clock_res 0.00 \  
  -pre_route_clock_cap 0.00 \  
  -post_route_res "1.00 1.00 1.00" \  
  -post_route_cap "1.00 1.00 1.00" \  
  -post_route_clock_res "0.00 0.00 0.00" \  
  -post_route_clock_cap "0.00 0.00 0.00" \  
  -post_route_cross_cap "1.00 1.00 1.00"  
  
create_rc_corner -name rcwcw_125 \  
  -T 125 \  
  -qrc_tech ../../libs/qrcTech/rcwcw/qrcTechFile \  
  -pre_route_res 1.00 \  
  -pre_route_cap 1.00 \  
  -pre_route_clock_res 0.00 \  
  -pre_route_clock_cap 0.00 \  
  -post_route_res "1.00 1.00 1.00" \  
  -post_route_cap "1.00 1.00 1.00" \  
  -post_route_clock_res "0.00 0.00 0.00" \  
  -post_route_clock_cap "0.00 0.00 0.00" \  
  -post_route_cross_cap "1.00 1.00 1.00"  
  
create_library_set -name libs-wcg \  
  -timing [list \  
    ../../libs/lib/wcg/stdcell11_wcg_0p72_ecsm.ldb \  
    ../../libs/lib/wcg/stdcell12_wcg_0p72_ecsm.ldb]  
create_library_set -name libs-bcg \  

```

```
-timing [list \  
../libs/lib/bcg/stdcell11_bcg_0p88_ecsm.ldb \  
../libs/lib/bcg/stdcell12_bcg_0p88_ecsm.ldb]  
  
create_library_set -name libs-tc \  
-timing [list \  
../libs/lib/tc/stdcell11_tc_0p80_ecsm.ldb \  
../libs/lib/tc/stdcell12_tc_0p80_ecsm.ldb]  
  
create_timing_condition -name tc_wcg \  
-library_sets [list libs-wcg]  
create_timing_condition -name tc_bcg \  
-library_sets [list libs-bcg]  
create_timing_condition -name tc_tc \  
-library_sets [list libs-tc]  
  
create_delay_corner -name delay_bcg_cbcb \  
-timing_condition tc_bcg \  
-rc_corner cbcb_0  
create_delay_corner -name delay_wcg_rcwcw \  
-timing_condition tc_wcg \  
-rc_corner rcwcw_125  
create_delay_corner -name delay_tc_cwcwt \  
-timing_condition tc_tc \  
-rc_corner cwcwt_25  
  
create_constraint_mode -name func \  
-sdc_files [list ../input/abc_mapped.sdc]  
  
create_analysis_view -name func_best \  
-constraint_mode func \  
-delay_corner delay_bcg_cbcb  
create_analysis_view -name func_worst \  
-constraint_mode func \  
-delay_corner delay_wcg_rcwcw  
  
set_analysis_view \  
-setup [list func_worst] \  
-hold [list func_best]
```

Support

Cadence Support Portal provides access to support resources, including an extensive knowledge base, access to software updates for Cadence products, and the ability to interact with Cadence Customer Support. Visit <https://support.cadence.com>.

Feedback

Email comments, questions, and suggestions to content_feedback@cadence.com.