ANSYS RedHawk-SC-Electrothermal™

Chip Centric 3DIC Thermal Intergrity Analysis

TCL Command Reference

June 04, 2024

Software Release 2024.R2

Copyright Notice and Proprietary Information

Disclaimer

Trademarks and Registered Trademarks

ANSYS INC
Semiconductor Business Unit
2645 Zanker Road
San Jose, CA 95134

Main: 408-457-2000
Fax: 408-428-9569

Table of Contents

TCL Commands

Introduction

The RedHawk-SC Electrothermal graphical user interface (GUI) provides a graphical, interactive way to setup and run system simulation. For an alternative method, the GUI also provides acommandline window for entering Tool Command Language (TCL) commands that execute individualsteps in the RedHawk-SC Electrothermal flow.

You can also completely bypass the GUI and run RedHawk-SC Electrothermal in non-graphical batchmode. In this case, you would submit a TCL file to set up and run the simulation. Use the following example to run RedHawk-SC Electrothermal in batch mode. After completing installation of RedHawk-SC Electrothermal, type a command line on your UNIX type system in the following format:

2D: redhawk_sc_et -ng <*path*>/<*tcl_file*>.tcl

3D: redhawk_sc_et -3dic -ng <*path*>/<*tcl_file*>.tcl

For example,

redhawk_sc_et -3dic -ng generic.tcl

This document describes the TCL commands and their syntax to use in the command file. The commands and parameters are not case-sensitive. You can use the TCL command file example as a template to create a custom batch file to set up and run a simulation on your own design.

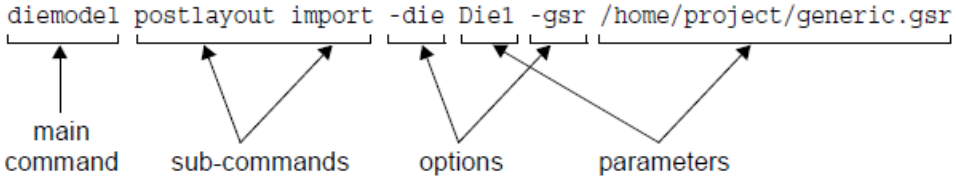
Syntax Conventions

RedHawk-SC Electrothermal uses the following TCL command syntax conventions:

Syntax	Description
<x>	x is a user-defined variable name or value
[a | b | c]	one of the a, b, c optional parameters is required
? x ?	x is an optional parameter
{a b c d}	a, b, c, and d are all required parameters
{x …}	additional x-type elements are optional

TCL Command Descriptions

TCL commands consist of a main command, up to two sub-commands, and option wordsusually followed by some parameters. Option words begin with a dash line (–). The following example illustrates these TCL command parts:



The main command and option words are case-sensitive. For clarity, some sub-command words in the syntax descriptions include mixed-case characters, but in use the sub-commands are not case-sensitive. Option parameters parsed from a file are case-sensitive.

When using the GUI, you can get syntax information on a command by typing the command words (shown in bold in the following descriptions) followed by -h or -help.

Wizard

project option

Description: Select flow type and modules.

Syntax:

project option set ?-flow <flow_type>? ?-chipcell <true|false>? ?-chipmodel <true|false>? ?-cpm <true|false>?

where

-flow <*flow type*>: flow type supported: 3DICSetup, 3DICTI, 3DICEMI, CPSTI, CPSEMI, CTMGen.

-chipcell: enable chip cell module in project tree.

-chipmodel: enable chip model module in project tree.

-cpm: enable CPM module in project tree.

Examples:

(1) project option set -flow CPSTI

Desktop

project help

Description: Used in the GUI. When you type this command in the TCL Window, the Message Window displays help information.

Syntax:

```
project [-h|-help] [new|open|info|save|close|script|set]
```

Examples:

(1) `project -h` ; Lists project commands and their syntax

(2) `project -help open` ; Lists the syntax for `project open`

(3) `help project open`

---

project new

Description: Creates a new project directory.

Syntax:

project new -name <*project_path/project_name*>

where

-name < *project_path/project_name*>: Specifies the path and name of the new project

Examples:

(1) project new -name /home/projects/case_demo

---

project open

Description: Opens an existing project and accesses files in the project directory.

Syntax:

project open -name <*project_path/project_name*>

where

-name < project_path/project_name>: Specifies the path and name of the new project

Examples:

(1) project open -name /home/projects/case_demo

---

project save

Description: Saves the current project to the original directory or specifies a new project name and directory.

Syntax:

project save ?-name < *project_path/project_name*>?

where

-name < project_path/project_name>: Specifies the path and name of the new project

Examples:

(1) project save

(2) project save -name /home/projects/case_demo

---

project close

Description: Closes the current project.

Syntax:

project close

---

project script

Description: Records and saves a TCL command sequence as a script; runs a previously saved script.

Syntax:

`project script record -file <`*path/filename*`> -start` ; or

`project script record -end` ; or

`project script run -file <`*path/filename*`>`

Examples:

(1) `project script record -start -file /home/projects/csm_case1/tcl`

(2) `project script record -end`

(3) `project script run -file /home/projects/csm_case1/tcl`

---

project info

Description: Used in the GUI. When you type this command in the TCL Window, the Message Window displays information about the project.

Syntax:

project info

---

project set simulator

Description: Set simulator path and default simulator.

Syntax:

project set simulator -simtype <simulator type> -simpath <simulator path> ?-default <hspice | nspice | eldo | spectre | lynxspice | nexxim | nexximinternal>?

project set simulator -mapdl <path>

project set simulator -aedticepak <path>

project set simulator -layout <path>

Examples:

project set simulator -simtype nspice -simpath /home/sso/bin/nspice -default hspice

project set simulator -mapdl /appls/tools/snpkg/mapdl/v201/ansys/bin/mapdl

project set simulator -simtype helic -simpath /projs00/Builds/Helic/helic_2023.R1.1

---

nxutil

Description: Export die thickness based on the Apache technology file.

Syntax:

nxutil tech info -name <varaible name> -file <tech_file> ?-output <output_file>?

where

-name < project_path/project_name>: Specifies the path and name of the new project

Examples:

nxutil tech info -name DIE4 -file /home/qli/chip.tech -output /home/qli/chip_thickness

---

set env

Description: update the FFT view in the “CPM Properties” viewer, the default frequency point value is 750.

Syntax:

set ::env(FFT_FREQ_POINT_No) <value>

Example

set:: env(FFT_FREQ_POINT_No) 1000

---

material setup

Description: support import stress-strain curve in material setup

Syntax:

material setup user -name <name> -plastic -file <file>

material setup user -duplicate -name <material name> -system

material setup user -thermal -name <material name> -ts <value> -k <value1, value2, value3> -density <value1, value2, value3> -cp <value1, value2, value3> -E < value1, value2, value3> -cte < value1, value2, value3> -pr <value1, value2, value3> -nu <value1, value2, value3> -G <value1, value2, value3> -emissivity <value1, value2, value3>

Examples:

material setup user -name C4_solder -plastic -file /home/user/StrainStressCurves.txt

material setup user -duplicate -name COPPER -system

material setup user -thermal -name COPPER -ts 25 -k { 300, 300, 300 } -density { 8.9, 8.9, 8.9 } -cp { 0.4, 0.4, 0.4 } -E { 120, 120, 120 } -cte { 17.7, 17.7, 17.7 } -pr { 0.38, 0.38, 0.38 } -nu { 0.38, 0.38, 0.38 } -G { 43.4783, 43.4783, 43.4783 } -emissivity { 0.9, 0.9, 0.9 }

Chip Model

---

cpm import

Description: Import original CPM model.

Syntax:

cpm import ?-name <new_model_name>? -file <cpm_file> -net {pwr1, gnd1, pwr2, gnd2, ...} -begin <N ns> -end <N ns>

where

-name: specify CPM model name. if no this option, set default name.

-file: specify CPM file.

-net: specify power domain net domain list.

-begin: specify CPM start length (ns).

-end: specify CPM end length (ns).

Example:

(1) cpm import -file $cpmfile -net $power_domain1

---

cpm model new

Description: Add manual, random, envelop CPM model.

Syntax:

cpm model new -cpmobj <parent CPM name> ?-type <manual | random | envelop>? ?-name <new model name>?

where

-cpmobj: specify parent import_cpm name or pseudo_cpm name.

-type: specify the submodel type (manual, random, envelop).

-name: specify the new model name. if no this option, set default name.

Example:

cpm model new -type manual -cpmobj import_cpm1

---

cpm cpm properties

Description: Change the unit of FFT curve

Syntax:

cpm cpm properties -model <CPM model name> -[Envelop_FFT | Full_FFT] -normalized_unit [on|off]

where

-normalized_unit: open or close the normalized unit

Example:

cpm cpm properties -model import_cpm1 -Envelop_FFT -normalized_unit off

---

cpm parasitic generate

Description: Rgenerate CPM parasitic part

Syntax:

cpm parasitic generate -model <model_name> -domain {power_net, ground_net} ?-accurate_low? -rcfreq {freq, rdie, cdie, ...}

where

-accurate_low: add this option to make low frequency more accuracy

Example:

cpm parasitic generate -model import_cpm3 -domain { VDD_15, VSS } -rcfreq { 1MHz, 57.9018mOhm, 799.802pF, 2MHz, 58mOhm, 799.72pF, 3.01709MHz, 159.063mOhm, 799.716pF, 9.10282MHz, 150.258mOhm, 798.676pF, 27.464MHz, 116.658mOhm, 796.333pF, 82.8614MHz, 89.413mOhm, 797.508pF, 250MHz, 97.1251mOhm, 843.512pF, 500MHz, 134.805mOhm, 965.865pF, 750MHz, 166.97mOhm, 1087.88pF, 1GHz, 188.58mOhm, 1166.29pF, 1.25GHz, 202.078mOhm, 1191.23pF, 1.5GHz, 210.178mOhm, 1172.97pF, 1.75GHz, 214.751mOhm, 1127.79pF, 2GHz, 217.01mOhm, 1069.88pF, 2.25GHz, 217.738mOhm, 1008.8pF, 2.5GHz, 217.455mOhm, 949.959pF, 2.75GHz, 216.553mOhm, 895.899pF, 3GHz, 215.207mOhm, 845.595pF, 3.25GHz, 213.392mOhm, 801.089pF, 3.5GHz, 211.577mOhm, 765.919pF, 3.75GHz, 209.613mOhm, 728.514pF, 4GHz, 207.334mOhm, 698.24pF, 4.25GHz, 204.91mOhm, 673.121pF, 4.5GHz, 202.726mOhm, 649.272pF, 4.75GHz, 200.223mOhm, 632.707pF, 5GHz, 197.674mOhm, 619.873pF }

---

cpm current import

Description: Import other CPM model file in CPM current editor dialog

Syntax:

cpm current import -model <model_name> -file <cpm_file>

Example:

cpm current import -file /home/sso/multipl_net_Export.sp -model import_cpm1

---

cpm current cut

Description: Cut CPM current to be used for creating new current profile in CPM current editor dialog

Syntax:

cpm current cut -model <model_name> -port <port name> -start < ns> -end <ns> ?-name <cut wave name>?

Example:

cpm current cut -port vdd1 -start 9.99ns -end 25.005ns -name Cut_3 -model import_cpm1

---

cpm current scale

Description: Adjust the frequency and magnitude of the cut CPM current in CPM current editor dialog

Syntax:

cpm current scale -model <model_name> -name <cut wave name> -freq_factor <number> -p2p_factor <number> -same_time_step <1 | 0>

where

-same_time_step <1 | 0>: 1, keep the same time step for scaled current with the original current

Example:

cpm current scale -name Cut_3 -freq_factor 2 -p2p_factor 0.5 -same_time_step 0 -model import_cpm1

---

cpm current waveform

Description: Adjust the frequency and magnitude of the cut CPM current in CPM current editor dialog

Syntax:

cpm current waveform -model <model_name> ?-name <wave_name>? -type [Sine|Triangular|Rectangular] -T1 <ns> -T2 <ns> -Current <N A> -period <number> -T_period <ns> -TR <ns> -TF <ns>

where

-T1 <ns> -T2 <ns>: Specify the active/idle wave length for Sine and Rectangular type

-Current <N A>: Specify the max current value

-period <number>: Specify the waveform period

-T_period <ns>: Specify the wave length for Triangular type

-TR <ns> -TF <ns>: Specify the rise/falling edge time length

Example:

cpm current waveform -name Triangular_1 -type Triangular -current 0.1A -period 1 -T_Period 50ns -TF 10ns -TR 10ns -model import_cpm1

---

cpm current create

Description: Create a new current profile in CPM current editor dialog

Syntax:

cpm current create -model <model_name> -name <current_name>

Example:

cpm current create -name Current_2 -model import_cpm1

---

cpm current update

Description: Add expression to the new current and adjust the di/dt, repeat time.

Syntax:

cpm current update -model <model_name> -name <current_name> -expression <"expression"> -di/dt <N A/ns> -repeat <N ns>

where

-expression <"expression">: Add expression to create new current profile.

Current edit Supported expression

+: CPM merge in time axis

shift(t, dc_value): Time shift in time axis. If user use shift(t)+CPM, the shift current dc value will use the first current value of CPM; If user use (CPM+shift(t)), the shift current dc value will use the last current value of CPM.

&: CPM merge in Y axis, user can do CPM1&CPM2 or only adjust DC value by CPM1&DC_value

*** two waveforms should have same “time step” & “length” for mergeing two waveform funtcion ***

| : user also can do subtract in Y axis, the usage is similar with &.

-di/dt <N A/ns>: Specify the di/dt for the connection point of current

-repeat <N ns>: Specify the repeat time for new current.

Example:

cpm current update -name Current_2 -expression PAR_0_0_VDD_15+Sine_1&0.2 -model import_cpm1

cpm current update -name Current_2 -di/dt 0.001 -repeat 25ns -model import_cpm1

---

cpm current multiple_port

Description: Create a new current profile for multiple ports

Syntax:

cpm current multiple_port -model <model_name> ?-net <net name>? -port {port1, port2,.....} -expression <"expression"> -di/dt <N A/ns> -repeat <N ns> -assign

Example:

cpm current multiple_port -port { FCHIP-V9, FCHIP-U9, FCHIP-U8, FCHIP-U7, FCHIP-V10, FCHIP-U12, FCHIP-U11, FCHIP-U10, FCHIP-B9, FCHIP-B8, FCHIP-B7, FCHIP-A9, FCHIP-B12, FCHIP-B11, FCHIP-B10, FCHIP-A10 } -expression FCHIP-V9+shift(10) -di/dt 0 -repeat 0ns -model import_cpm3

---

cpm current assign

Description: Assign a new current profile to multiple ports

Syntax:

cpm current assign -model <model_name> -port {port1, port2,....} -current {current1,current2,...}

Example:

cpm current assign -port { FCHIP-V9, FCHIP-U9 } -current current1 -model import_cpm3

---

cpm current generate

Description: Regenerate the CPM current part with new current profile.

Syntax:

cpm current generate -model <model_name>

Example:

cpm current generate -model import_cpm3

---

cpm manual set

Description: Select to use same setting for all power domians.

Syntax:

cpm manual set ?-model <model name>? -domain {power_net, ground_net} -same_envelop <true | false>

Example:

cpm manual set -model manual_cpm1 -same_envelop true

---

cpm manual addequa

Syntax:

cpm manual addequa ?-model <model name>? -domain {power_net, ground_net} -type <TR|TF|Active|idle> -name <cpm name> -period <ns> -current <A> -delay <ns> -rising <ns> -falling <ns>

where

-domain: Specify the power net and ground net for multiple power domian CPM.

-name <cpm name>: Specify the cpm name that used for active/idle period.

Example:

cpm manual addequa -model manual_cpm1 -domain {VDD VSS} -type tr -delay 0 -current 0.444367mA -rising 10ns

---

cpm manual delequa

Syntax:

cpm manual delequa ?-model <model name>? -domain {power_net, ground_net} -id <equation index>

where,

-id: Specify the equation index.

Example:

cpm manual delequa -model manual_cpm1 -domain {VDD VSS} -id 1

---

cpm manual addexcpm

cpm manual delexcpm

Description: Add/Delete multiple CPM for manual CPM creation.

Syntax:

cpm manual [addexcpm | delexcpm] ?-model <model name>? -name <extra_cpm_name> -begin <N ns> -end <N ns>

Example:

cpm manual addexcpm -model manual_cpm1 -name new_extra -file $extra_cpm

---

cpm manual renexcpm

Description: Rename the extra CPM in manual CPM.

Syntax:

cpm manual renexcpm ?-model <model name>? -name <extra_cpm_name> -newname <new extra name>

Example:

cpm manual renexcpm -model manual_cpm1 -name new_extra -newname cpm1

---

cpm random edit

Description: Set parameters for genearting random CPM.

Syntax:

cpm random edit ?-model <model name>? -domain {power_net, ground_net} -time <> -mincur <> -maxcur <> -minfreq <> -maxfreq <>

Examples:

cpm random edit -model random_cpm1 -domain $power_domain1 -time 10us -mincur 0.44367mA -maxcur 90.2703mA -minfreq 1MHz -maxfreq 0.02GHz

---

cpm random generate_envelop

Description: Generate random CPM envelop.

Syntax:

cpm random generate_envelop -model <model_name> -domain {power_net, ground_net}

Example:

cpm random generate_envelop -model random_cpm1 -domain {VDD, VSS}

---

cpm envelop edit

Description: Set parameters for genearting envelop CPM.

Syntax:

cpm envelop edit ?-model <model name>? -domain {power_net, ground_net} -type <other | power> -other <.txt/.tr0/.fsdb file> -curve <curve in .tr0/.fsdb file> -average <average power file> -profile <profile power file> -pwrnet <profile power net>

where,

-other <.txt/.tr0/.fsdb file>: Specify the current file(time/current), used for type “other”.

-average <average power file>: Specify the average.rpt file generated by PowerArtist, used for type “-power”.

-profile <profile power file>: Specify the profile_power.txt file generated by PowerArtist, used for type “-power”.

-pwrnet <profile power net>: Select the current curve in profile power.

Examples:

cpm envelop edit -model envelop_cpm1 -domain $power_domain1 -type power -average $average_file -profile $profile_power -pwrnet VDD_1.1

---

cpm envelop generate_envelop

Description: Generate envelop CPM.

Syntax:

cpm envelop generate_envelop -model <model_name> -domain {power_net, ground_net}

Example:

cpm envelop generate_envelop -model envelop_cpm1 -domain {VDD, VSS}

---

cpm model generate

Description: Do pin group and generate CPM.

Syntax:

cpm model generate -model <model_name> ?-simplify <true|false>? ?-group <group_name_list>? ?-<group_name> { <node_list>}?

Example:

cpm model generate -model import_cpm3 -group { vdd1, vss1 } -vdd1 { FCHIP-A10, FCHIP-A9, FCHIP-B10, FCHIP-B11, FCHIP-B12, FCHIP-B7 } -vss1 { FCHIP-B17, FCHIP-B2, FCHIP-C10, FCHIP-C11, FCHIP-C12, FCHIP-C3 }

---

cpm pseudo new

Description: Create pseudo CPM.

Syntax:

cpm pseudo new ?-name <model_name>? ?-curtype <file | userset>? ?-curfile <current file>? ?-curnet <power net>? ?-curfreq {freq, mincur, maxcur, ...}? ?-modeltype <classic | laplace |foster>??-paratype <file | userset>? ?-rcfile <filename>? ?-pwrnet <power net>? ?-gndnet <ground net>? ?-rcfreq {freq, rdie, cdie, ...}?

where,

-curfile <current file>: ipwr.domain file generated by Redhawk Dynamic flow, used for curtype “file”.

-curnet <power net>: Select current curve in current file.

-rcfile <filename>: If the paratype is “file”, this option is used for importing .Cdie or .inc file; If the paratype is “userset”, this option is used for importing frequency table file.

-pwrnet/gndnet: If the paratype is “file”, this option is used for selecting the net name in .Cdie or .inc file; If the paratype is “userset”, this option is used for defining the power/ground net name.

Example:

cpm pseudo new -model pre_cpm -curtype userset -curfreq {10MHz, 100mA, 1A, 1GHz, 100mA, 1A} -modeltype Laplace -paratype userset -pwrnet VDD -gndnet VSS

---

cpm model copy

Description: copy/paste cpm model

Syntax:

cpm model copy -model <model_name> ?-to <new_cpm_name>?

Example:

cpm model copy -model import_cpm1

---

csm model create

csm model delete

csm model rename

Description: Creates, renames or deletes a CSM model.

Syntax for create/delete:

```
csm model [create|delete] -die <die_name> -model <model_name>
```

Syntax for rename:

```
csm model rename -die <die_name> -model <model_name> -name
<new_model_name>
```

Example:

(1) `csm model create -die die1 -model Model1`

(2) `csm model rename -die die1 -model Model1 -name aaaa`

(3) `csm model delete -die die1 -model Model1`

---

csm model setinst

csm model addinst

csm model delinst

Description: Set, add or delete I/O cell instance to or from the CSM model.

Syntax:

```
    csm model setinst -die <die_name> -model <model_name> -inst {inst list} -
file <instance file> [-force]
   csm model [addinst|delinst] -die <die_name> -model <model_name> -inst
{inst_list} -file <instance file> [-force]
```

Example:

```
    csm model addinst -die Die1 -model Model1 -inst {PAD1,PAD3,PAD4,PAD5}
```

---

csm model setploc

csm model addploc

csm model delploc

Description: Add, set or delete the ploc pins for CSM.

Syntax:

```
    csm model [addploc|setploc|delploc] -die <die_name> -model <model_name> -
ploc {ploc list} -file <ploc file> ?-force?
```

Examples:

(1) `csm model delploc -die Die1 -model Model1 -ploc {vdd1, vss1}`

(2) `csm model addploc -die Die1 -model Model1 -ploc {vdd1, vss1}`

---

csm model addgrpploc

csm model delgrpploc

Description: Add or delete the ploc group for CSM.

Syntax:

```
    csm model [addgrpploc|delgrpploc] -die <die_name> -model <model_name> -
group <groupname> -ploc {ploc list} -file <ploc file>
```

Examples:

(1) `csm model addgrpploc -die Die1 -group group2 -ploc {VDD1, VPP1, VPP2} -model Model1`

(2) `csm model delgrpploc -die Die1 -group group1 -ploc VSS1 -model Model1`

---

csm model setmacroinst

csm model addmacroinst

csm model delmacroinst

Description: Set or ddd or delete the core instance for CSM.

Syntax:

```
csm model [setmacroinst| addmacroinst| delmacroinst] -die <die_name> -model
<model_name> -macroinst {inst list} -file <instance file> [-force]
```

Examples:

```
    csm model addmacroinst -die Die1 -model model1 -macroinst {coreinst1,
coreinst2, coreinst3}
```

---

csm model diesetup

csm model cellsetup

Description: Configures selected I/O cell instances; specifies macro model or transistor level model for CSM generation.

Syntax for diesetup:

```
    csm model diesetup -die <die_name> -model <model_name> [-allciom | -
allxtor | -allibis | -allac | -allibismodeling] [-ciom <instance list>] [-
xtor [instance list>] [-ibismodeling [instance list] [-ibis <instance list>]
[-prototype <instance list>] [-ac <instance list]] [-sigModel on|off] [-
pgGridModel on|off] [-pgGridType <classic | laplace | foster> [-freqModel
true|false] [-incpkg on/off]  [-decouple on/off]
    csm model diesetup -die <die_name> -model <model_name> ?[-allciom|-
allxtor|-allibis|-allibismodeling]? ?-ciom {inst list}? ?-xtor {inst
list}  ?-ibis {instance list}? ?-ibismodeling {instance list}? ?-prototype
{instance list}? ?-pgGridModel <true|false>? ?-sigModel <true|false>? [-
incpkg on/off]  [-decouple on/off]
```

where

- `-allciom`: Set all I/O instances to use macromodel.
- `-allxtor`: Set all I/O instances to use transistor-level model.
- `-ciom`: Set selected I/O instances to use macromodel.
- `-xtor`: Set selected I/O instances to use transistor-level model.
- `-pgGridModel`: Specify whether to include the P/G grid model.
- -pgGridType: Specify the pg grid model type
- `-sigModel`: Specify whether to include the signal model.
- `-freqModel`: If you use allac or ac option,you must set -freqModel as true at the same time
- `-incpkg`: Include package model when generate CSM.
- `-decouple`: include the coupling between signal and pg when generate CSM.

Example:

```
    csm model diesetup -xtor { PAD1, PAD3, PAD4, PAD5 } -model Model1
```

```
csm model diesetup -freqModel on -model Model1
```

---

csm model pgoption

Description: Specifies P/G grid settings for CSM model generation.

Syntax for linear sampling:

```
csm model pgoption -die <die_name> -model <model_name>
?-reuse [true|false]? ?-sampletype <lin>? ?-fmax <maximum_frequency>?
?-fstep <frequency_step>? ?-numpoles <initial_poles>?
?-ptol <pole_tolerance>?
```

Example:

```
csm model pgoption -model Model1 -reuse false -sampletype lin
-fmax 5Ghz -fstep 250MHz -numpoles 5 -ptol 1e-06
```

Syntax for logarithmic sampling:

```
csm model pgoption -die <die_name> -model <model_name>
?-reuse [true|false]? ?-sampletype log? ?-fmax <maximum_frequency>?
?-fmin <minimum_frequency>? ?-numsample <number_of_samples>?
?-numpoles <initial_poles>? ?-ptol <pole_tolerance>?
```

Example:

```
csm model pgoption -model Model1 -reuse true -sampletype log
-fmax 5Ghz –fmin 10MHz –numsample 12 -numpoles 5 -ptol 1e-06
```

---

csm model setdmp

Description: Set the DMP options in PDN tab

Syntax:

```
csm model setdmp -die <die_name> -model <model_name> -dmp on/off -
dmp_import <file> -dmp_create -dmp_job <number> -dmp_grid
<SSH|LSF|RTDA|SGE> -dmp_host <host> -dmp_queue <xxx> -dmp_large <xxx> -
dmp_small <xxx> -dmp_palenv <xxx> -dmp_append <file>
```

where

```
-dmp [on|off]: Enable or disable DMP extraction.
-dmp_import: Specify DMP external config file. this option can not exist
with -dmp_create: Specify DMP config file created by configuration
-dmp_job: Specify DMP job number
-dmp_grid: Specify DMP grid type, SSH | LSF | RTDA | SGE
-dmp_host: Specify candidate hosts for SSH or LSF type. notes multiple
           hots is splitted by char.
-dmp_queue: Specify DMP queue arguments for LSF/RTDA/SGE
-dmp_large: Specify DMP arguments for large jobs of LSF/RTDA/SGE
-dmp_small: Specify DMP arguments for small jobs of LSF/RTDA/SGE
-dmp_palenv: Specify DMP parallel environment for SGE
-dmp_append: Append one external file into DMP config config
```

Example:

```
csm model setdmp -die Die1 -model Model1 -dmp on
csm model setdmp -die Die1 -model Model1 -dmp on -dmp_job 4 -dmp_grid
lsf -dmp_small 2
```

---

csm model setmodel

Syntax for setupmodel:

```
csm model setmodel -die <die_name> -inst <inst_name> -activemodel
<model_name> -model <model_name>
```

Example:

```
csm model setmodel -die Die1 -inst mp2 -driver -model Model1
csm model setmodel -die Die1 -inst minst1 -activemodel Model2 -model
Model1
```

---

csm model generate

Description: Generates the CSM model.

Syntax:

```
csm model generate -die <die_name> -model <model_name>
```

Example:

```
csm model generate -die Die1 -model Model1
```

---

csm model export

Description: After CSM model generation, exports the model files.

Syntax:

```
csm model export -die <die_name> -model <model_name> -folder <path>
```

Example:

```
csm model export -die Die1 -model Model1 -folder /home/rhcsm/Export_csm
```

---

csm model import

Description: Import the CSM model files.

Syntax:

```
csm model import -die <die_name> -file <fileName> -model <modelName>
```

Example:

```
csm model import -die Die1 -model Model1 -folder /home/rhcsm/Export_csm
```

---

ctm|mhs import

Description: Import power source file.

Syntax:

ctm import –model <model_name> -path <file_path> or

mhs import –model <model_name> -path <file_path>

Exampel:

ctm import -model adsThermal -path ${RUN_DIR}/design_data/DIE_ctm/adsThermal.tar.gz

ctm import -model die2 -path ${RUN_DIR}/design_data/pd_map.txt

mhs import -model die2 -path ${RUN_DIR}/design_data/pd_map.txt

---

ctm|mhs create

ctm|mhs copy

ctm|mhs remove

Description: Create prototype power source model.

Syntax:

ctm|mhs create –model <model_name> -area <Length x Wide, mm>

ctm|mhs copy -model <model_name> -newname <new_name>

ctm|mhs remove –model <model_name>

Example:

ctm create -model test -area { 5, 5, mm}

ctm copy –model die1_ctm –newname die2_ctm

ctm remove –model test

---

ctm|mhs block add

ctm|mhs block modify

ctm|mhs block copy

ctm|mhs block remove

ctm|mhs change

Description: Define power block, modify power block, define constant power.

Syntax:

```
    ctm|mhs block add –model <model_name> -pwrblock <value> -area<LLx, LLy,
URx, URy, mm>
    ctm|mhs block modify –model <model_name> -pwrblock <Blockx> -area<LLx,
LLy, URx, URy, mm> -power <value>
    ctm|mhs block copy -model <model_name> -block <name1, name2>
```

```
ctm|mhs block remove -model <model_name> -block <name1, name2>
ctm|mhs change –model <model_name> -basepower <value>
```

Example:

ctm block add -model test -pwrblock 0.1 -area {-1.1795, 0.245143, -0.352392, 1.13379, mm}

ctm block modify -model test -pwrblock Block3 area {-1.1795, 0.245143, -0.352392, 1.13379, mm} -power 0.5

ctm change -model test -basepower 0.2 or

mhs change -model test -basepower 0.2

---

ctm metal scale

ctm metal remove

utctm export

Description: edit CTMv1 metal density

Syntax:

```
ctm metal scale -add -model <name> -factor< value> -area <llx, lly, urx, ury> ?-layer <name>? -shrink <true|false>
ctm metal scale -set -model <name> -value < value>[0~100], -area<llx, lly, urx, ury>  ?-layer <name> ? -shrink  <true|false>
ctm metal remove –model <name> -index <index>
ctm export -model <modelname> -folder <dir> -name exportedname
```

Example:

ctm metal scale -add -model ctm6 -factor 0.6 -area {0, 0, 122.651, 122.657} -shrink true

ctm export -model ctm6 -folder /nfs/sjocpsqa2.data/ssoqa//qa -name ctm_all

ctm metal scale -add -model ctm9 -factor 0.3 -area {83, 70, 91, 76 } -shrink true

ctm export -model ctm9 -folder /nfs/sjocpsqa2.data/ssoqa/qa -name ctm_sm

ctm metal scale -add -model ctm7 -factor 0.8 -area {0, 0, 122.651, 122.657} -layer {VIA5, M11, M14} -shink true

ctm export -model ctm7 -folder /nfs/sjocpsqa2.data/ssoqa/qa/ -name ctm_layers

ctm metal remove -model ctm7 -index 1

---

ctm utility

Description: it uses to add substrate to the CTM when there is no substrate in the CTM model, and user also can change the substrate thickness

Syntax:

ctm utility addsubstrate -thickness <value> -input <inputfile> -output <outputfile> -tech <Apache_tech_file>

ctm utility changesubstrate -input inputfile -output outputfile -thickness <value>

Please note, the unit is um.

Example:

ctm utility **addsubstrate** -thickness 50 -input /nfs/sjocpsqa2.data/ssoqa/test.tar.gz -output /nfs/sjocpsqa2.data/ssoqa/ -tech / /nfs/sjocpsqa1.data/ssoqa/testcases/MiM_typical.tech

ctm utility **changesubstrate** -input /home/qli/ctm.tar.gz -thickness 200 -output /home/qli//test

---

ctm view

Description: it uses to dump CTM image.

Syntax:

ctm view ctrl -model name -fitall

ctm view ctrl -model name -showcursorlocation true/false

ctm view ctrl -model name -range {min, max}

ctm view ctrl -model name -bgcolor reset

ctm view ctrl -model name -bgcolor { r, g, b }/{ r, g, b, r, g, b }(range 0-255) -vertical true/false(optional)

ctm view ctrl -model name -font -color { r, g, b }

ctm view ctrl -model name -font -size size

ctm view ctrl -model name -precision precision

ctm view export -model name -file filename

Example:

ctm view ctrl -model ctm_1w -fitall

ctm view ctrl -model ctm_1w -showcursorlocation true

ctm view ctrl -model ctm_1w -range {0, 0.8}

ctm view ctrl -model ctm_1w -bgcolor reset

ctm view ctrl -model name -bgcolor { 0,255,255}

ctm view ctrl -model name -font -color { 255,255,255 }

ctm view ctrl -model name -precision 6

ctm view export -model name -file /my_folder/ctm_1w.png

Layout

---

layout import

Description: import layout file.

Syntax:

layout import –name <model_name> -file <path>

Example:

layout import -name fccsp_ctm0 -file /data/ssoqa/test.xfl

---

layout setup nets

Description: layout net

Syntax:

layout setup net -model <model_name> -name [<netname> | all] ?-file <filename>? ? -type [power | ground| signal]? ?-include [true(1) | false(0)]? -model <model_name>

where,

-model: specifies the layout model name.

-name <netname> | all: specifies the nets. <netname> is a list of net names (case sensitive) separated by space. It can use * to indicate any characters (e.g. VDD* stands for all nets whose name starts with VDD). To select all nets, use "-name all".

-file <filename>: Specifies a file that contains listed net names which are separated by space or newline.

-type [power | ground | signal]: Sets the net type. This option is not required.

-include [true(1) | false(0)]: Specifies whether to include the net(s) in package extraction. This option is not required.

-model <model_name>: Indicate which cpa model.

Examples:

1. Set VDD to be power net, and VSS to be ground net. Include those two nets in CPA extraction.

layout setup net -model fccsp_demo -name VDD –type power –include 1 -model fccsp

layout setup net -model fccsp_demo -name VSS –type ground –include 1 -model fccsp

2. Clear the net selection, and include the nets listed in a file.

layout setup net -model <model_name> –name all –include 0 -model fccsp

layout setup net -model <model_name> –file PG_nets.txt –include 1 -model fccsp

where PG_nets.info contains a list of net names, e.g.

VDD

VDD_15

VSS

TCL command to combine nets. The syntax is

layout setup net -model <model_name> ?-combine? ?-name <netname_list|all>? -model

where,

-**combine:** Specifies to combine nets

-name <netname_list>: Specifies the net name list to be combined; all nets are allowed.

TCL command to remove degassing holes as void composite. The syntax is:

layout setup net del -model <model_name> ?-name <net..> -layer <layer...>? -voidnp <num_points>?

where,

-name: specifies net name list. If net name does not exist, work on all nets;

-layer: specifies layer name list. If layer is not specified, work on all layers;

-voidnp: specifies number of points (typically 4) of void composite to be deleted

---

layout setup parts

Syntax:

layout setup part -model <model_name> –name <partname> ?-type [die | resistor | inductor |

capacitor | bga |other]? ?-r <R_value>? ?-l <L_value>? ?-c <C_value>? ?-neverflip <true|false>??-netlist <filename>? ?-subckt <subckt_name>? ?-pinmap {pin1, node1, … }??-pingroup [H_division V_division | perpin]? ?-net <netname>??-thickness <value>?

where,

-model: specifies the model name

-name <partname>: specifies the parts. <partname> can be a list of part names (case sensitive) which are separated by space.

-type [die | resistor | inductor | capacitor | bga |other]: sets the type for the specified parts. This option is not required.

-r <R_value> -l <L_value> -c <C_value>: if the type of a part is resistor, inductor, capacitor or other, this option assigns a generic circuit model to the part with the RLC value. Only a numeric number need to be given as the R/L/C value, and the unit will be determined automatically. Scale symbols are allowed as R/L/C value, e.g. 1m=0.001, 1n=1e-9. Note: if the part has 2 pins, the connection between pins and circuit nodes will be made automatically.

-**neverflip <true|false>:** specifies NeverFlip is on (true) or off (false).

-netlist <filename>: specifies a circuit model to the part. <filename> can be a .sp file that contains one or more subckt. It is not allowed to mix the use of -netlist option and -r/-l/-c option.

-subckt <subckt_name>: specifies the subckt name when there are multiply subckts defined in the .sp file. It not specified, the first subckt will be used. This option has to be used together with -netlist.

-pinmap {pin1, node1, …}: connects the pins of the part to the nodes of the specified circuit model. Every two entries within the bracket is a pair. In this example, pin1 connects to node1. This option has to be used together with -netlist. Note: if the -pinmap option is not used, but the part has 2 pins and the circuit model has 2 nodes, an attempt of connection will be made.

-pingroup [H_division V_division | perpin]: specifies the pin grouping of a part. There are two usages of this tcl: 1) group by grid. “-pingroup m n” divides the part into an m by n grid, the pins inside one division become one group; 2) group each pin. “-pingroup perpin” creates a group for each pin. Use “-pingroup 1 1” to lump all pins together.

-net <netname>: specifies the net that the pin grouping is applying to. If not specified, the pin grouping is applied to all nets by default. This option has to be used together with -pingroup.

-thickness value: specifies part thickness.

Examples:

1. Specify the type of part “U2” as bga.

layout setup part -model fccsp -name U2 -type bga

2. Set the type of part “DECAP” as capacitor, and assign the RLC value.

layout setup part -model fccsp -name DECAP -type capacitor -r 1m -c 10p

3. Assign a circuit model to part “DECAP”, specify the subckt name and connect and pins to circuit nodes.

layout setup part -model fccsp -name DECAP -netlist decap_model.sp -subckt decap_1 -pinmap {pin1, vdd, pin2,vss}

4.Group the VDD pins of the part “FCHIP” into 3 by 2 grid.

layout setup part -model fccsp -name FCHIP -pingroup 3 2 -net VDD

5.Lump all VSS pins of part “FCHIP”.

layout setup part -model fccsp -name FCHIP -pingroup 1 1 -net VSS

6. Group each pin of part “FCHIP” for all nets.

layout setup part -model fccsp -name FCHIP -pingroup perpin

TCL to Create Molding Part. The syntax is

layout setup part -model <model_name> ?-add? -name <partname> ?-type molding? ?-size <LLX LLy URx URy>?

where,

-add: specifies to add a new part;

-name partname: specifies the new added molding part name;

-type molding: specifies molding part type. We only support to create a new molding part by TCL.

-size LLx LLy URx URy: specifies molding position.

Example

1. Create a new molding part MOLD_PART at <-5 -5 5 5>

channel setup part -add -name MOLD_PART -type molding -size -5 -5 5 5 -model fccsp

TCL to Delete Part. The syntax is

channel setup part -del -name <part list> -model <model_name>

where,

-name <part list>: specifies part list to delete.

TCL to rename part

layout setup part -name <Name1> -newname <Name2> -model <Model Name>

Example

layout setup part -name U1 -newname Die -model Package

---

layout setup components

layout setup component -name <componentName> ? -mount <layerName>? ?-position [above|below]? ?-dx <value>? ?-dy <value>? ?-rot <rotationAngle>? –include [1|0] -model<model_name>

where,

-name <componentname>: specifies component(s).

-mount <layerName>: specifies a mounting layer for a component.

-position [above|below]: specifies mounting direction.

-dx <value>: specifies offset size along the x direction.

-dy <value>: specifies offset size along the y direction.

-rot <rotationAngle>: specifies rotation angle.

-include [true(1) | false(0)]: determines whether to include the component in package extraction. At least one die and one bga component need to be included.

Examples

1. Specify mounting layer as below M6 for the BGA component

layout setup component -name BGA –mount M6 –position below -model fccsp

2. Dis-include few cap. components

layout setup component –name C1 C2 C3 –include 0 -model fccsp

TCL to Create Molding Component. The syntax is

layout setup component ?-add? ?-name <component> ?-part <partname>? ?-layer <layername>?

where,

-model <model_name>

-add: specifies to add a new component;

-name component: specifies the name of molding component;

-part partname: specifies the name of molding part;

-layer layername: specifies the mounting layer for molding component.

Example

create a new molding component Molding mounting on TOP layer

layout setup component -add -name Molding -part MOLD_PART -layer TOP -model fccsp

TCL to Delete Component. The syntax is

layout setup component -del -name <component list> -model <model_name>

where,

-name <component list>: specifies component list to delete

Example

layout setup -model fccsp -del -name BGA

---

layout setup material

layout setup material -add -type [conductor | dielectric] -name <material_name> ?-p <relative_permittivity>? ?-m <relative_permeability>? ?-l <loss_tangent>? ?-c <conductivity>? -model <model_name>

where,

-add: currently, it is only allowed to add a material. This option is required.

-type [conductor | dielectric]: specifies the material type. For conductor, only conductivity is needed. For dielectric materials, relative permittivity, relative permeability, loss tangent and conductivity need to be specified.

-name <material_name>: specifies the material name.

-p <relative_permittivity>: is required for dielectric material.

-m <relative_permeability>: applied to dielectric material. If not specified, the default value is 1.

-l <loss_tangent>: applied to dielectric material. If not specified, the default value is 0.

-c <conductivity>: is required for conductor material. For dielectric material, if not specified, the default value is 0. The unit is S/mm.

Examples:

1. Add a new material “my_copper”.

layout setup material -add -type conductor -name my_copper -c 59600 -model fccsp

2. Add a new material “my_FR4”.

layout setup material -add -type dielectric -name my_FR4 -p 4.7 -m 1 -l 0.018 –c 0 -model fccsp

TCL command to specify temperature coefficients and reference temperature.

layout setup material -name <all | material_name> [-rt1 <float> -rt2 <float> -tref <t>]

where,

-name material_name: specify the conductor material that it need the modeling to be applied.

-rt1 <float>: specifies temperature coefficient rt1. Default value of rt1 is zero.

-rt2 <float>: specifies temperature coefficient rt2. Default value of rt2 is zero.

-tref <t>: specifies reference temperature in degree Centigrade.

Example

layout setup material -model interposer -name GDS_MB -thermal -ts 30 -k 0.00124 -density 1.1 -cp 0.8 -e 8 -etc 12 -pr 0.02 -nu 0.02 -g 2.45

TCL command to import strain-stress curve

layout setup material -name <name> -model <model> -plastic -file <file>

Example

layout setup material -name COPPER -model PCB -plastic -file /home/user/StrainStressCurves.txt

---

layout setup layer

layout setup layer –name <layername> ?-type [dielectric | signal | power]? ?-thickness <value>? ?-cond_mat <material_name>? ?-diel_mat <material_name>? -model <model_name>

where,

-name <layername>: specifies an existing layer.

-type [dielectric | signal | power]: specifies/changes the layer type.

-thickness <value>: specifies the thickness of a layer. The unit is mm.

-cond_mat <material_name>: specifies the conducting material for the layer.

-diel_mat <material_name>: specifies the dielectric material for the layer.

Examples:

1. Modify the thickness of a dielectric layer, and specifies its type and material.

layout setup layer -name Diel_1 -type dielectric -thickness 0.35 -model fccsp

2. Specifies its material of a signal layer.

layout setup layer -name TOP -type signal -cond_mat COPPER -diel_mat AIR -model fccsp

---

layout setup stackup

layout setup layer [-add | -del] -name <layername> ?[-above | -below] -newname <layername>? -model <model_name>

where,

[-add | -del]: to add or delete a layer from the stackup.

-name <layername>: specifies the layer name to be added from or deleted.

[-above | -below]: determines to add above or below the specified layer. This option is required for adding a layer.

-newname <layername>: specifies the name for the added layer. This option is required for adding a layer.

layout setup layer -del ?-name <layername_list>? -model <model_name>

where,

-del: to delete layers from the stackup

Examples:

1. Add a layer above “TOP” layer.

layout setup layer -add -name TOP -above -newname Diel_0 -model fccsp

2. Delete layers “Die*” “DIE_AP” “T*” “BOTTOM”.

layout setup layer -del -name Die* DIE_AP T* BOTTOM -model fccsp

---

layout setup padstack

syntax: Set Via material for Padstack.

layout setup padstack -name <PadStack Name> -viamaterial <material name> -model <model name>

where,

-name: specify the padstack name.

-viamaterial: specifies the material of padstack.

Examples:

layout setup padstack -name pad1 -viamaterial via_mat1 -model PKG

---

layout setup solder ball

layout setup ball -d1 <value> -d2 <value> -dmax <value> -ht <value>?-material [solder | gold | silver | copper | <material_name>]? -model <model_name>

where,

-d1 <value> -d2 <value> -dmax <value> -ht <value>: specifies the dimension of the solder ball for BGA component. The unit is mm.

-material [solder | gold | silver | copper | <material_name>]: specifies the material of solder ball. It can be the material in the default library or user defined material. If this option is not use, “solder” is used by default.

Examples:

1. Specify the dimension and material of solder ball.

layout setup ball -d1 0.4 -d2 0.4 -dmax 0.5 -ht 0.5 -material solder -model fccsp

---

layout setup solder bump

layout setup bump -d1 <value> -d2 <value> -dmax <value> -ht <value>?-material [solder | gold | silver | copper | <material_name>]? ?-component [<component_name> | all]? -model <model_name>

where,

-d1 <value> -d2 <value> -dmax <value> -ht <value>: specifies the dimension of the solder bump for die component. The unit is mm.

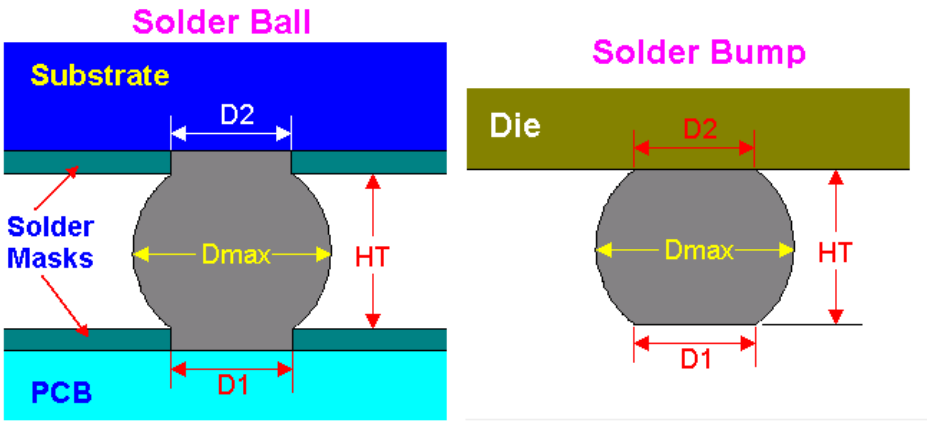
-material [solder | gold | silver | copper | <material_name>]: specifies the material of solder bump. It can be the material in the default library or user defined material. If this option is not use, “gold” is used by default.

-component [<component_name> | all]: specifies the component if multiple die components exist in the design. “-component all” applied the dimension and material of solder ball to all die components.

Examples:

1. Specify the dimension and material of solder bump for “FCHIP”.

layout setup bump -d1 0.08 -d2 0.08 -dmax 0.1 -ht 0.1 -material solder -component FCHIP -model fccsp



---

layout setup bondwire

layout setup bondwire -autowireends -model <model_name>

With system-in-a-package (SiP) applications, the imported 2-D bond wire information does not specify the die connections. The Auto Wire Ends function connects the wires to their proper die pins in three dimensions.

To add and edit the bondwire, use the following command:

Syntax: layout setup bondwire **-add** -name <bondwire_name> -net <net_name> -start <starting location> -end <ending location> -flip <true | false> -profilename <profile_name> -model <model_name>

Example 1

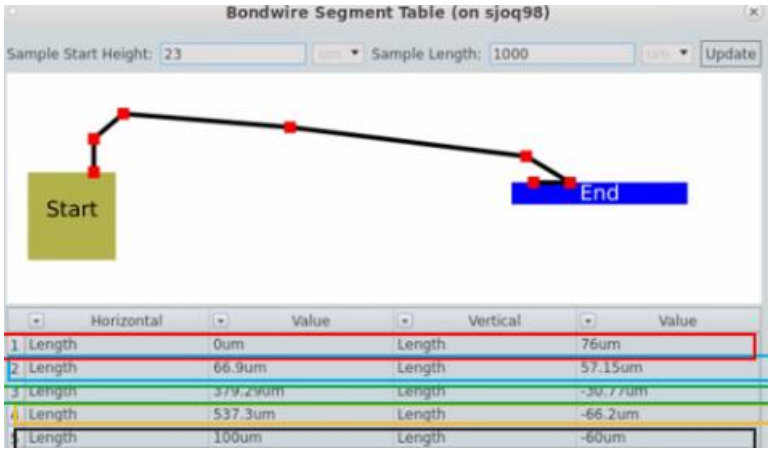
layout setup bondwire -add BW11 -net VDD -start METTOP -end VSS_C1 -flip true -profilename BWProfile1 -model fccp

Example 2

layout setup bwprofile -add -name BWProfile4 -model fccsp

layout setup bwprofile -name BWProfile4 -type NSegment -segment { Angle, 90, Length, 0.178, Length, 0.1, Length, 0.02, Angle, 90, Length, 0.1 } -model fccsp

layout setup bwprofile -name test_wire2 -type NSegment -segment {Length, 0, Length, 0.076, Length, 0.0669, Length, 0.05715, Length, 0.37929, Length, -0.03077, Length, 0.5373, Length, -0.0662, Length, 0.1, Length, -0.06} -model PWR_EXP_wirebond



layout setup bondwire -add -name BW11 -net VDD -profilename BWProfile1 -model fccsp
layout setup bondwire -name BW11 -start TOP -end VSS_C1 -flip true -model fccsp
layout setup bondwire -name BW11 -profilename BWProfile2 -model fccsp

---

layout setup bondwire profile

layout setup bwprofile -name <bwprofile_name> -diameter <value> -material <material_name> -height <value> -model <model_name>

where,

-name <bwprofile_name>: specifies the name of bondwire profile
-diameter <value>: specifies diameter value. Default unit is mm. A data value with unit is allowed.
-material <material_name>: specifies conducting material for bondwire profile.
-height <value>: specifies height value.

Examples

Change the diameter of bondwire profile BWProfile1 to 20um with GOLD material.
layout setup bwprofile -name BWProfile1 -diameter 20um -material GOLD -model fccsp
layout setup bwprofile -name BWProfile1 -diameter 0.02 -material GOLD -model fccsp

To add the bondwire profile, use the following command:
layout setup bwprofile -add -name <bondwireprofile_name> -type <profiletype> -segment {Angle , Length} -model <model_name>

Example

layout setup bwprofile -add -name BWProfile11 -type NSegement -segment {Angle, 90, Length, 0.178, Length, 0.1, Length, 0.02, Angle, 30, Length, 0.1} -model fccp

---

layout prototype create

Description: Create prototype package layout and pcb layout model

Syntax:

layout prototype create –pre <path> -size <low_left_x, low_left_y, up-right_x, up-right_y> -layers <layer_name, thickness, metal_composition>

Example:

layout prototype create -pre ${RUN_DIR}/${NAME_CASE}/sub_pkg.pre -size { -1.893, -0.95,1.893, 0.95 } -layers { TOP, 0.04, 0, 99, D1, 0.1, 1, 1, BOTTOM, 0.04, 0, 99 } -molding { MOLD, -1.9, -1.9, 1.9, 1.9, 0.4 } -solderball {0.5, 10, 0.5, 10 }
layout prototype create -pre ${RUN_DIR}/${NAME_CASE}/pcb.pre -size { -50, -50, 50, 50 } -layers { METAL1, 0.074, 0, 20, VIA1, 0.45, 1, 0.1, METAL2, 0.037, 0, 90, VIA2, 0.45, 1, 0.1, METAL3, 0.037, 0, 90, VIA3, 0.45, 1, 0.1, METAL4, 0.074, 0, 20 }

---

layout special create

layout connection create

Description: Create special component: bump/ball, molding or heatsink.

Syntax:

layout special create -name <model_name> -length <value> -width <value> ?-file <path>?

layout connection create –name <model_name> -length <value> -width <value>

Where

-name: specify layout model name

-length: specify the length of interface size.

-width: specify the width of interface size.

-file: specify the *.ploc, *.con or *.tsv file.

Example:

layout connection create -name BGA -length 4.48 -width 4.48 -file ${RUN_DIR}/design_data/BGA.con

layout special create -name HeatSink -length 5 -width 5

---

layout connection change

Description: specify bump/ball model property.

Syntax:

layout connection change -model <model_name> -add –file <path>

layout connection change -model <model_name> –shift <value>

layout connection change -model <model_name> –thickness <value>

layout connection change -model <model_name> -diemeter <value> -all

layout connection change -remove –index

layout connection change -array -position {x, y} -pitch {x, y} -num {x, y}

layout connection change -size { llx, lly, urx, ury}

Example:

layout connection change -model bump_topdie -add –file $RUN_DIR/design_data/BUMP.ploc

layout connection change -model bump_topdie -shift {-0.3, -0.3}

layout connection change -model bump_topdie -thickness 0.005

layout connection change -model bump_topdie -diemeter 0.022 -all

layout connection change -model DIE_Bump -remove -index { 1, 2, 3, 4 }

layout connection change -model DIE_Bump -array -position { -0.275, -0.277 } -pitch { 0, 0 } -num { 1, 1 }

layout connection change -model DIE_Bump -size { -0.506, -0.507, 0.506, 0.507 }

Off-Chip Model

channel model create

Description: Add generic/s-parameter/prototype channel model

Syntax:

channel model create -type <type_name> ?-name <model_name>?

where

-type <type_name>: Specify model type, generic/sparam/prototype/vrm/rlc/cp

Examples:

channel model create -type generic -name generic1

---

channel model remove

Description: Remove any channel model

Syntax:

channel model remove -model <model_name>

Examples:

channel model remove -name generic1

---

channel model copy

Description: Copy/paste any channel model

Syntax:

channel model copy -model <model_name> -to <new_name>

Examples:

channel model copy -model vrm1 -to vrm1_copy1

---

channel generic set

Description: Import generic spice model file

Syntax:

channel generic set -model <model name> -file <spice model file> -topsubckt <top subckt name>

Examples:

channel generic set -model generic3 -file /home/sso/cpa_rh_pkg_wrapper_ASCII.sp -topsubckt package_CPA

---

channel component set

Description: Create Components

Syntax:

channel component set -model <model name> -comp <component name> -node { node list }

Examples:

channel component set -model generic3 -comp FCHIP_FCHIP -node { FCHIP_VDD_15_Group,

FCHIP_VSS_Group }

---

channel prototype setnodes

Description: Set the net number for prototype channel object.

Syntax:

channel prototype setnodes –model <model_name> -protoobj <prototype_model_name> -type {Signal | Power | Ground} –nodes {node_list} –netnum [value]

Examples:

channel prototype setnodes -model prototype1 -protoobj prototype1 -type Power -nodes { prototype1_P_l1, prototype1_P_r1 } -netnum 1

---

channel prototype addsegment

channel prototype delsegment

Description: Add or delete the segment for prototype channel.

Syntax:

channel prototype [addsegment | delsegment] –model <model_name> -protoobj <prototype_model> [-signal | -power | -ground] –segtype [1|2|3|4] –R {value} ?-L {value}? ?-C {value}? ?-g {value}? ?-w{value}? ?-v {value}? ?–segment {value} –hl {value} –t {value} –cond {value} –Er {value} –lengh {value} –Loss {value}? –geomtype {value}

Examples:

channel prototype addsegment -model prototype1 -protoobj prototype1 -power -segtype 1 -R 10mOhm -L 0.25nH -C 0.1pF -w 60um -segmenth 40um -h1 40um -t 25um -cond 5.8e+07S -Er 4.5 -length 10mm -Loss 0.02 -geomtype 0

---

channel prototype generate

Description: Generate prototype channel model.

Syntax:

channel prototype generate –model <model_name> -protoobj <prototype_model_name>

Examples:

channel prototype generate -model prototype1 -protoobj prototype1

---

channel setup port

Description: setup port

Syntax:

channel setup port -model <model_name> -net <net_name>- pin<pin_list> -pinfile< pin in file > -include [0|1] -type [Coaxial Open|Coaxial Shorted|Gap|Lumped] -refNet <net_name> -zref<xx> -file <setupfile>

---

channel setup general

Description:Specify the general setting.

Syntax:

channel setup general -model <model_name> -UsePSI [1 |0] -AFSErrorTolerance [-35|-45|-55|-65|-75] -AdvancedFreqSweep [1|0] -FastSweep [1|0] -AdaptiveSampling [1|0] -EnforceDC[1|0] frequency 10, 2000,Linear,100,2000,3000,Log,100

---

channel vrm set

Description: Set the vrm model.

Syntax:

channel vrm set -model <model_name> -pins { <pin_name>,<voltage> }

Examples:

channel vrm set -model vrm3 -pins { vdd, 1.5, vss, 0 }

---

channel rlc set

Description: Set the rlc model.

Syntax:

channel rlc set -model<model_name> -pin1 <pin1_name> -pin2 <pin2_name> -usec true/false -usel true/false -user true/false -usev true/false -C <value> -R <value> -L<value> -V <value>

Examples:

channel rlc set -model rlc1 -pin1 pin1 -pin2 pin2 -usec false -usel false -user true -R 10Ohm -usev false

---

channel cp set

Description: Set the cp model which used only for 3DIC setup flow

Syntax:

channel cp set -model <Model name> -R <n Ohm> -L <n pH> -C <n pF> -length <n um> -width <n um> -height <n um>

Examples:

channel cp set -model Model2 -R 1e-05Ohm -L 10pH -C 2pF -length 0.1um -width 0.1um -height 1e-05um

---

channel sparam set

Description: Import the netlist which include touchstone file or touchstone file directly.

Syntax:

channel sparam set -model <model name> -file <spice netlist> -topsubckt <topsubckt name>

Examples:

channel sparam set -model sparam3 -file /home/sso/PKG_Sparam_AC.sp -topsubckt PKG_Sparam_AC_fws

---

channel sparam updateselem

Description: Set s-parameter options.

Syntax:

channel sparam updateselem -model <Model name> -selem pkg -simoption [useoriginal | autogroup] -convopt [convert | handledbysimulator] -simtype [Hspice | Spectre | Eldo | Nexxim | Nexxininternal] -properties {simulator properties} -refnodes [true | false] -hasoldgroup false

Examples:

channel sparam updateselem -model sparam3 -selem pkg -simoption useoriginal -convopt handledbysimulator -simtype Hspice -properties { delayhandle, 0, enforce_passive, 1, intdattyp, 1, rational_func, 1 } -refnodes true

---

channel sparam generate

Description: Generate s-parameter model.

Syntax:

channel sparam generate -model <Model name>

Examples:

channel sparam generate -model sparam3

---

channel cpa import

Description: Creating a CPA model.

Syntax:

channel cpa import -file <package layout file> ?-config <gds2xfl config file> ?-name <model_name>?

Example:

channel cpa import -file /home/sso/analysis.xfl -name analysis1

---

channel report emcheck

Description: to do EM check

Syntax:

channel report emcheck -model <model name> ?[-o <emcheck_file]? ?-i <tech_file>? ?-layer <layer1, layer2, …>? ? -limit <limit1, limit2, …>

Example:

channel report emcheck model Test1

Setup Nets

Syntax:

channel setup net -name [<netname> | all] ?-file <filename>? ? -type [power | ground| signal]? ?-include [true(1) | false(0)]? -model <model_name>

where,

-name <netname> | all: Specifies the nets. <netname> is a list of net names (case sensitive) separated by space. It can use * to indicate any characters (e.g. VDD* stands for all nets whose name starts with VDD). To select all nets, use “-name all”.

-file <filename>: Specifies a file that contains listed net names which are separated by space or newline.

-type [power | ground | signal]: Sets the net type. This option is not required.

-include [true(1) | false(0)]: Specifies whether to include the net(s) in package extraction. This option is not required.

-model <model_name>: Indicate which cpa model.

Examples:

1. Set VDD to be power net, and VSS to be ground net. Include those two nets in CPA extraction.

channel setup net –name VDD –type power –include 1 -model fccsp

channel setup net –name VSS –type ground –include 1 -model fccsp

2. Clear the net selection, and include the nets listed in a file.

channel setup net –name all –include 0 -model fccsp

channel setup net –file PG_nets.txt –include 1 -model fccsp

where PG_nets.info contains a list of net names, e.g.

VDD

VDD_15

VSS

TCL command to combine nets. The syntax is

channel setup net ?-combine? ?-name <netname_list|all>? -model

where,

-combine: Specifies to combine nets

-name <netname_list>: Specifies the net name list to be combined; all nets are allowed.

TCL command to remove degassing holes as void composite. The syntax is:

channel setup net del ?-name <net..> -layer <layer...>? -voidnp <num_points>? -model <model_name>

where,

-name: specifies net name list. If net name does not exist, work on all nets;

-layer: specifies layer name list. If layer is not specified, work on all layers;

-voidnp: specifies number of points (typically 4) of void composite to be deleted.

Setup Parts

channel setup part –name <partname> ?-type [die | resistor | inductor | capacitor | bga |other]? ?-r <R_value>? ?-l <L_value>? ?-c <C_value>? ?-neverflip <true|false>??-netlist <filename>? ?-subckt <subckt_name>? ?-pinmap {pin1, node1, … }??-pingroup [H_division V_division | perpin]? ?-net

<netname>??-thickness <value>? -model <model_name>

where,

-name <partname>: specifies the parts. <partname> can be a list of part names (case sensitive) which are separated by space.

-type [die | resistor | inductor | capacitor | bga |other]: sets the type for the specified parts. This option is not required.

-r <R_value> -l <L_value> -c <C_value>: if the type of a part is resistor, inductor, capacitor or other, this option assigns a generic circuit model to the part with the RLC value. Only a numeric number need to be given as the R/L/C value, and the unit will be determined automatically. Scale symbols are allowed as R/L/C value, e.g. 1m=0.001, 1n=1e-9. Note: if the part has 2 pins, the connection between pins and circuit nodes will be made automatically.

-neverflip <true|false>: specifies NeverFlip is on (true) or off (false).

-netlist <filename>: specifies a circuit model to the part. <filename> can be a .sp file that contains one or more subckt. It is not allowed to mix the use of -netlist option and -r/-l/-c option.

-subckt <subckt_name>: specifies the subckt name when there are multiply subckts defined in the .sp file. It not specified, the first subckt will be used. This option has to be used together with -netlist.

-pinmap {pin1, node1, …}: connects the pins of the part to the nodes of the specified circuit model. Every two entries within the bracket is a pair. In this example, pin1 connects to node1. This option has to be used together with -netlist. Note: if the -pinmap option is not used, but the part has 2 pins and the circuit model has 2 nodes, an attempt of connection will be made.

-pingroup [H_division V_division | perpin]: specifies the pin grouping of a part. There are two usages of this tcl: 1) group by grid. “-pingroup m n” divides the part into an m by n grid, the pins inside one division become one group; 2) group each pin. “-pingroup perpin” creates a group for each pin. Use “-pingroup 1 1” to lump all pins together.

-net <netname>: specifies the net that the pin grouping is applying to. If not specified, the pin grouping is applied to all nets by default. This option has to be used together with -pingroup.

-thickness value: specifies part thickness.

Examples:

1. Specify the type of part “U2” as bga.

channel setup part -name U2 -type bga -model fccsp

2. Set the type of part “DECAP” as capacitor, and assign the RLC value.

channel setup part -name DECAP -type capacitor -r 1m -c 10p -model fccsp

3. Assign a circuit model to part “DECAP”, specify the subckt name and connect and pins to circuit nodes.

channel setup part -name DECAP -netlist decap_model.sp -subckt decap_1 -pinmap {pin1, vdd, pin2,vss} -model fccsp

4.Group the VDD pins of the part “FCHIP” into 3 by 2 grid.

channel setup part -name FCHIP -pingroup 3 2 -net VDD -model fccsp

5.Lump all VSS pins of part “FCHIP”.

channel setup part -name FCHIP -pingroup 1 1 -net VSS -model fccsp

6. Group each pin of part “FCHIP” for all nets.

channel setup part -name FCHIP -pingroup perpin -model fccsp

TCL to Create Molding Part. The syntax is

channel setup part ?-add? -name <partname> ?-type molding? ?-size <LLX LLy URx URy>?

where,

-add: specifies to add a new part;

-name partname: specifies the new added molding part name;

-type molding: specifies molding part type. We only support to create a new molding part by TCL.

-size LLx LLy URx URy: specifies molding position.

Examples

1. Create a new molding part MOLD_PART at <-5 -5 5 5>

channel setup part -add -name MOLD_PART -type molding -size -5 -5 5 5 -model fccsp

TCL to Delete Part. The syntax is

channel setup part -del -name <part list> -model <model_name>

where,

-name <part list>: specifies part list to delete.

TCL to edit pins. The syntax is

channel setup part -name <partname> ?-addPins <name1 x1(mm) y1(mm) name2 x2 y2 ...>? ?-clearPins <name1 name2 ...>? ?-pinsFromPloc <ploc file>? ?-filpPins? -model<model_name>

-name <partname>: specifies the part name.

-addPins: specifies the pins to add

-clearPins: specifies the pins to clear

-pinsFromPloc: specifies to import pins from PLOC file and specifies the coordinate box.

-filpPins: specify to flip pins along the Y-axis.

Setup Components

channel setup component -name <componentName> ? -mount <layerName>? ?-position [above|below]? ?-dx <value>? ?-dy <value>? ?-rot <rotationAngle>? –include [1|0] -model<model_name>

where,

-name <componentname>: specifies component(s).

-mount <layerName>: specifies a mounting layer for a component.

-position [above|below]: specifies mounting direction.

-dx <value>: specifies offset size along the x direction.

-dy <value>: specifies offset size along the y direction.

-rot <rotationAngle>: specifies rotation angle.

-include [true(1) | false(0)]: determines whether to include the component in package extraction. At least one die and one bga component need to be included.

Examples

3. Specify mounting layer as below M6 for the BGA component

channel setup component -name BGA –mount M6 –position below -model fccsp

4. Dis-include few cap. components

channel setup component –name C1 C2 C3 –include 0 -model fccsp

TCL to Create Molding Component. The syntax is

channel setup component ?-add? ?-name <component> ?-part <partname>? ?-layer <layername>?

where,

-model <model_name>

-add: specifies to add a new component;

-name component: specifies the name of molding component;

-part partname: specifies the name of molding part;

-layer layername: specifies the mounting layer for molding component.

Example

Create a new molding component Molding mounting on TOP layer

channel setup component -add -name Molding -part MOLD_PART -layer TOP -model fccsp

TCL to Delete Component. The syntax is

channel setup component -del -name <component list> -model <model_name>

where,

-name <component list>: specifies component list to delete

Setup Material

channel setup material -add -type [conductor | dielectric] -name <material_name> ?-p <relative_permittivity>? ?-m <relative_permeability>? ?-l <loss_tangent>? ?-c <conductivity>? -model <model_name>

where,

-add: currently, it is only allowed to add a material. This option is required.

-type [conductor | dielectric]: specifies the material type. For conductor, only conductivity is needed. For dielectric materials, relative permittivity, relative permeability, loss tangent and conductivity need to be specified.

-name <material_name>: specifies the material name.

-p <relative_permittivity>: is required for dielectric material.

-m <relative_permeability>: applied to dielectric material. If not specified, the default value is 1.

-l <loss_tangent>: applied to dielectric material. If not specified, the default value is 0.

-c <conductivity>: is required for conductor material. For dielectric material, if not specified, the default value is 0. The unit is S/mm.

Examples:

1. Add a new material “my_copper”.

channel setup material -add -type conductor -name my_copper -c 59600 -model fccsp

2. Add a new material “my_FR4”.

channel setup material -add -type dielectric -name my_FR4 -p 4.7 -m 1 -l 0.018 –c 0 -model fccsp

TCL command to specify temperature coefficients and reference temperature.

channel setup material -name <all | material_name> [-rt1 <float> -rt2 <float> -tref <t>]

where,

-name material_name: specify the conductor material that it need the modeling to be applied.

-rt1 <float>: specifies temperature coefficient rt1. Default value of rt1 is zero.

-rt2 <float>: specifies temperature coefficient rt2. Default value of rt2 is zero.

-tref <t>: specifies reference temperature in degree Centigrade.

Setup Layer

channel setup layer –name <layername> ?-type [dielectric | signal | power]? ?-thickness <value>? ?-cond_mat <material_name>? ?-diel_mat <material_name>? -model <model_name>

where,

-name <layername>: specifies an existing layer.

-type [dielectric | signal | power]: specifies/changes the layer type.

-thickness <value>: specifies the thickness of a layer. The unit is mm.

-cond_mat <material_name>: specifies the conducting material for the layer.

-diel_mat <material_name>: specifies the dielectric material for the layer.

Examples:

1. Modify the thickness of a dielectric layer, and specifies its type and material.

channel setup layer -name Diel_1 -type dielectric -thickness 0.35 -model fccsp

2. Specifies its material of a signal layer.

channel setup layer -name TOP -type signal -cond_mat COPPER -diel_mat AIR -model fccsp

Setup Stackup

channel setup layer [-add | -del] -name <layername> ?[-above | -below] -newname <layername>? -model <model_name>

where,

[-add | -del]: to add or delete a layer from the stackup.

-name <layername>: specifies the layer name to be added from or deleted.

[-above | -below]: determines to add above or below the specified layer. This option is required for adding a layer.

-newname <layername>: specifies the name for the added layer. This option is required for adding a layer.

channel setup layer -del ?-name <layername_list>? -model <model_name>

where,

-del: to delete layers from the stackup

Examples:

1. Add a layer above “TOP” layer.

channel setup layer -add -name TOP -above -newname Diel_0 -model fccsp

2. Delete layers “Die*” “DIE_AP” “T*” “BOTTOM”.

channel setup layer -del -name Die* DIE_AP T* BOTTOM -model fccsp

Setup Solder Ball

channel setup ball -d1 <value> -d2 <value> -dmax <value> -ht <value>?-material [solder | gold | silver | copper | <material_name>]? -model <model_name>

where,

-d1 <value> -d2 <value> -dmax <value> -ht <value>: specifies the dimension of the solder ball for BGA component. The unit is mm.

-material [solder | gold | silver | copper | <material_name>]: specifies the material of solder ball. It can be the material in the default library or user defined material. If this option is not use, “solder” is used by default.

Examples:

1. Specify the dimension and material of solder ball.

channel setup ball -d1 0.4 -d2 0.4 -dmax 0.5 -ht 0.5 -material solder -model fccsp

Setup Solder Bump

channel setup bump -d1 <value> -d2 <value> -dmax <value> -ht <value>?-material [solder | gold | silver | copper | <material_name>]? ?-component [<component_name> | all]? -model <model_name>

where,

-d1 <value> -d2 <value> -dmax <value> -ht <value>: specifies the dimension of the solder bump for die component. The unit is mm.

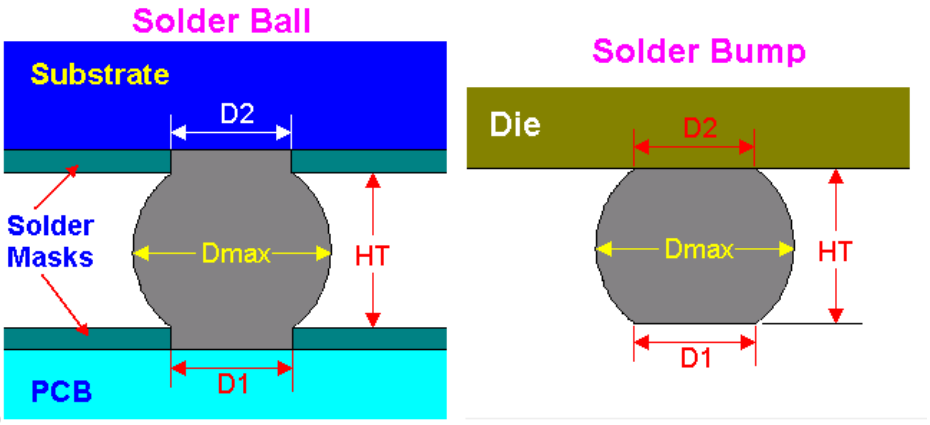
-material [solder | gold | silver | copper | <material_name>]: specifies the material of solder bump. It can be the material in the default library or user defined material. If this option is not use, “gold” is used by default.

-component [<component_name> | all]: specifies the component if multiple die components exist in the design. “-component all” applied the dimension and material of solder ball to all die components.

Examples:

1. Specify the dimension and material of solder bump for “FCHIP”.

channel setup bump -d1 0.08 -d2 0.08 -dmax 0.1 -ht 0.1 -material solder -component FCHIP -model fccsp



Setup Bondwire

channel setup bondwire -autowireends -model <model_name>

With system-in-a-package (SiP) applications, the imported 2-D bond wire information does not specify the die connections. The Auto Wire Ends function connects the wires to their proper die pins in three dimensions.

Setup Bondwire Profile

channel setup bwprofile -name <bwprofile_name> -diameter <value> -material <material_name> -height <value> -model <model_name>

where,

-name <bwprofile_name>: specifies the name of bondwire profile

-diameter <value>: specifies diameter value. Default unit is mm. A data value with unit is allowed.

-material <material_name>: specifies conducting material for bondwire profile.

-height <value>: specifies height value.

Examples

Change the diameter of bondwire profile BWProfile1 to 20um with GOLD material.

channel setup bwprofile -name BWProfile1 -diameter 20um -material GOLD -model fccsp

channel setup bwprofile -name BWProfile1 -diameter 0.02 -material GOLD -model fccsp

Setup External Environment

channel setup env ? –h1 <h1_height>? ? –h2 <h2_height>? [?-ground <1|0>?] [?-mdensity <value>?] -model <model_name>

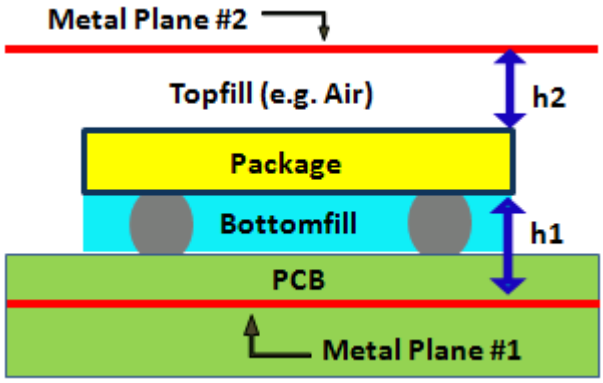
where

–h1 <h1_height>: specifies h1 height

–h2 <h2_height>: specifies h2 height

-ground <1|0>: specifies a ground plane or not;

-mdensity <value>: specifies the plane metal density value.



Setup General Configuration

channel setup general -esd [1|0] -freq <value(MHz)> -cpu <number> -lossmodel [1|2|3] -holes [auto|value(mm)] -Accuracy <Balanced|Accurate> -model <model_name>

where,

-esd [1|0]: specify to extract a pure-resistance ESD model or not.

-freq <value(MHz)>: specifies extraction frequency in MHz. Default frequency = 100 MHz.

-cpu <number>: specifies cpu number. By default, all CPU is used.

-lossmodel [1|2|3]: specifies level of complexity in the loss model.

-holes [auto|value(mm)]: specifies auto-detect small holes; or set a hole diameter value in mm unit

-Accuracy <Balanced|Accurate> : Control the mesh setting

Examples:

1. Specify to extract an ESD model

   channel setup general -esd 1 -model fccsp

2. Specifies extraction frequency = 400 MHz, loss model Level2, auto-detect small holes to ignore during extraction.

   channel setup general -esd 0 -freq 400 -lossmodel 2 -holes auto -model fccsp

3. Specifies extraction frequency = 400 MHz, loss model Level2, ignore holes smaller than 0.05mm

   channel setup general -esd 0 -freq 400 -lossmodel 2 -holes 0.05 -model fccsp

Setup Polygon Merge

channel setup merge ?-circle <discretization_value>? ?-tolerance <size>(mm)? ?-drc [true|false]? ?-autofix [true|false]? ?-pinconn [true|false]? ?-modeltype [PKG | IC | PCB]? -model <model_name>

where,

-circle: number of polygon vertices used to approximate a circle. Default: 12.
-tolerance: Gap below which the polygon merge joins two nearby shapes into one. Default: 1e-5 mm
-drc: sets the polygon merge to perform design rule checking (the default) or not.
-autofix: if enabled, attempts to correct certain error types found by the DRC check
-pinconn: checks for unconnected net segments (the default).
-modeltype: IC for RDL or silicon interposers; PKG for wirebond, flip-chip, etc package designs; PCB for board designs.

Save Modified

channel setup save -model <model_name>
This tcl command saves the imported package layout file and simulation setups into project directory

Perform DRC Check

perform cpa drc -model <model_name>
This is a command to run DRC check for included nets at any time. PLOC connection is not needed.

Perform Channel Extraction

channel perform extraction -model <model_name>
This is a package extraction command that:
1. Save project files to <project_path>;
2. DRC check;
3. Generate package model and interface files that connects package model.

GDS

---

gdsconv project new

Description: Create a GDS conversion project
Syntax:

```
gdsconv project new ?-flow top|cell|mmx? -id <id_num> -name
 <project_name> ?-folder <project_folder>?
```

where

```
-flow: TOP Gds flow or cell gds flow or mmx flow
-id: user must provide one unique number as this project id
-name: project name
-folder: project folder. If empty, current folder will be as project folder
```

Example:

```
     gdsconv project new -id 13 -flow top -name gds_batch
```

---

gdsconv project open

Description: Open a GDS conversion project

Syntax:

```
    gdsconv project open -id <id_num> -name <project_name> ?-folder
    <project_folder>?
where:
    -id: user must provide one unique number as this project id
    -name: project name
    -folder: project folder. If empty, current folder will be as project
folder
```

Example:

```
     gdsconv project open -id 13 -name gds_batch3
```

---

gdsconv project save

Description: Save project.

Syntax:

```
    gdsconv project save -id <id_num>
where:
    -id: the openning project id.
```

Example:

```
     gdsconv project save -id 12
```

---

gdsconv project close

Description: Close project.

Syntax:

```
    gdsconv project close -id <id_num>
where:
    -id: the openning project id.
```

Example:

```
     gdsconv project close -id 12
```

---

gdsconv config file importing

Description: Import a gdsconv config file for case setup.

Syntax:

```
    gdsconv import -id <id_num> -config <config_file>
where:
```

```
    -id: the openning project id.
    -config: gds conversion config file path.
```

Example:

```
     gdsconv import -id 12 -config …/test.config
```

---

gdsconv design set

Description: Set design name for top GDS flow.

Syntax:

```
    gdsconv design set -id <id_num> -cell <design_name>
where:
    -id: the openning project id.
    -cell: cell name
```

Example:

```
     gdsconv design set -id 12 -cell CHIP
```

---

gdsconv xtormap_file set

Description: Set xtor map file for mmx flow.

Syntax:

```
    gdsconv xtormap_file set -id <id_num> -file <xtor_map_file>
where:
    -id: the openning project id.
    -file: xtor map file path
```

Example:

```
     gdsconv xtormap_file set -id 12 -file …/test.gdsmmx
```

---

gdsconv tech_file set

Description: Set tech file for top gds flow with TSV or mmx flow

Syntax:

```
    gdsconv tech_file set -id <id_num> -file <technology_file>
where:
    -id: the openning project id.
    -file: apache tech file path
```

Example:

```
     gdsconv tech_file set -id 12 -file …/test.tech
```

---

gdsconv layer_file set

Description: Set layer mapping file.

Syntax:

```
    gdsconv layer_file set -id <id_num> -file <layer_mapping_file>
where:
    -id: the openning project id.
    -file: file name
```

Example:

```
     gdsconv layer_file set -id 12 -file …/layout.map
```

---

gdsconv gds_file add

Description: Add gds file.

Syntax:

```
    gdsconv gds_file add -id <id_num> -file <gds_file>
where:
    -id: the openning project id.
    -file: file name
```

Example:

```
     gdsconv gds_file add -id 12 -file …/layout.gds
```

---

gdsconv gds_file delete

Description: Delete gds file.

Syntax:

```
    gdsconv gds_file delete -id <id_num> -file <gds_file>
where:
    -id: the openning project id.
    -file: file name
```

Example:

```
     gdsconv gds_file delete -id 12 -file …/layout.gds
```

---

gdsconv casis_file add

Description: Add OASIS file.

Syntax:

```
    gdsconv casis_file add -id <id_num> -file <casis_file>
where:
    -id: the openning project id.
    -file: OASIS file name
```

Example:

```
     gdsconv casis_file add -id 12 -file …/casis file
```

---

gdsconv casis_file delete

Description: Delete OASIS file.

Syntax:

```
    gdsconv casis_file delete -id <id_num> -file <casis_file>
where:
    -id: the openning project id.
    -file: OASIS file name
```

Example:

```
     gdsconv casis_file delete -id 12 -file …/casis file
```

---

gdsconv lef_file add

Description: Add LEF file.

Syntax:

```
    gdsconv lef_file add -id <id_num> -file <casis_file>
where:
    -id: the openning project id.
    -file: LEF file name
```

Example:

```
     gdsconv lef_file add -id 12 -file …/deq.lef
```

---

gdsconv lef_file delete

Description: Delete LEF file.

Syntax:

```
    gdsconv lef_file delete -id <id_num> -file <casis_file>
where:
    -id: the openning project id.
    -file: LEF file name
```

Example:

```
     gdsconv lef_file delete -id 12 -file …/dq.lef
```

---

gdsconv ignore_cell add

Description: Add ignored cell.

Syntax:

```
    gdsconv ignore_cell add -id <id_num> -cell {cell list}
where:
    -id: the openning project id.
    -cell: ignored cell list.
```

Example:

```
     gdsconv ignore_cell add -id 12 -cell buf
```

---

gdsconv ignore_cell delete

Description: Delete ignored cell.

Syntax:

```
    gdsconv ignore_cell delete -id <id_num> -cell {cell list}
where:
    -id: the openning project id.
    -cell: ignored cell list.
```

Example:

```
     gdsconv ignore_cell delete -id 12 -cell buf
```

---

gdsconv target_cell add

Description: Add target cell.

Syntax:

```
    gdsconv target_cell add -id <id_num> ?-regen? -cell {cell list}
where:
    -id: the openning project id.
    -regen: if has this option, the {cell list} will be all regenerated.
    -cell: target cell list.
```

Example:

```
     gdsconv target_cell add -id 12 -cell buf
```

---

gdsconv target_cell delete

Description: Delete target cell.

Syntax:

```
    gdsconv target_cell delete -id <id_num> ?-regen? -cell {cell list}
where:
    -id: the openning project id.
    -regen: if has this option, the {cell list} will be all regenerated.
    -cell: target cell list.
```

Example:

```
     gdsconv target_cell delete -id 12 -cell buf
```

---

gdsconv vdd_net add

Description: Add vdd net.

Syntax:

```
    gdsconv vdd_net add -id <id_num> -net {net list}
where:
```

```
    -id: the openning project id.
    -net: net list.
```

Example:

```
     gdsconv vdd_net add -id 12 -net VDD
```

---

gdsconv vdd_net delete

Description: Delete vdd net.

Syntax:

```
    gdsconv vdd_net delete -id <id_num> -net {net list}
where:
    -id: the openning project id.
    -net: net list.
```

Example:

```
     gdsconv vdd_net delete -id 12 -net VDD
```

---

gdsconv gnd_net add

Description: Add gnd net.

Syntax:

```
    gdsconv gnd_net add -id <id_num> -net {net list}
where:
    -id: the openning project id.
    -net: net list.
```

Example:

```
     gdsconv gnd_net add -id 12 VSS
```

---

gdsconv gnd_net delete

Description: Delete gnd net.

Syntax:

```
    gdsconv gnd_net delete -id <id_num> -net {net list}
where:
    -id: the openning project id.
    -net: net list.
```

Example:

```
     gdsconv gnd_net delete -id 12 VSS
```

---

gdsconv signal_net add

Description: Add signal net.

Syntax:

```
    gdsconv signal_net add -id <id_num> -net {net list}
where:
    -id: the openning project id.
    -net: net list.
```

Example:

```
     gdsconv signal_net add -id 12 DQS0
```

---

gdsconv signal_net delete

Description: Delete signal net.

Syntax:

```
    gdsconv signal_net delete -id <id_num> -net {net list}
where:
    -id: the openning project id.
    -net: net list.
```

Example:

```
     gdsconv signal_net delete -id 12 -net DQS0
```

---

gdsconv advopt

Description: Set advance options for GDS flow.

Syntax:

```
    gdsconv advopt set -id <id_num> -append_conf {append_file ? -
net_from_top_level_text <True|False>? ?-net_from_lef_pin <True|False>? ? -
check_net_short <True|False>? ?-uniquify_target_cell <True| False>? ?-
merge_internal_net <True|False>? ?-pin_creation_text_label <True|False>? ?-
pin_text_label <pin_text_label>? ?-pin_creation_layer <True|False>? ?-
pin_layer <pin_layer_name>? ?-pin_apm_layer <True|False>? ?-pin_layer
<pin_layer_name>? ?-compress <True|False>? ?-speedup <True|False>?
where:
    -id: the openning project id.
    -append_file: appended filename.
    -net_from_top_level_text: net extract from top level text.
    -net_from_lef_pin: net extract from LEF pins.
    -check_net_short: check whether net is shorted or not.
    -net_case_sensitive: net name case sensitive.
    -uniquify_target_cell: set uniquify target cell for Top GDS flow.
    -merge_internal_net: set merge internal net for Top GDS flow.
    -pin_creation_text_label: enable or disable pin creation text label for
   Top GDS flow.
    -pin_text_label: set pin creation text label for Top GDS flow.
    -pin_creation_layer: enable or disable pin creation layer for Top GDS
flow.
```

```
    -pin_layer: set pin creation layer name for Top GDS flow.
    -pin_apm_layer: enable or disable pin APM layer for Cell GDS flow.
    -pin_layer: set pin APM layer name for Cell GDS flow.
    -compress: compress the generated DEF file.
    -speedup: select multi-threading method to speed up extraction.
```

---

gdsconv convert

Description: Generate DEF and LEF from GDS file

Syntax:

```
    gdsconv convert -id <id_num>
where:
    -id: the openning project id.
```

Example:

```
    gdsconv  convert -id 12
```

Die (postlayout)

---

diemodel help

Description: Used in the GUI. When you type this command in the TCL Window, the Message Window displays help information about the `diemodel` TCL command syntax and usage.

Syntax:

```
    diemodel <-h|-help>
```

Example:

```
    diemodel -h
```

---

diemodel set

Description: Specifies the model type for the die.

Syntax:

```
    diemodel set -die <die_name> -type <postlayout|prelayout|generic>
```

Example:

```
    diemodel set -die Die1 -type postlayout
```

---

diemodel rename

Description: Renames the die model.

Syntax:

```
    diemodel rename -die <die_name> -name <new_die_name>
```

Example:

```
diemodel rename -die Die1 -name NewDie
```

---

diemodel info

Description: Used in the GUI. When you type this command in the TCL Window, the Message Window displays the die model type.

Syntax:

```
diemodel info -die <die_name>
```

Example:

```
diemodel info -die Die1
```

---

diemodel postlayout create

Description: Create a postlayout die model.

Syntax:

```
diemodel postlayout create -name <die_name>
```

Example

```
diemodel postlayout create -name Die1
```

---

diemodel postlayout setgeom

Description: Set the geometry of the postlayout die model.

Syntax:

```
diemodel postlayout setgeom -die <die_name> -length <value um> -width
<value um> -height <value um>
```

Example

```
diemodel postlayout setgeom -die INTERPOSER -length 4480um -width 4480um
-height 150um
```

---

diemodel postlayout import

Description: Imports a RedHawk *.gsr* file for the postlayout die model.

Syntax:

```
diemodel postlayout import -die <die_name> -gsr <gsr_file>
```

Example:

```
diemodel postlayout import -die Die1 -gsr /home/projects/generic.gsr
```

---

diemodel postlayout updatedb

Description: Updates the data for the postlayout die model after importing gsr, lef/def, etc. Once you import a new file such as gsr, lef, def, ploc, you shoud use this cmd to update data.

Syntax:

```
diemodel postlayout updatedb -die <die_name>
```

Example:

```
diemodel postlayout import -die Die1 -gsr /home/projects/generic.gsr
diemodel postlayout updatedb –die Die1
```

---

diemodel postlayout info

Description: Used in the GUI. When you type this command in the TCL Window, the Message Window displays all or selected information about the post-layout die model.Default: `all`

Syntax:

```
diemodel postlayout info -die <die_name> ?[all|def|lef|techFile|topDef
|techLef|pgarc|masterCell|vddNets|gndNets|pgcell|sloc|ploc|gdscell|
rdlcell|rdlpgmap|iotype|iocell]?
```

; or

```
diemodel postlayout info -die
<die_name> ?[available_all|available_pgcell|available_vddnet|availabe_gndnet
|available_pgarccell|available_iocell|available_ioinst]?
```

Example:

```
diemodel postlayout info -die Die1 def lef available_all
```

---

diemodel postlayout addData

Description: Adds the specified files to the named die model.

Syntax:

```
diemodel postlayout addData -die <die_name> -tech <tech_file> ?-techtype
<N/A|7nm|10nm|14nm|16nm|20nm>? {<def_file1><def_file2> ...}? ?-lef
{<lef_file1><lef_file2>  -swfile {<switch_model_file>…}
```

Examples:

```
diemodel postlayout addData -die Die1 -tech home/sso/generic.tech
```

---

diemodel postlayout delData

Description: Deletes the specified files from the named die model.

Syntax:

```
diemodel postlayout delData -die <die_name> ?-tech <tech_file>?
?-def {<def_file1><def_file2> ...}? ?-lef {<lef_file1><lef_file2> ...}? -swfile
{<switch_model_file>…}
```

Examples:

```
diemodel postlayout delData -die Die1 -tech /home/sso/generic.tech
```

diemodel postlayout set

diemodel postlayout unset

Description: Sets or unsets *def_file* as the top DEF and sets or unsets *lef_file* as the technology LEF.

Syntax:

```
diemodel postlayout set -die <die_name> -topdef <topdef_file> -techlef
<techlef_file> ?-techtype <N/A|7nm|10nm|14nm|16nm|20nm>?
diemodel postlayout unset -die <die_name> -topdef –techlef
```

Example:

```
diemodel postlayout set -die Die1 -topdef /home/projects/sso/generic.def
-techlef /home/projects/sso/generic.lef
```

---

diemodel postlayout addploc

diemodel postlayout delploc

Description: Adds or deletes ploc pins in die IO Layout view.

Syntax:

```
diemodel postlayout [addploc|delploc] -die <die_name> -ploc {ploc list}
-file <file_name> -force
```

where

`-ploc` : Specify the ploc list.

`-file` : Specify one file it has ploc. One line one column.

`-force` : Specify whether it need to check ploc validation or not.

---

diemodel postlayout setploctype

diemodel postlayout plocfilter

Description: Controls Ploc generation. `setploctype` selects generation method. `plocfilter` configures Ploc generation from top DEF/RDL layer.

Syntax for setploctype:

```
diemodel postlayout setploctype -die <die_name>
-type [plocfile|deffile|pgcell]
```

where

`plocfile` : Uses a user-specified Ploc file.

`deffile` : Generates from top DEF/RDL (PIN section).

`pgcell` : Generates from user-specified P/G cells.

Syntax for plocfilter (when `deffile` selected)**:**

```
diemodel postlayout plocfilter -die <die_name>
```

```
    -userregion [true|false] -llx <x> -lly <y> -urx <x> -ury <y>
    -userlayer [true|false] -layer {<layer1,layer2...}
```

---

diemodel postlayout addPad

diemodel postlayout delPad

Description: Adds or deletes the sloc to post-layout die model.

Syntax:

```
    diemodel postlayout [addpad|delpad] -die <die_name> -sloc <sloc_file> -
ploc <ploc_file> -pgcell <pgcells>
```

Examples:

(1) `diemodel postlayout addpad -die Die1 -sloc /home/ssoqa/data/gds.sloc –pgcell {PAD50R_IN, PAD50R_MDL, PAD50R_OUT}`

(2) `diemodel postlayout delpad -die Die1 -sloc /home/ssoqa/data/gds.sloc –pgcell {PAD50R_IN, PAD50R_MDL, PAD50R_OUT}`

---

diemodel postlayout genploc

Description: If user did not specify the ploc file in the “Edit Data” dialog and have selected some PG cells in that tab, then using this cmd to generate a PLOC file.

Syntax:

```
    diemodel postlayout genploc -die <die_name>
```

Example:

```
    diemodel postlayout genploc -die Die1
```

---

diemodel postlayout createinterface

Description: setup the interface for thermal flow.

Syntax:

diemodel postlayout createinterface –die <die_name> -interfacename {interface1, interface2, ...} - interfacelayer {top, top,...} - interfacearea {bottom_left_x1; bottom_left_y1; top_right_x1; top_right_y1 bottom_left_x2; bottom_left_y2; top_right_x2; top_right_y2 ...}

Example:

diemodel postlayout createinterface -die INTERPOSER –interfacename {die1_interface die2_interface die4_interface die3_interface top_interface bot_interface} -interfacelayer {top top top top top bot} -interfacearea {929;2542;1939;3556 2239;2542;3249;3556 929;1228;1939;2242 2239;1228;3249;2242 NA NA}

---

diemodel postlayout addnet

diemodel postlayout delnet

Description: Adds or deletes the power/ground net to post-layout die model.

Syntax:

```
diemodel postlayout [addnet|delnet] -die <die_name>
[?-vdd {<net1> <net2> ...}? ?-gnd {<net3> <net4> ...}?|
?-vdd <net1> -equivalent {<net2> <net3> ...}?|
?-gnd <net1> -equivalent {<net2> <net3> ...}?]
```

Examples:

(1) `diemodel postlayout addnet -die Die1 -vdd VDD -gnd VSS`

(2) `diemodel postlayout delnet -die Die1 -vdd atestv -equivalent {vdd,vpp}`

---

diemodel postlayout setpgarc

diemodel postlayout addpgarc

diemodel postlayout delpgarc

Description: Sets, adds or deletes PG Arc power/ground pin mapping globally or for local cell. If cell not specified, pin pair used globally. `setpgarc` first clears previous settings.

Syntax:

```
diemodel postlayout [setpgarc|addpgarc|delpgarc] -die <die_name>
?-cell <cell_name>? -pinpairs {<power_pin>, <ground_pin>}
```

Example:

```
diemodel postlayout addpgarc -die Die1 -global -pinpairs {VDD,VSS}
```

---

diemodel postlayout settsv

Description: Set tsv model for thermal flow.

Syntax:

```
diemodel postlayout settsv -die <die_name> -tsv thermal -file <tsv_file>
```

Example:

```
diemodel postlayout settsv -die INTERPOSER -tsv thermal -file
${RUN_DIR}/design_data/INTERPOSER_thermal.tsv
```

---

diemodel postlayout setgds

diemodel postlayout addgds

diemodel postlayout delgds

Description: Set, add or delete GDS cells to or from a specified project folder. `setgds` first clears

previous settings.

Syntax:

```
diemodel postlayout [setgds|addgds|delgds]
-die <die_name> -folder <path/folder_name> ?-mmx? -cell {<cell1>
<cell2> ...}
```

Example:

```
diemodel postlayout delgds -die Die1
-folder /nfs /sso/Data/gds2def/gdsout_io -cell CAP
```

---

diemodel postlayout setrdlcell

Description: Imports redistribution layer (RDL) cell as a DEF block and sets its orientation, (x, y) location, and (x, y) offset distance relative to the top DEF coordinate system. (x, y) units in microns; defaults: (0,0).

Syntax:

```
diemodel postlayout setrdlcell -die <die_name> -def <DEF_file>
?-orientation [N|S|W|E|FN|FS|FW|FE]? ?-lx <x_location>?
?-ly <y_location>? ?-ox <x_offset>? ?-oy <y_offset>?
```

where

`-orientation` : Specifies one of the following orientations relative to theoriginal cell definition and remaining at the same cell location:

N – North, as defined (default)

S – South, rotated 180°

W – West, rotated 90° counter-clockwise (CCW)

E – East, rotated 90° clockwise (CW)

FN – North orientation flipped east-west

FS – South orientation flipped east-west

FW – West orientation flipped east-west

FE – East orientation flipped east-west

---

diemodel postlayout setrdlpin

diemodel postlayout addrdlpin

diemodel postlayout delrdlpin

Description: Specifies, adds or deletes connections of RDL cell pins to top-level nets. `setrdlpin` first clears any previous pin-to-net mapping.

Syntax:

```
diemodel postlayout [setrdlpin|addrdlpin|delrdlpin]
-die <die_name> -pinpairs {<net1> <pin1> <net2> <pin2> ...}
```

Example:

```
diemodel postlayout addrdlpin -die Die1 -pinpairs {VDD VDD VSS VSS}
```

---

diemodel postlayout addswcell

diemodel postlayout delswcell

diemodel postlayout editswcell

Description: Adds, deletes or edits a switch cell to post-layout die model.

Syntax:

```
diemodel postlayout [addswcell|delswcell] -die <die_name>
-cell <cell_list>
diemodel postlayout editswcell -die <die_name> -cell <cell_name>
-type [Header|Footer] -extpin [external pin] -intpin [internal pin]
-ron <double>
```

---

diemodel postlayout setinst

diemodel postlayout addinst

diemodel postlayout delinst

Description: Sets, adds or deletes instance to post-layout die model. `set` first clears the previous selection.

Syntax:

```
diemodel postlayout setinst -die <die_name>
-inst {inst list} -file <file_name> -force
diemodel postlayout [addinst|delinst] -die <die_name> ?-probe?
-inst {inst list} -file <file_name> -force
```

where

`-probe`：To make instance probed.

Examples:

(1) `diemodel postlayout setinst -die Die1 -inst {PAD1 PAD3 PAD4}`

(2) `diemodel postlayout addinst -die Die1 -inst {PAD1 PAD3 PAD4}`

---

diemodel postlayout extract

Description: Extracts the named die. Completes the data setup before using this command. Set the DMP options.

Syntax:

```
diemodel postlayout extract -die <die_name> -mode [R|RC|RLC] -incSignal
<true|false> -cellPinConnect <true|false> -temp <temperature value> -gsr
<file_name> -fao <file_name>-dmp on/off -dmp_import <file> -dmp_create -
dmp_job <number> -dmp_grid <SSH|LSF|RTDA|SGE> -dmp_host <host> -dmp_queue
<xxx> -dmp_large <xxx> -dmp_small <xxx> -dmp_palenv <xxx> -dmp_append
<file>
```

where

`-incSignal [on|off]`: Specifies a single-pass extraction.

`-temp`: Specify the temperature value.

`-gsr`: Specify the gsr file path.

`-cellPinConnect [on|off]`: Specifies that cell pins connect internally.

`-mode [R|RC|RLC]`: Specifies the extraction model type.

`-fao <`*path/filename*`>`: Specifies the FAO file.

```
-dmp [on|off]: Enable or disable DMP extraction.
-dmp_import: Specify DMP external config file. this option can not exist
with -dmp_create: Specify DMP config file created by configuration
-dmp_job: Specify DMP job number
-dmp_grid: Specify DMP grid type, SSH | LSF | RTDA | SGE
-dmp_host: Specify candidate hosts for SSH or LSF type. notes multiple
           hots is splitted by char.
-dmp_queue: Specify DMP queue arguments for LSF/RTDA/SGE
-dmp_large: Specify DMP arguments for large jobs of LSF/RTDA/SGE
-dmp_small: Specify DMP arguments for small jobs of LSF/RTDA/SGE
-dmp_palenv: Specify DMP parallel environment for SGE
-dmp_append: Append one external file into DMP config config
```

Example:

```
diemodel postlayout extract -die Die1 -mode RC -incSignal off
-cellPinConnect off
diemodel postlayout extract -die Die1 -dmp on
diemodel postlayout extract -die Die1 -dmp on -dmp_job 4 -dmp_grid lsf -
dmp_small 2
```

---

diemodel postlayout view

Description: Run view physical design after finishing data setting.

Syntax:

```
diemodel postlayout view -die <die_name>
```

where

```
-die: specify die name
```

Examples:

(1) `diemodel postlayout view -die Die1`

Pre-Layout Die

---

predie model create

predie model delete

Description: Create or delete pre-layout die model.

Syntax:

predie model create -model <model_name>

predie model delete -model <model_name>

Example:

predie model create -model Pre-Die1

predie model delete -model Pre-Die1

---

predie model addinst

Description: Add instances to pre-layout die model

Syntax:

predie model addinst -model <predie_name> -cell <cell_name> -prefix <format> -start <start_number> -end <end_number>

Where:

-prefix: specifies format of instance, you can specify four types: Name_%N, Name[%N], Name<%N> and Name%N.

Example:

predie model addinst -model Pre-Die1 -cell PrototypeCell1 -prefix rec_%N -start 1 -end 4

---

predie model delinst

Description: Delete instance.

Syntax:

predie model delinst -model <predie_name> -cell <cell_name> -inst {instance list}

Example:

predie model delinst -model Pre-Die1 -cell dqbuff -inst {rec_1, rec2, rec_4}

---

predie model setmodel

predie model diesetup

Description: Setting pre-layout die model

Syntax:

predie model setmodel -model <predie_name> ?-driver|-receiver? ?-activemodel <model_name>? -inst {instance list}

predie model diesetup -model <predie_name> -xtor|-ciom|-ibis <instance list>

Example:

predie model setmodel -inst {dri_1, dri_2} –driver -model Pre-Die1

predie model diesetup -xtor {dri_1, dri_2} –driver -model Pre-Die1

predie model setmodel –inst {dri_1, dri_2} -activemodel Model3 -model Pre-Die1

---

predie model cellsetup

Description: Setting cell PG pin

Syntax:

predie model cellsetup -model <predie_name> -cell <cell_name> -pgpin {PG pin list} ?-activemodel <model_name>? ?-probepin <pin name>

Example:

predie model cellsetup -model Pre-Die1 –cell dqbuff –pgpin {vdd, vss} –activemodel Model2

---

predie model generate

Description: Generate pre-layout die model

Syntax:

predie model generate -model <predie_name> ?-folder <folder_name>?

Example:

predie model generate -model Pre-Die1

I/O Circuit

---

iocircuit device info

Description: Used in the GUI. When you type this command in the TCL Window, the Message Window displays information about the I/O model: the file containing the model and the performance corners if listed; current device information if no files specified.

Syntax:

```
iocircuit device info ?die<die_name>? -file {file_list}
```

---

iocircuit device set

iocircuit device add

iocircuit device delete

Description: Sets, adds or deletes an I/O cell device model and sets process corners. You can import the device model separately or attached to the die model.

Syntax for set:

```
iocircuit device set ?-die <die_name>? -file <path/filename>
```

```
-corner <corners>                                                    ; or
iocircuit device set -die <dieName> -cell <cell_name> -model <model_name>
-file <filename> -corner <corners>
```

Syntax for add:

```
iocircuit device add ?-die <die_name>? -file <path/filename>
-corner <corners>
```

Syntax for delete:

```
iocircuit device delete ?-die <die_name>? -file <path/filename>
-corner <corners>
```

Examples:

(1) `iocircuit device set -file /home/projects/sso/de0231.cnr -corner TT`

(2) `iocircuit device add -file /home/projects/sso/generic.cnr`

(3) `iocircuit device delete -file /home/projects/sso/de0231.cnr -corner TT`

---

iocircuit netlist info

Description: Shows imported netlist and simulation cells in the Message Window.

Syntax:

```
iocircuit netlist info ?-die <dieName>? -file < file_list>
```

---

iocircuit netlist set

iocircuit netlist add

iocircuit netlist delete

Description: Sets or deletes an I/O cell netlist. You can import the netlists separately or attached to the die model.

Syntax for set:

```
iocircuit netlist set ?-die <die_name>? -file <path/filename>
-cell <cell_list>
```

Syntax for add:

```
iocircuit netlist add ?-die <die_name>? -file <path/filename>
-cell <cell_list>
```

Syntax for delete:

```
iocircuit netlist delete ?-die <die_name>? -file <path/filename>        ; or
iocircuit netlist delete ?-die <die_name>? -cell <cell_list>
```

Example:

```
iocircuit netlist set -file /home/sso/generic/dqbuff.inc
-cell dqbuff
```

---

iocircuit cell create

iocircuit cell delete

Description: Creates or deletes an I/O cell.

Syntax:

```
iocircuit cell [create|delete] ?-die <die_name>? -cell <cell_name>
```

Example:

```
iocircuit cell create -die die1 -cell dqbuff
```

---

iocircuit model create

iocircuit model rename

iocircuit model delete

Description: Creates, renames or deletes a driver or receiver type I/O cell model.

Syntax for create:

```
iocircuit model create ?-die <die_name>? -cell <cell_name>
[-driver|-receiver] -model <model_name>
```

Syntax for rename:

```
iocircuit model rename ?-die <die_name>? -cell <cell_name>? -model
<model_name> -name <new_model_name>
```

Syntax for delete:

```
iocircuit model delete ?-die <die_name>? -cell <cell_name>
-model <model_name>
```

Examples:

(1) `iocircuit model create -cell dqbuff -driver -model Model1`

(2) `iocircuit model rename -cell dqbuff -model Model1 -name Model2`

(3) `iocircuit model delete -cell dqbuff -model Model2`

---

iocircuit model importbias

iocircuit model deletebias

Description: Imports or deletes a bias file.

Syntax for importbias:

```
iocircuit model importbias ?-die <die_name>? -cell <cell_name>
-model <model_name> -bias <bias_file> -extra <extra_bias_file>
```

where

`-extra`: Specifies the extra bias file in “Advanced Option” dialog of “Model control”

Syntax for deletebias:

```
iocircuit model deletebias ?-die <die_name>? -cell <cell_name>
-model <model_name>
```

Examples:

```
    iocircuit model importbias -cell dqbuff -model Model1
    -bias /home/projects/sso/generic/bias/dqbuff_bias.sp
```

---

iocircuit model control

Description: Controls the macro model generation. Sets the I/O cell parameters, SPICE simulator settings, convergence/accuracy options, core P/G modeling, and hierarchy scaling.

Syntax1:

```
    iocircuit model control ?-die <die_name>? -cell <cell_name>
    -model <mdlname> -simType [hspice|eldo|generic|spectre|lynxspice]
    -simPath <path> ?-simOption <path/filename>? ?-temperature <num>degC?
    ?-freq <num>hz? ?-vl <num>v? ?-vh <num>v? ?-tr <num>ns? ?-tf <num>ns?
    ?-optimtype <0|1|2>? ?-acculevel <num>? ?-usescale [on|off]?
    ?-scaleval <num>? ?-tsettle <settle_time ns>? ?-corePG [on|off]?  ?-
coreP {pin list}?
    ?-coreG {pin list}?
```

Syntax2 for global:

```
    iocircuit model control ?-die <die_name>? -cell <cell_name>
    [-driver|-receiver] -allmodel [true|false]
    -simType [hspice|eldo|generic|spectre|lynxspice]
    -simPath <path> ?-simOption <path/filename>? ?-temperature <num>degC?
    ?-freq <num>hz? ?-vl <num>v? ?-vh <num>v? ?-tr <num>ns? ?-tf <num>ns?
    ?-optimtype <0|1|2>? ?-acculevel <num>? ?-usescale [true|false]?
    ?-scaleval <num>? -ibisoptimtype 0
```

where

`-vl`: Specifies the input LOW voltage.

`-vh`: Specifies the input HIGH voltage.

`-tr`: Specifies the rise time.

`-tf`: Specifies the fall time.

`-optimtype [0|1|2]`: Selects optimization for convergence or accuracy:

0 default

1 convergence

2 accuracy

`-usescale [true|false]`: Selects hierarchy scaling settings.

`-scaleval <num>`: Specifies the number by which to increase the scaling.

Example:

```
    iocircuit model control -cell dqbuff -model Model1 -driver
    -simtype GenericSpice -simpath /rhcsm/rd/linux64/bin/nspice
    -temperature 27degC -freq 100MHz -vl 0V -vh 3.3V -tr 0.1ns -tf 0.1ns
    -optimtype 1 -usescale off -scaleval 1
```

---

iocircuit model cellpin

Description: Configures the pins of the I/O cell model.

Syntax1:

```
iocircuit model cellpin ?-die <die_name>? -cell <cell_name>
[-driver|-receiver] -model <model_name> -input {<pin1>,<pin2>...}
-output {<pin3>,<pin4>...} -ioP {<pin_name>,<voltage_value>}
-ioG {<pin_name>, <voltage_value>}
-coreP {<pin_name1>,<voltage_value>,<pin_name2>,<voltage_value>...}
-coreG {<pin_name>,<voltage_value>} -enable <pin>
```

Syntax2 for global:

```
iocircuit model cellpin ?-die <die_name>? -cell <cell_name>
[-driver|-receiver] -model <model_name> -input {<pin1>,<pin2>...}
-output {<pin3>,<pin4>...} -ioP {<pin_name>,<voltage_value>}
-ioG {<pin_name>, <voltage_value>}
-coreP {<pin_name1>,<voltage_value>,<pin_name2>,<voltage_value>...}
-coreG {<pin_name>,<voltage_value>} -enable <pin> -allmode [true|false]
```

Example:

```
iocircuit model cellpin -cell dqbuff -model Model1 -driver -input input
-output output -ioP {vpp,3.3v} -ioG {vgg,0v} -coreP {vdd,3.3v} -coreG
{vss,0v}
```

---

iocircuit model validate

Description: Before I/O model generation, validates the setup conditions. After I/O model generation, validates the model. Run this check before and after model generation.

Syntax for setup validation:

```
iocircuit model validate ?-die <die_name>? -cell <cell_name> -model
<model_name> -setup ?-ibis?
```

Example:

```
iocircuit model validate -cell dqbuff -model Model1 -setup
```

Syntax for model validation:

```
iocircuit model validate -die <die_name> -cell <cell_name>
-model <model_name> ?-ibis?
```

Example:

```
iocircuit model validate -cell dqbuff -model Model1
iocircuit model validate -cell dqbuff -model Model1 -setup -ibis
```

---

iocircuit model generate

Description: If setup validation passes, generates the I/O model.

Syntax:

```
iocircuit model generate ?-die <die_name>? -cell <cell_name> -model
```

```
<model_name> ?-ibis? ?-all?
where
    -all: generate both setup validation and model validtion
```

Example:

```
    iocircuit model generate -cell dqbuff -model Model1
    iocircuit model generate -cell dqbuff -model Model1 -all
    iocircuit model generate -cell dqbuff -model Model1 -ibis -all
```

---

iocircuit model generateac

Description: Generates the AC model.

Syntax:

```
    iocircuit model generateac ?-die <die>? -cell <cell_name> -model
<model_name>
```

Example:

```
    iocircuit model generateac -cell dqbuff -model Model1
```

Analysis

---

analysis setup new

analysis setup importphotonic

Description: Create an analysis setup, import photonic configurations file.

Syntax:

analysis setup new -type [PI | SI | 3DICSetup | IBISAMI] ?-name <new setup name>?

analysis setup importphotonic -name <model name> -file <config file>

Examples:

analysis setup new -type PI -name PI1

analysis setup importphotonic -name TI1 -file /home/…/config.txt

---

analysis material add

analysis material setup

Description: Add new material

Syntax:

analysis material add -name uniquename -type [conductor|dielectric] ?<-tref value>? ?<-trran value>? ?<-ts value>? ?< -k value>? ?<-density -value>? ?<-cp value>? ?<-e value>? ?<-cte value>? ?<-pr value>? ?<-nu value>? ?<-g value>?

analysis material add -name name -from [system|project] -to project

analysis material setup -name uniquename -type [conductor|dielectric] ?<-tref value>? ?<-trran value>? ?<-ts value>? ?< -k value>? ?<-density -value>? ?<-cp value>? ?<-e value>? ?<-cte value>? ?<-pr value>? ?<-nu value>? ?<-g value>?

Examples:

analysis material add -name BGA_solder -type conductor -tref 25 -ts 25 -k 60 -density 8500e-3 -cp 200e-3

analysis material add -name BGA_underfill -type dielectric -tref 25 -ts 25 -k 0.5 -density 1900e-3 -cp 800e-3

analysis object setproperty -name STATIC_CTM -object INFO_BUMP -material BGA_solder -scope Project

analysis object setproperty -name STATIC_CTM -object INFO_BUMP -underfillmat BGA_underfill -underfillscope Project

Examples:

analysis object add -name PI1 -type Chip -object Chip

---

analysis object add

Description: Add new objects to an analysis setup.

Syntax:

analysis object add -name <setup name> -type [Chip | PKG | Board | VRM | Interposer] -object <object name>

Examples:

analysis object add -name PI1 -type Chip -object Chip

---

analysis object setproperty

Description: Set the base object for thermal flow, set power position for the chip model.

Syntax:

analysis object setproperty -name <model_name> -object <object name> -setasbase

analysis object setproperty -name <model name> -object <die object> -shrink <value>

Examples:

analysis object setproperty -name TI1 -object PCB -setasbase

analysis object setproperty -name TI1 -object Die1 -shrink 0.9

Description: Assign thermal material

Syntax:

analysis object setproperty -name <model name> -object <object name> -material -scope <System|Project>

analysis object setproperty -name <model name> -object <object name> -underfillmat -underfillscope <System|Project>

Example:

analysis object setproperty -name TI1 -object Bump_to_logic -material COPPER -scope System
analysis object setproperty -name TI1 -object Bump_to_logic -underfillmat FR-4 -underfillscope Project

Syntax:

analysis object setproperty –name <model name> -object <object name> -autopowerposition <value>

Example:

###disable Auto Power Position####
analysis object setproperty -name TI -object DIE1 -autopowerposition 0
########Set the power position####
analysis object setproperty -name TI -object DIE1 -powerposition 0.242

---

analysis object delete

Description: Delete objects in analysis setup.

Syntax:

analysis object delete -name <setup name> -object <object name>

Examples:

analysis object delete -name PI1 -object { Rx_PKG, Chip, Mem, Interposer, PKG }

---

analysis object export

Description: Get the component dimensional parameters.

Syntax:

analysis object export -name <model name> -object <die name list>/all -file <file name>

Examples:

analysis object export -name TI1 -object all -file /home/dump/geometry.txt
analysis object export -name TI1 -object Die1 -file /home/dump/geometry.txt
analysis object export -name TI1 -object Die1 Die2 -file /home/dump/geometry.txt

---

analysis connection add

analysis connection delete

Description: add or delete Bus line for 2 objects in analysis setup.

Syntax:

analysis connection [add | delete] -name <setup name> -object {object1, object2}
analysis connection add -name <model_name> -baseobject <object_name> -baseinterface <interface_name> -refobject <interface_name> -refinterface <interface_name.
analysis connection add -name <analysis name> -DieAutoConnection

Examples:

analysis connection add -name PI1 -object { Soc, Driver_PKG }
analysis connection delete -name PI1 -object { Soc, Driver_PKG }
analysis connection add -name TI1 -baseobject PKG -baseinterface Bottom_Interface -refobject

BGA -refinterface TOP

analysis connection add -name TI1_Static -DieAutoConnection

---

analysis object setmodel

Description: Assign cpm/channel model for each object in analysis setup.

Syntax:

analysis object setmodel -name <setup name> -object <object name> -model <model name in project tree>

Examples:

analysis object setmodel -name PI1 -object Soc -model import_cpm1

---

analysis connection component

Description: Select component for each object to be connected.

Syntax:

analysis connection component -interface { object1, object2 } -selcomp1 <object1 component> -selcomp2 < object2 component > -name <setup name>

Examples:

analysis connection component -interface { Driver_PKG, PCB } -selcomp1 CSP_BGA_BGA -selcomp2 SMD_VT3275_BGA865_U16 -name PI1

---

analysis connection footprint

Description: Do footprint connection for 2 objects.

Syntax:

analysis connection footprint ?-auto ?- interface {obj1,obj2} ?-scale <value>? ?-tolerance <value_in_mm>? ?-byname? -name <setup name> -mapfile ./pin_mapped.txt | 100

analysis connection footprint ?-manual? -interface {obj1,obj2} ?-scale <value>? -shiftx <value> -shifty <value> -rot <value> -flip [yes| no] -tolerance <value_in_mm> -name <setup name>

Examples:

analysis connection footprint -auto -interface { Soc, Driver_PKG } -name PI1

---

analysis connection addmap

Description: Connect 2 objects by node mapping.

Syntax:

analysis connection addmap -interface {object1, object1} -netpairs { net or node list } -name <setup name> -file <node mapping file>

Examples:

analysis connection addmap -interface { Driver_PKG, PCB } -netpairs { BGA_VDD_15_Group_SINK_, U16_VCORE_Group, BGA_VSS_Group_SINK_, U16_GND_Group } -

name PI1

---

analysis simulation control

Description: Edit the simulation parameter.

Syntax:

analysis simulation control –name <setup_name> ?-simoption <file_path>? ?-simcmdopt <option>? ?-temperature <value>? ?–simtime <value>? ?-tstep <value>? ?-core_num <value>? ?-ac <true/false>? ?-transient <true/false>? -simtype [Hspice | Eldo | GenericSpice | Spectre | LynxSpice | Nexxim | Nexximinternal] -simpath <simulator binary path> -advtimestepunit [fs | ps | ns | us | ms | s] -advtimestepdata { 0, stop_time1, time_step1, start_time, stop time2, time_step2,… } -freqsweepunit [Hz | KHz | MHz | GHz] -freqsweepdata { start_freq1, end_freq1, type(Log:1, Linear 0), sampling_point_num, …} -addpair true -pairobj0 <object1> -pairobj1 <object2> -paircomp0 <object1's component> -paircomp1 <object2's component> -pairnet0 <comp1's net> -pairnet1 <comp2's net> -pairnode0 <net1's node> -pairnode1 <net2's node> -removepair true -object <object to add probe> -probing <node name> ?-tf -file <TSMC Thermal tech file> -die <name> -tfdie <name>?

Examples:

analysis simulation control -simtype Hspice -simpath /appls/synopsys/K-2015.06-SP2/hspice/bin/hspice64 -name PI1

analysis simulation control -addpair true -pairobj0 Soc -pairobj1 Soc -paircomp0 CPM_IF_DEFAULT -paircomp1 CPM_IF_DEFAULT -pairnet0 VDD_15 -pairnet1 VSS -name PI1

analysis simulation control -object PCB -probing U11_GND_Group_SINK_ -name PI1

Syntax:

analysis simulation control -name <model_name> ? -mlptype<Multiphysics type>? ? -simtype <simulation type>? ?-boundary? ?-boundarytype <boundary_type>? ?-temperature <value C>? ?-airspeed <value>? ?-fixedbc? ?-object <object_name>? ?-layer <layer_name>? ?-topbc <value> ? ?-botbc <value> ? ?-enable <PKGTOP, PCBTOP, PKGSIDE, PCBBOTTOM, PCBSIDE> ? ?-pkgtop <value>? ?-pkgside <value>? ?-pcbtop <value>? ?<pcbbottom <valude>? ?pcbside <value>? ? -icepakregion <wall type>? ? -icpeakwall <status>? ? -icepakairspeed <value>? ? icepakpadding <aire padding percentage>? ?- solvertype <solver type>? ? -core_num <value>? ?- powerratio <value>? ?-maxiteration <value>? ? tempdiff <temperature diff>? ? -mesh_tolerance <value>? ?-mesh_maxLength <value>? ?-mesh_triangleCtrl <value>? ?-mesh_minAngle <value>? ?- dielectricfill <value>? ?- dumpneu <0|1>? ?- smartmesh <0|1>? ? imprintpowerbox <0|1>? ?- bumpmodel < line|solid >? ?- ballmodel <line|solid>? ?- viamodel < line|solid>? ?- tsvmodel< line|solid>?

Syntax:

analysis simulation control -name <model_name> -finegrid -clear (clear all)

analysis simulation control -name <model_name> -finegrid -remove <die_object_name>

analysis simulation control -name <model_name> -finegrid -add { die_object_name, llx(mm), lly(mm), urx(mm), ury(mm), pitch(um) }

Examples

analysis simulation control -name TI1-mlptype ThermalStress

analysis simulation control -name TI1 -mlptype Thermal

analysis simulation control -name TI1 -simtype Static

analysis simulation control -name TI1 -simtype Transient

analysis simulation control -name TI1_icepak -boundary -boundarytype ICEPAK

analysis simulation control -name TI1 -boundary -fixedbc -object interposer_bump -layer interposer_bump -topbc 25C -botbc 0.5H

analysis simulation control -name TI1 -boundary -boundarytype USER -enable { PKGTOP, PCBTOP, PKGSIDE, PCBBOTTOM, PCBSIDE } -pkgtop 20 -pkgside 20 -pcbtop 20 -pcbbottom 20 -pcbside 20

analysis simulation control -name TI1 -imprintpowerbox 1

analysis simulation control -name TI1 -finegrid -add { interposer_die_inst_Obj, 18.367, 5.52479, 25.0876, 10.7807, 672.064 }

Syntax:

analysis simulation control -name <model_name> -tf -file <file_name>

analysis simulation control -name <model_name> -tf -die <name> -tfdie <name>

analysis simulation control -name <model_name> -tf -die <name> bump <C4 BUMP|U_BUMP>

analysis simulation control -name <model_name> -tf -TIM? -object <name> -component <name>

analysis simulation control -name <model_name> -tf -MOLDING ?-object <name> -component <name>?

analysis simulation control -name <model_name> -tf -object <name> -component <name> -material <MOLDING_COMPOUND>

Examples:

analysis simulation control -name TI1 -tf -file ${DESIGN_DATA}/TF.xml.enc

analysis simulation control -name TI1 -tf -tim 1 -object -component

analysis simulation control -name TI1 -tf -molding 1 -object Molding -component MOLDING

analysis simulation control -name TI1 -tf -die INTERPOSER -tfdie TIS -bump C4_BUMP

analysis simulation control -name TI1 -tf -die SOC -tfdie SoC_die -bump U_BUMP

Syntax:

analysis simulation control -name <model_name> -transient -add -state <state_name>

analysis simulation control -name <model_name> -transient -state <state_name> -die <die_name> ? -ctm <power source model>? ?

analysis simulation control -name <model_name> -transient -add - activity < activity_name>

analysis simulation control -name <model_name> -transient - activity < activity_name> -duration <value> ?-cycle <value?

Examples:

analysis simulation control -name TI2 -transient -add -state State_1

analysis simulation control -name TI2 -transient -state State_1 -die interposer -ctm interposer_power

analysis simulation control -name TI2 -transient -state State_1 -die Die1 -ctm Die_ConstantPower

analysis simulation control -name TI2 -transient -add -activity Activity_1 -duration 10

analysis simulation control -name TI2 -transient -cycle 2

Syntax:

analysis simulation control -name <model_name> -gsrfile <gsr file>

analysis simulation control -name <model name> -gsr -key MetalMetalSupport -value 1

-**name**: specifies the analysis thermal model name.

-**gsrfile**: specifies thermal gsr file

Examples:

analysis simulation control -name TI1 -gsrfile ./appended.gsr

analysis simulation control -name TI1 -gsr -key MetalMetalSupport -value 1

analysis simulation control -name TI1 -gsr -key MetalDensityPrecision -value 1000

keyword purpose: MetalDensityPrecision is set to 100 by default, which means that ET can handle metal densities with values of 0%, 1% and 2%. If a tile has a metal density of 1.4%, ET will treat it as 1%, if the metal density is 1.6%, ET treats it as 2%

analysis simulation control -name TI1 -gsr -key ConsiderInjectionAsEqualLayer -value 1

###**keyword purpose:** Some designs include TSV/VIA structures passing through the layers. This keyword detects whether TSV/VIA passes through these layers/gaps. If ConsiderInjectionAsEqualLayer is 0, no TSV/VIA pass through the layers/gaps will be detected. If 1, TSV/VIA passing through the layers/gaps will be detected. ###

analysis simulation control -name TI1 -gsr -key DefaultK -value 0.3

###**keyword purpose:** When ET is doing a detailed simulation using CTM and cannot find any materials assigned to the CTM layer, ET will use the default material with K equal to 150 W/m-C, which is the property of silicon, i.e., bulk material type for a die. This keyword allows users to specify a default material k value if users know the K for those CTM layers. ###

Syntax: TCL for dumping image with a defined region and probe in 3D Postprocessing.

analysis simulation control -name <model> -image postpro -region <llx lly llz urx ury urz| clear> ?-showValue? ?-color <r g b>?

analysis simulation control -name <model> -image postpro -probe <name x y z | clear> ?-showValue? ?-color <r g b>?

analysis simulation control -name <model> -image postpro -start -szie <width hight> -bgcolor <color)> -textcolor <color> -precision <num>

analysis simulation control -name <model> -image postpro -contour <Temperature | Heatflux | XHeatflux | YHeatflux | ZHeatflux | Displacement | XDisplacement | YDisplacement | ZDisplacement | Current | XCurrent | YCurrent | ZCurrent | Voltage | Stress | ModelOnly >

analysis simulation control -name <model> -image postpro -node_report <report file> -contourlist Displacement Stress -component <component name list>

analysis simulation control -name <model> -image postpro -range <auto|max min>

analysis simulation control -name <model> -image postpro -show <all|objects name>

analysis simulation control -name <model> -image postpro -o <file> -view <top | bottom | front | back | left | right | SE> ?-showname? ?-showmaxmin <Color>?

analysis simulation control -name <model> -image postpro -maxmin_report <report file>

analysis simulation control -name <model> -image postpro -end

analysis simulation control -name <name> -image postpro -section_report <output_file_name> -contour Displacement -component <component name or all> -xplane <value> -oppositev

Examples:

analysis simulation control -name TI2 -image postpro -start -size 700 700 -bgcolor 0 255 255 -textcolor 0 0 0 -precision 3

analysis simulation control -name TI2 -image postpro -SectionView -Zplane 1.1

analysis simulation control -name TI2 -image postpro -show Die

analysis simulation control -name TI2 -image postpro -region 17.168 -7.219 1.1 23.168 -1.219 1.1 -

color 0 0 0

analysis simulation control -name TI2 -image postpro -probe N2 19.168 -6.219 1.1 -showValue

analysis simulation control -name TI2 -image postpro -view top -o /home/user/Neu7.png

analysis simulation control -name TI2_hole -image postpro -start -size 600 700 -bgcolor 0 255 255 -textcolor 0 0 0 -precision 6

analysis simulation control -name TI2_hole -image postpro -contourlist temperature

analysis simulation control -name TI2_hole -image postpro -range 21 21.5

analysis simulation control -name TI2_hole -image postpro -show PKG_Top -view se -o /my_folder/Neu21.png -showname

analysis simulation control -name TI2_hole -image postpro -contour XHeatFlux

analysis simulation control -name TI2_hole -image postpro -show all -view top -o my_folder /Neu31.png -showMaxMin 0 0 255

analysis simulation control -name TI2_hole -image postpro -maxmin_report / my_folder /maxmin.txt

analysis simulation control -name TI2_hole -image postpro -end

analysis simulation control -name TI1_stress -image postpro -section_report ./section_dis.txt -contour Displacement -component all -xplane -50 -oppositev

Description: TCL for exporting region report file and node information report in 3D Postprocessing.

Syntax:

analysis simulation control -name <model> -image postpro -region_report <file>

analysis simulation control -name <model> -image postpro -region_report <file> -selected component1 component2

analysis simulation control -name <model> -image postpro -start

analysis simulation control -name <model> -image postpro -node_report <report file> -contourlist Displacement/Stress -component <component name list>

Examples

analysis simulation control -name TI2_hole -image postpro -start -size 700 700 -bgcolor 0 255 255 -textcolor 0 0 0 -precision 3

analysis simulation control -name TI2_hole -image postpro -SectionView -Zplane 1.1

analysis simulation control -name TI2_hole -image postpro -show Die

analysis simulation control -name TI2_hole -image postpro -region 17.168 -7.219 1.1 23.168 -1.219 1.1 -color 0 0 0

analysis simulation control -name TI2_hole -image postpro -probe N2 19.168 -6.219 1.1

analysis simulation control -name TI2_hole -image postpro -region_report /my_folder /123.txt –selected

analysis simulation control -name TI1_ThermalStress -image postpro –start

analysis simulation control -name TI1_ThermalStress -image postpro -node_report / my_folder /Displacement.txt -contourlist Displacement -component Die4

Description: Dump cross-section view image in 3D-Postprocessing

Syntax

analysis simulation control -name <model> -image postpro -SectionView -undo/-redo/-restore

analysis simulation control -name <model> -image postpro -SectionView -Xplane <Xcoordinate> (-

oppositeV)

analysis simulation control -name <model> -image postpro -SectionView -Yplane <Ycoordinate> (-oppositeV)analysis simulation control -name <model> -image postpro -SectionView -ABC <A B C> (-oppositeV)

analysis simulation control -name <model> -image postpro -SectionView -TwoPoints <point1X point1Y point2X point2Y> (-oppositeV)

analysis simulation control -name <model> -image postpro -SectionView -PointNormal <pointX pointY normalX normalY> (-oppositeV)

analysis simulation control -name <model> -image postpro -SectionView -Zplane <Zcoordinate> (-oppositeH)

Examples:

analysis simulation control -name TI2_hole -image postpro -SectionView -restoreanalysis simulation control -name TI2_hole -image postpro -SectionView -PointNormal 1.1 2.2 3.3 4.4analysis simulation control -name TI2_hole -image postpro -SectionView -ABC 1.1 2.2 3.3 -oppositeV

analysis simulation control -name TI2_hole -image postpro -SectionView -Zplane 0.5

analysis simulation control -name TI2_hole -image postpro -SectionView -Zplane 0.5 -oppositeH

analysis simulation control -name TI2_hole -image postpro -SectionView -Xplane 3.3 -Zplane 0.5

analysis simulation control -name TI2_hole -image postpro -SectionView -TwoPoints 1.1 2.2 3.3 4.4 -Zplane 0.5 -oppositeH

analysis simulation control -name TI2_hole -image postpro -SectionView -Yplane 2.2 -oppositeV -Zplane 0.5 -oppositeH

Syntax: dump assembly model image

analysis simulation control -name <model> -image assembly -start -szie <width hight> -bgcolor <color(r g b (0~255))>

analysis simulation control -name <model> -image assembly -show <all|objects name>

analysis simulation control -name <model> -image assembly -o <file> -view <top|bottom|front|back|left|right|SE> ?-showname?

analysis simulation control -name <model> -image assembly –end

Examples:

analysis simulation control -name TI2_hole -image assembly -start -size 600 600 -bgcolor 255 255 255

analysis simulation control -name TI2_hole -image assembly -show PKG

analysis simulation control -name TI2_hole -image assembly -view Top -showname -o / my_folder /PKG1.png

analysis simulation control -name TI2_hole -image assembly -show all

analysis simulation control -name TI2_hole -image assembly -view front -o /my_folder/all.png

analysis simulation control -name TI2_hole -image assembly -end

Syntax: Probe temperature at a particular point in ng mode.

analysis simulation control -name <AnalysisName> -image postpro -ng -probe <path>/probe.txt -outputfile <path>/Tprobe.txt -contour Temperature

Examples:

analysis simulation control -name TI1 -image postpro -ng -probe /home/user/probe.txt -outputfile

/home/user/Tprobe.txt -contour Temperature

Input file – probe.txt

```
-2000 2000 1572
93.015 -120.19 1879.09
```

Output file

```
#X Y Z Value
-2000.000000 2000.000000 1572.000000 78.489998
693.015000 -120.190000 1879.090000 133.799952
```

Syntax: specify the HPC setting.

TCL command for SSH options.

analysis simulation control -name <analysis model> -hpcsetting -hpcenable 1 -hpcconfigstatus 1 -hpcsupporttype <1| 2> ? -hpccurhostname <host name>?

analysis simulation control -name <analysis model> -hpcsetting -hpcserversinsheet -clear

analysis simulation control -name <analysis model> -hpcsetting -hpcserversinsheet -add {machineA, processors }

analysis simulation control -name <analysis model> -hpcsetting -hpcservers -clear

analysis simulation control -name <analysis model> -hpcsetting -hpcservers -add {machineA, processors }

TCL command for slurm options

analysis simulation control -name <analysis model> -hpcsetting -hpcenable 1 -hpcsupporttype <type value> -hpcmodetype 1 -hpcschedulerconfigstatus 1 -hpcschedulercluster <partition name > -hpcschedulertasks <value> -hpcschedulernodes <value> -hpcschedulermemory <value>

- hpcsupporttype: specify the HPC mode, there are 5 types, 0|1|2|3|4.

0: smp mode

1: local server dmp mode

2: SSH dmp

3: Slurm

4: LSF

- hpcschedulercluster: specify Slurm partition.

Example 1, enable multiple machines

analysis simulation control -name TI3 -hpcsetting -hpcenable 1 -hpcconfigstatus 1 -hpcsupporttype 2

analysis simulation control -name TI3 -hpcsetting -hpcserversinsheet -clear

analysis simulation control -name TI3 -hpcsetting -hpcserversinsheet -add { sjocpsqa2, 32 }

analysis simulation control -name TI3 -hpcsetting -hpcserversinsheet -add { sjocpsqa1, 32 }

analysis simulation control -name TI3 -hpcsetting -hpcservers -clear

analysis simulation control -name TI3 -hpcsetting -hpcservers -add { sjocpsqa1, 32 }

analysis simulation control -name TI3 -hpcsetting -hpcservers -add { sjocpsqa2, 32 }

Example 2, enable one machine

analysis simulation control -name TI1 -hpcsetting -hpcenable 1 -hpcconfigstatus 1 -hpcsupporttype 1

analysis simulation control -name TI1 -hpcsetting -hpcserversinsheet -clear

analysis simulation control -name TI1 -hpcsetting -hpcserversinsheet -add { sjocpsqa2, 32 }

analysis simulation control -name TI1 -hpcsetting -hpcservers -clear

analysis simulation control -name TI1 -hpcsetting -hpcservers -add { sjocpsqa2, 32 }

Example 3, enable one machine but do not know the machine name, it applies to submitting a job with Slurm/LSF command.

analysis simulation control -name TI1 -hpcsetting -hpcenable 1 -hpcconfigstatus 1 -hpcsupporttype 1 -hpccurhostname $HOST

analysis simulation control -name Tran_Thermal -hpcsetting -hpcserversinsheet -clear

analysis simulation control -name TI1 -hpcsetting -hpcserversinsheet -add { $HOST, 32 }

analysis simulation control -name TI1 -hpcsetting -hpcservers -clear

analysis simulation control -name TI1 -hpcsetting -hpcservers -add { $HOST, 32 }

Example4, enable slurm setting.

analysis simulation control -name TI1 -hpcsetting -hpcenable 1 -hpcsupporttype 3 -hpcmodetype 1 -hpcschedulerconfigstatus 1 -hpcschedulertasks 16 -hpcschedulernodes 4 -hpcschedulermemory 100

Exmaple4, enable LSF setting.

analysis simulation control -name TI1 -hpcsetting -hpcsupporttype 4 -hpcmodetype 1 -hpcschedulerconfigstatus 1 -hpcschedulercluster testchip_1 -hpcenable 1 -hpcschedulememory 100 -hpcschedulernodes 1 -hpcscheduletasks 8

Syntax: specify native transient setting.

analysis simulation control -name <model_name> -transientflowtype [Native-FEM | State-based]

analysis simulation control -name <model_name> -nativetransient -timepower -clear

analysis simulation control -name <model_name> -nativetransient -stepcontrol -clear

analysis simulation control -name <model_name> -nativetransient -cycle <value>

analysis simulation control -name <model_name> -nativetransient -timepower -add -die <die name> -timepoints {value1, value2, …} -powerlist{value1, value2, …}

analysis simulation control -name <model_name> -nativetransient -stepcontrol -add -step <step name> -tstart <value> -tend <value>

analysis simulation control -name <model_name> -nativetransient -stepcontrol -step <step name> -type [0|1|2] ?-timeincrement <value>? ?-haslimit <0 | 1>? ?-MaxTimeStep <value>? ?-MinTimeStep <value>?

Examples:

analysis simulation control -name Tran -transientflowtype Native-FEM

analysis simulation control -name Tran -nativetransient -timepower -clear

analysis simulation control -name Tran -nativetransient -timepower -add -die DIE -timepoints { 0.5, 1, 1.5 } -powerlist { electrothermal_1, electrothermal_2, electrothermal_3 }

analysis simulation control -name Tran -nativetransient -stepcontrol -add -step Step_1 -tstart 0 -tend 0.5

analysis simulation control -name Tran -nativetransient -stepcontrol -add -step Step_2 -tstart 0.5 -tend 1

analysis simulation control -name Tran -nativetransient -stepcontrol -step Step_2 -type 2 -timeincrement 3 -haslimit 1 -MaxTimeStep 3 -MinTimeStep 0.05

analysis simulation control -name Tran -nativetransient -cycle 2

---

analysis simulation run

Description: Run transient/AC/fast simulation.

Syntax:

analysis simulation run -name <setup name> ?-fastsim?

Examples:

analysis simulation run -name TI1

---

analysis simulation report

Description: Export transient curve to a report file, export transient animation data to a txt file.

Syntax:

analysis simulation report -name <setup name> -transient ?-probe? ?-grid <axb>? ?-clear? ?-file <path>? ?-die <die_name>? ?-layer <layer name>?

analysis simulation report -name <setup name> -transient -probe -set|-add|-del -point

analysis simulation export -name <model name> -transient -animation -steptime <step value in s> -die <die_object> -file <file name>

analysis simulation export -name <model name> -die <die object> -static -thermalprofile -outputfolder <path>

analysis simulation export -name <model_name> -die <die_name> -static -temperature -outputfolder <path> -probe <user input probe file>

analysis simulation export -name <model_name> -die <die_name> -static -temperature -layer <layer name> -outputfolder <path>

Examples:

analysis simulation report -name TI1_TRAN -transient -probe -grid 1 1 -die DIE

analysis simulation export -name TI1_TRAN -transient -file ./test.dat

analysis simulation report -name TI1_TRAN -transient -probe clear

analysis simulation report -name TI1_TRAN -transient -probe -file ./test.txt -die SOC -layer M1

analysis simulation report -name TI1_TRAN -transient -probe -file ./test.txt

analysis simulation report -name TI1_TRAN -transient -probe -set -point { -589, -242, 650, -317}

analysis simulation export -name TI1 -transient -animation -steptime 10 -die Log_die -file /home/user/test.txt

analysis simulation export -name TI1_CTM -die INTERPOSER -static -thermalprofile -outputfolder /home/user/test.txt

analysis simulation export -name TI1_CTM -die INTERPOSER -static -temperature -outputfolder /home/bwu/test -probe /home/bwu/test/probe.txt

analysis simulation export -name DEMO_CTM -die DieT1 -static -temperature -layer BPMS -outputfolder /home/test -probe /home/user/test.txt

---

analysis simulation export

Description: Export image of thermal profile.

Syntax: analysis simulation export -name <model_name. -die <die name> -layer <layer name | all> -static -image ?-image_type <type>?-thermalProfile -outputfolder <path>

Example:

analysis simulation export -name Thermal_TOP_balance -die u_SoC -layer M1 -static -image -thermalProfile -outputfolder /home/qli/test/image/2.bmp

analysis simulation export -name Thermal_TOP_balance -die u_SoC -layer all -static -image -thermalProfile -outputfolder /home/qli/test/image

analysis simulation export -name Thermal_TOP_balance -die u_SoC -layer M1 -static -image -image_type jpg -thermalProfile -outputfolder home/qli/test/image/2.jpg

Description: Export thermal gradient file report for static analysis, it's based on thermal profile, the output files are ThermalMap* file and ThermalGradient* file.

Syntax 1: analysis simulation export -name <model_name> -die <die_name> -static -device -distance {list} -thermalprofile -outputfolder <path>

Syntax 2: analysis simulation export -name <model_name> -die <die_name> -static -device -distance {<distance_list>} -thermalprofile -analysis {< model_name _list>} -objects {object_name} -outputfolder <path>

Example1:

analysis simulation export -name TI1 -die DIE1 -static -device -distance {11, 15, 19, 24} -thermalprofile -outputfolder /home/qli/test

Example2: export thermal gradient and thermal map from multiple thermal analysis

analysis simulation export -name TI1 -die u_MEM -static -device -distance {10, 20, 30} -thermalprofile -analysis {TI2, TI3} -objects {u_MEM, u_MEM} -outputfolder /home/qli/test2

Debug

nxmsg

Description: Outputs the debug informations, used in TCL script.

Syntax:

nxmsg "message string"

---

nxlog

Description: Outputs the debug informations, used in TCL script.

Syntax:

nxlog "log string"
