cadence®

PSpice TCL Sample Scripts

Product Version 23.1
September 2023

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Contents

1

Overview

This document provides list of sample scripts that can be used to perform common task in PSpice and PSpice Advanced Analysis, such as performing processing on PSpice data file, accessing PSpice UI and simulating a design. Using these scripts, you will able to do the following tasks:

- Accessing PSpice using TCL Scripts
- Accessing PSpice Data Files using TCL Scripts
- Modifying Parameters in PSpice using TCL Scripts
- Performing PSpice Advanaced Analysis Simulations using TCL Scripts

2

Accessing PSpice using TCL Scripts

This chapter provides some sample scripts that will be helpful in accessing PSpice using TCL. In this chapter, we have used the following circuit file (.cir) as sample for all the sample scripts mentioned in the chapter.

rc. cir

```
**Simple Resistor circuit
R1 N1 0 RMOD 1000
R2 N1 N2 1000
V1 N2 0 5 ac=5

.model rmod res tc1=.01 tc2=.01

.tran 0 1
.probe
.end
**********
```

⚠ Remember the following points before you source the TCL script in PSpice command window:

- The present working directory(`pwd`) in the command window is same as the circuit file directory.
- The path changes in the script are aligned to your system.

Running Parametric Sweep Over Temperature For Transient Analysis

Source the following script in PSpice command window using the `source` command. Once the script is executed successfully, you will see an output file (.out) and data file (.dat) getting generated at the same location as the circuit file.

Parametric Sweep Over Temperature For Transient Analysis

```
#Sample TCL code to run parametric sweep over temperature for transient analysis:
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
source {D:\Cadence\SPB_17.2\tools\pspice\tclscripts\pspDB\pspice.tcl}
PSpiceSetupAnalysis rc.cir rc.out rc.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
PSpiceSetProbeTitle "R1=1000"
## Temperature is written in Probe header
PSpiceCommandDo DoTRAN true
PSpiceSetProbeTitle "R1=2000"
PSpiceParamSetValue R1.value 2000
PSpiceCommandDo DoTRAN true
PSpiceSetSimulationTemperature 50
PSpiceSetProbeTitle "R1=2000"
PSpiceCommandDo DoTRAN true
PSpiceSetSimulationTemperature 37
PSpiceParamSetValue R1.value 1000
PSpiceSetSimulationTemperature 50
PSpiceSetProbeTitle "R1=1000"
PSpiceCommandDo DoTRAN true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

Running AC Analysis

Source the following script in PSpice command window using the `source` command. Once the script is executed successfully, you will see an output file (.out) and data file (.dat) getting generated at the same location as the circuit file.

Running AC Analysis

```
#AC Analysis
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
source {D:\Cadence\SPB_17.2\tools\pspice\tclscripts\pspDB\pspice.tcl}
PSpiceSetupAnalysis rc.cir rc.out rc.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
PSpiceTranRun 1 false
PSpiceGetVoltage N1
PSpiceTranRun 1 true
PSpiceGetVoltage N1
PSpiceTranEnd
```

Runnung Transient Analysis in Time Steps

Source the following script in PSpice command window using the `source` command. Once the script is executed successfully, you will see an output file (.out) and data file (.dat) getting generated at the same location as the circuit file.

Traient Analysis Run in Time Steps

```
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis rc.cir rc.out rc.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
PSpiceTranRun 1 false
PSpiceGetVoltage N1
PSpiceTranRun 1 true
PSpiceGetVoltage N1
PSpiceTranEnd
```

Running Worstcase Analysis

Source the following script in PSpice command window using the `source` command. Once the script is executed successfully, you will see an output file (.out) and data file (.dat) getting generated at the same location as the circuit file.

Running Worstcase Analysis

```
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis rc.cir rc.out rc.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
## Params: Analysis
PSpiceWCSetup TRAN
#Set function to be evaluated e.g. max/min etc
PSpiceSetupMCFunction YMAX
# Set output variable to be evaulated
PSpiceSetupOutputVariable V N1
PSpiceCommandDo DoMC true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

Running MonteCarlo Analysis

Source the following script in PSpice command window using the `source` command. Once the script is executed successfully, you will see an output file (.out) and data file (.dat) getting generated at the same location as the circuit file.

Running MonteCarlo

```
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis rc.cir rc.out rc.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
## Params: RunCount Analysis
PSpiceMCSetup 10 TRAN
## Set number of output runs
PSpiceSetOutputRuns 10
#Set function to be evaluated e.g. max/min etc
PSpiceSetupMCFunction YMAX
# Call this to see param listing in out file
PSpiceSetupListParams
# Set output variable to be evaulated
PSpiceSetupOutputVariable V N1
## voltage between 2 nodes
PSpiceSetupOutputVariable V N1 N2
## Voltage across a 2-terminal device
PSpiceSetupOutputVariable V R1
#Current through a 2-terminal device
PSpiceSetupOutputVariable I R3
# doMC is a new option
PSpiceCommandDo doMC true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

3

Modifying Parameters in PSpice using TCL Scripts

In this chapter, we will use some sample circuit file and modify the parameters using the TCL script. Following topics will be covered in this chapter:

- Modifying Instance Parameters
- Modifying Instance Parameters in Hierarchy
- Modifying Global and Model Parameters

⚠ Remember the following points before you source the TCL script in PSpice command window:

- The present working directory(`pwd`) in the command window is same as the circuit file directory.
- The path changes in the script are aligned to your system.

Modifying Instance Parameters

Use the following circuit file (`rc.cir`) to modify instance parameters using the TCL script.

rc.cir

```
**********
R1 1 0 100 TC=1
R2 1 2 100
V1 2 0 5
.tran 0 .123
.end
```

To modify the instance parameter values, source the following TCL script in PSpice command window:

Modify Instance Paramters

```
#Sample TCL code to modify instance paramters
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis rc.cir rc.out rc.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
PSpiceParamSetValue R1.value 200
PSpiceCommandDo doTran true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

Modify Instance Paramters in Hierarchy

Use the following circuit file (`hier.cir`) to modify instance paramters in hierarchy:

hier.cir

```
**********
.subckt mysub 1 2
R1 1 0 100 TC=1
R2 1 2 100
.ends
X1 1 2 mysub
V1 2 0 5
.lib nom.lib
.options ACCT
.tran 0 .123
.op
.probe v(*)
.watch tran V([2])
.end
```

To modify the instance parameter values in hierarchy, source the following TCL script in PSpice command window:

Modify Instance Paramters in Hierarchy

```
#Sample TCL code to modify parameter values in a hierarchical design:
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis hier.cir hier.out hier.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
PSpiceParamSetValue X1.R1.value 200
PSpiceCommandDo doTran true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

Modifying Global and Model Parameters

Use the following circuit file (`models.cir`) to modify global and model paramters:

models.cir

```
**********
.param myval=100
.subckt mysub 1 2
R1 1 0 rmod {myval}
R2 1 2 rmod 100
.ends
.model rmod res r=2
.model rmod1 res r=2
X1 1 2 mysub
V1 2 0 5
.tran 0 .123
.watch tran V([2])
.end
```

To modify global and model paramters, source the following TCL script in PSpice command window:

Modifying Global and Model Paramters

```
#Sample TCL code to modify global and model parameter values:
load orPSP_ENG64.dll orpspeng
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis models.cir models.out models.dat "D:\Cadence\SPB_17.2\tools\pspice\library"
PSpiceParamSetValue X1.R1.value 2 true
PSpiceParamSetValue myval 200
PSpiceCommandDo doTran true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

4

Accessing PSpice Data Files using TCL Scripts

In this chapter , you will see some sample TCL scripts that will be used to access the .dat file. The output of the .dat file will be displayed on the PSpice command window.

Following are the sample TCL scripts that can be used to access PSpice .dat file:

- Direct Access API
- Evaluating an Expression
- Creating an AC Data File
- Reading a Multi-sectioned Data File

⚠ Remember the following points before you source the TCL script file in PSpice command window:

- The present working directory(`pwd`) in the command window is same as the circuit file directory.
- The path changes in the script are aligned to your system.

Accessing a PSpice Data File

Source the following TCL scripts in PSpice command window to access the PSpice data file:

Direct Access API

```
#Direct Access API
load [file normalize "D:/Cadence/SPB_17.2/tools/bin/orPIOData64.dll"]
# Create an object of the PIOData object
orPIOObject objFile
#load the data file for evualation
set filePath "D:/Cadence/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/transient/transient.dat"
objFile loadFile $filePath
#Get the number of block. For monte carlo runs 1 to 10 this will return 10
objFile getNumberofBlocks
#Set the active block number
objFile setBlock 0
#Get The header for the current block.this returns entire header.
puts [objFile getJSON]
#Get All the trace names for analog
objFile getDataNames 0
#Get The header for the current block.this returns entire header.
objFile getHeaderList
#To get only a single header item.This returns only one item
set lDataPoints [objFile getHeaderData "NumberofAnalogRows"]
#get the Data for "Time" in array TimeVector
set TimeVector [objFile get "Time"]
set I_C2 [objFile get "I(C2)"]
puts "$TimeVector $I_C2"
#access time data on shell
set Time0 $TimeVector(0)
#close the opened file
objFile closeFile
```

Evaluating an Expression

Similar to `orPIOObject`, PSpice Data File TCL Interface provides access to `orExprEval` objects, which can be used to evaluate an expression.

This is the TCL equivalent to using operators or functions in PSpice probe using voltage, current, or power data values.

For example:

max(V(N1)), abs(I(R1)), V(N1)-V(N2)

The `orExprEval` objects are created and initialized just like `orPIOObject` objects, and support most `orPIOObject` functions, for example, `loadFile`, `setBlock`, and `getDataNames`.

The primary difference from `orPIOObject` is that `orExprEval` also supports expression evaluation using evaluate function.

For example:

objFile evaluate Max(I(C2))

Source the following TCL script in PSpice command window to evaluate an expression in PSpice:

expression_eval.tcl

```
# Evaluate Expression
load [file normalize "D:/Cadence/SPB_17.2/tools/bin/orevalexpr64.dll"]
# Create an object of the PIOData object
orExprEval objFile
#load the data file for evualation
objFile loadFile "D:/Cadence/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/transient/transient.dat"
#Get the number of block.For monte carlo runs 1 to 10 this will return 10
puts [objFile getNumberofBlocks]
#Set the active block number
objFile setBlock 0
#Get The header for the current block.this returns entire header.
puts [objFile getJSON]
#Get All the trace names for analog
objFile getDataNames 0
#Get The header for the current block.this returns entire header.
objFile getHeaderList
#To get only a single header item.This returns only one item
set lDataPoints [objFile getHeaderData "NumberofAnalogRows"]
#get the Data for "Time" in array TimeVector
objFile get "Time" "TimeVector"
objFile get "I(C2)" "I_C2"
for {set i 0} {$i < $lDataPoints} {incr i} {
puts "Time - $TimeVector($i) I(C2)=$I_C2($i)"
}
puts "I(C2) MAX=[objFile evaluate Max(I(C2))],MIN=[objFile evaluate Avg(I(C2))],Average=[objFile
evaluate Min(I(C2))]"
#access time data on shell
set Time0 $TimeVector(0)
#close the opened file
objFile closeFile
```

Creating an AC Data File

Source the following TCL script to create an AC data file from PSpice command window:

create_ac_dat.tcl

```
#Create AC Dat File
load [file normalize "D:/Cadence/SPB_17.2/tools/bin/orevalexpr64.dll"]
orExprEval objFileWr
objFileWr create { AC.dat} 10
objFileWr addBlock
# Set the header of the block
objFileWr setHeaderData "CircuitName" "TCL Test circuit"
objFileWr setHeaderData "CircuitSubtitle" "TCL Sub circuit"
objFileWr setHeaderData "SimulationTime" "14:00::53"
objFileWr setHeaderData "Date" "01/01/2016"
objFileWr setHeaderData "RevisionNumber" "17.2"
objFileWr setHeaderData "Temperature" 17.0
objFileWr setHeaderData "ProgramID" 45645
#if the data to be stored is complex then set value as 2 otherwise 1
objFileWr setHeaderData "ComplexFlag" 1
#Datatype = 0 for Analog,1 for digital,2 for mixed Signal
objFileWr setHeaderData "DataType" 0
objFileWr setHeaderData "AnalysisName" "AC Analysis"
#if all cols are stored .else 0
objFileWr setHeaderData "DotProbeWithVariables" 1
#These api will set the number of columns(data names)
objFileWr setHeaderData "NumberofAnalogColumns" 2
objFileWr setHeaderData "SweepMode" 2
#starting and ending sweep value
objFileWr setHeaderData "StartingSweepValue" 10
objFileWr setHeaderData "EndingSweepValue" 1000000
objFileWr setHeaderData "ComplexFlag" 2
#To set sweep param
objFileWr setHeaderData "PrimaryParamName" "Frequency"
objFileWr addDataName "Frequency" 0
objFileWr addDataName "Gain" 0
set freq 0
for {set currBlock 0} {$currBlock < 100} {incr currBlock} {
incr freq
objFileWr addTimePoint $freq 0
#set complex values
objFileWr addDataPoint 2.0 1.0
}
objFileWr closeFile
```

Reading a Multi-sectioned PSpice Data file

Source the following TCL script to read a multi-sectioned data file from PSpice command window:

read_multisection_dat.tcl

```
# Reading Multi-sectioned Data File
load [file normalize "D:/Cadence/SPB_17.2/tools/bin/orevalexpr64.dll"]
set node1 A
set node2 B
# Create an object of the PIOData object
orExprEval objFile
objFile loadFile "D:/Cadence/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/transient/transient.dat"
set blockCount [objFile getNumberofBlocks]
puts $blockCount
for {set currBlock 0} {$currBlock < $blockCount} {incr currBlock} {
puts -nonewline "$currBlock of $blockCount\t"
objFile setBlock $currBlock
set lDataPoints [objFile getHeaderData "NumberofAnalogRows"]
#get the Data for "Time" in array TimeVector
objFile get "Time" "Time"
objFile get "V($node1)" "Va"
set Aavg [objFile evaluate Avg(V(a))]
puts -nonewline "AA - $Aavg\t"
set count 0
set Aavg 0
for {set i 0} {$i < $lDataPoints} {incr i} {
set Aavg [expr $Va($i) + $Aavg]
incr count
}
set Aavg [expr $Aavg / $count]
puts "Calculated - $Aavg"
}
#close the opened file
objFile closeFile
```

5

Performing PSpice Advanaced Analysis Simulations using TCL Scripts

In this chapter, you will use TCL scripts to perform the following tasks:

- Performing All PSpice Advanced Analsyis
- Performing Processing on a Design
- Performing Smoke Analysis

⚠ Remember the following points before you source the TCL script in PSpice command window:

- The present working directory(`pwd`) in the command window is same as the circuit file directory.
- The path changes in the script are aligned to your system.

Performing All the PSpice Adavnaced Analysis Simulation using TCL Script

Using the following TCL script, you can perform all the following analysis in one go: Sensitivity, Optimizer, Monte Carlo, Parameter Plot, and Smoke.

To use the script from PSpice Advanced Analysis command window, use the source command.

all_pspaa_analysis.tcl

```
#Perform all types of analysis from a single file
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/oradvctrlr64.dll"] PspAACtrlTcl
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/orPsp_Eng64.dll"] orPspEng
pspAASetLicenseBatchMode PSpiceAA
PSpiceSetLicenseBatchMode PSpiceAD
set pAAInterface [pspAAcreateInterface]
```

```
$pAAInterface openAAProfile
"D:/Cadence_lite/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/transient/bpf.aap"
set pSensitivityController [$pAAInterface getSensitivityController]
$pSensitivityController addSpecifications NULL "-1"
set pMeasurement [new_CSensSpecification]
$pMeasurement setState 1
$pMeasurement setProfileName "ac.sim"
$pMeasurement setName "Max(V(OUT))"
$pMeasurement setNotes "Okay"
$pMeasurement setSeverity 1
$pSensitivityController addSpecifications $pMeasurement
$pSensitivityController addSpecifications NULL "1"
$pSensitivityController start 1 1
set pMonteController [$pAAInterface getMonteController]
$pMonteController addSpecifications NULL "-1"
set pMeasurement [new_CMonteSpecification]
$pMeasurement setState 1
$pMeasurement setProfileName "tran.sim"
$pMeasurement setName "Max(V(OUT))"
$pMeasurement setNotes "Okay"
$pMeasurement setSeverity 1
$pMonteController addSpecifications $pMeasurement
$pMonteController addSpecifications NULL "1"
$pMonteController start 1 1
set pPplotController [$pAAInterface getPplotController]
$pPplotController addSpecifications NULL "-1"
$pPplotController addParameters NULL "-1"
set pMeasurement [new_CPplotSpecification]
$pMeasurement setState 1
$pMeasurement setProfileName "tran.sim"
$pMeasurement setName "Max(I(R3))"
$pMeasurement setNotes "Okay"
$pMeasurement setSeverity 1
$pPplotController addSpecifications $pMeasurement
$pPplotController addSpecifications NULL "1"
set pParam [new_CPplotParameter]
$pParam setState 1
$pParam setNotes "Okay"
$pParam setRefDes "R2"
$pParam setParamName "VALUE"
$pParam setSweepVariable "outer"
$pParam setSweepType $::PSP_SWEEP_LINEAR
$pParam setPoints "30,3000,300"
$pPplotController addParameters $pParam
$pPplotController addParameters NULL "1"
set pPlotSettings [new_CPplotSettings]
$pPlotSettings setSweepFilePath
"D:/Cadence_lite/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/transient/CurrentPlot.tbl"
$pPplotController addSettings $pPlotSettings
delete_CPplotSettings $pPlotSettings
set pPlotInfo [new_CPplotPlotInfo]
$pPlotInfo setValue "Plot 1" "R3::VALUE" "X-Axis"
$pPlotInfo setIsMeasurement "Plot 1" "R3::VALUE" 0
```

```
$pPlotInfo setBasePPlotPos "Plot 1" "R3::VALUE" 0
$pPlotInfo setBasePPlotConstPos "Plot 1" "R3::VALUE" -1
$pPlotInfo setValue "Plot 1" "tran.sim::max(i(R3))" "Y-Axis"
$pPlotInfo setIsMeasurement "Plot 1" "tran.sim::max(i(R3))" 1
$pPlotInfo setBasePPlotPos "Plot 1" "tran.sim::max(i(R3))" 0
$pPlotInfo setBasePPlotConstPos "Plot 1" "tran.sim::max(i(R3))" -1
$pPplotController addPPlotInfo $pPlotInfo
delete_CPplotPlotInfo $pPlotInfo
$pPplotController clearSweepValues
$pPplotController writePplotDataFile
"D:/Cadence_lite/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/bpf/CurrentPlot.tbl"
$pPplotController start 1 1
$pPplotController writePplotDataFile
"D:/Cadence_lite/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/transient/CurrentPlot.tbl"
set pOptController [$pAAInterface getOptimizerController]
$pAAInterface setCurrentEngine $::HYBRID
$pAAInterface setCurrentGear $::DEFAULT_GEAR
$pOptController addParameters NULL "-1"
set pOptParam [new_COptParameter]
$pOptParam setState 1
$pOptParam setLocked 0
$pOptParam setRefDes "R3"
$pOptParam setParamName "VALUE"
$pOptParam setMin “22k”
$pOptParam setMax “2200k”
$pOptParam setParamVal “220k”
$pOptParam setCurrent “220k”
$pOptParam setDiscreteTableAlias ""
$pOptParam setDiscreteTable ""
$pOptParam setNotes "Okay"
$pOptParam setSeverity 1
$pOptController addParameters $pOptParam
$pOptController addParameters NULL "1"
$pOptController addSpecifications NULL 0 "-1"
set pMeasurement [new_COptSpecification]
$pMeasurement setState 1
$pMeasurement setGraphState 1
$pMeasurement setIsCurveFit 0
$pMeasurement setIsGoal 1
$pMeasurement setProfileName "tran.sim"
$pMeasurement setName "Max(I(R3))"
$pMeasurement setSpecification 0
$pMeasurement setSpecification2 100
$pMeasurement setWeight 1
$pMeasurement setNominal ""
$pMeasurement setCurrent ""
$pMeasurement setError ""
$pMeasurement setNotes "Okay"
$pMeasurement setSeverity 1
$pOptController addSpecifications $pMeasurement 0
$pOptController addSpecifications NULL 0 "1"
$pOptController start 1 1
```

```
set pSmokeController [$pAAInterface getSmokeController]
$pSmokeController runSmoke
"D:/Cadence_lite/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/transient/transient.sim"
$pAAInterface saveAAProfile ""
pspAAdeleteInterface $pAAInterface
```

Performing Processing of a Design

Using the following TCL script, you can do processing on a PSpice Advanced Analysis file (.aap) to assign tolerance for instance parameters. Use the source command to source the script in PSpice Advanced Analysis command window.

process_design.tcl

```
#Perform processing of a design
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/orCommonTcl64.dll"] orCommonTcl
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/oradvctrlr64.dll"] PspAACtrlTcl
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/orpsppmpsvr64.dll"] PspAATcl
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/orPsp_Eng64.dll"] orPspEng
proc pspDesignTrue { args } {
return 1;
}
proc processDesignStart { } {
puts "In Process Design Start"
PSAAReader_openDesign
PSAAReader_enableAAInst_TclEval "R1"
}
proc processDesignEnd { } {
puts "In Process Design End"
}
proc processPostInst { } {
puts "In post Process Inst"
}
proc processInst { pInstName } {
puts "In Process Inst"
set instTypeC [string first C $pInstName]
if {$instTypeC==0} {
PSAAReader_setAAInstParam_Tol $mCurInstance "MY_CVAL" "10%" "10%"
}
set instTypeR [string first R $pInstName]
if {$instTypeR==0} {
set my_rval1 1
set my_rval2 1
PSAAReader_setPSpiceInstParam $pInstName "my_rval1" $my_rval1
PSAAReader_setPSpiceInstParam $pInstName "my_rval2" $my_rval2
set pos_tol [expr $my_rval1 + $my_rval2]
set neg_tol $pos_tol
PSAAReader_setAAInstParam_Tol $pInstName "MY_RVAL" $pos_tol% $neg_tol%
}
```

```
}
}
RegisterAction "_cdnPspAAProcessDesignStart" "::pspDesignTrue" "" "::processDesignStart" ""
RegisterAction "_cdnPspAAProcessInst" "::pspDesignTrue" "" "::processInst" ""
RegisterAction "_cdnPspAAProcessPostInst" "::pspDesignTrue" "" "::processPostInst" ""
RegisterAction "_cdnPspAAProcessDesignEnd" "::pspDesignTrue" "" "::processDesignEnd" ""
pspAASetLicenseBatchMode PSpiceAA
PSpiceSetLicenseBatchMode PSpiceAD
set pAAInterface [pspAAcreateInterface]
$pAAInterface openAAProfile
"D:/Cadence_lite/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/SCHEMATIC1.aap"
set pSensitivityController [$pAAInterface getSensitivityController]
$pSensitivityController addSpecifications NULL "-1"
set pMeasurement [new_CSensSpecification]
$pMeasurement setState 1
$pMeasurement setProfileName "ac.sim"
$pMeasurement setName "Max(V(OUT))"
$pMeasurement setNotes "Okay"
$pMeasurement setSeverity 1
$pSensitivityController addSpecifications $pMeasurement
$pSensitivityController addSpecifications NULL "1"
$pSensitivityController start 1 1
pspAAdeleteInterface $pAAInterface
```

Performing Smok Analysis using TCL Script

Using the following TCL Script, you can perform smoke analysis in PSpice Advanced Analysis. To source the TCL script file in PSpice Advanced Analysis, use the `source` command.

pspaa_smoke.tcl

```
#Perform Smoke Analysis
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/orCommonTcl64.dll"] orCommonTcl
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/oradvctrlr64.dll"] PspAACtrlTcl
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/orpsppmpsvr64.dll"] PspAATcl
load [file normalize "D:/Cadence_lite/SPB_17.2/tools/bin/orPsp_Eng64.dll"] orPspEng
pspAASetLicenseBatchMode PSpiceAA
PSpiceSetLicenseBatchMode PSpiceAD
proc pspTrue { args } {
return 1;
}
proc processDesignStart { } {
puts "Design Start"
PSAAReader_openDesign
PSAAReader_enableAAInst_TclEval "R1"
PSAAReader_enableAAInstDevSmoke "R1"
}
proc ::smokeDerateReliability { pInstName pParamName pDerFactor pDerFile } {
puts "Derate Reliability"
```

```
}
proc ::smokeDerateTemp { pInstName pParamName } {
puts "Derate Temp $pInstName $pParamName"
set lInst [$::pAASmokeInterface getInst "R1"]
set lrba [$lInst GetMaxOpValue "RBA"]
set lParam [$::pAASmokeInterface getInstSmokeParam "R1" "PDM"]
$lParam SetPeakDerMax 45
puts "Exit Derate Temp $pInstName $pParamName"
}
proc ::smokePostPDM&TJ { pInstName } {
}
proc ::smokePostPDM&TB { pInstName } {
puts "PostPDML&TB"
set lInst [$::pAASmokeInterface getInst "R1"]
set lrba [$lInst GetMaxOpValue "RBA"]
set lParam [$::pAASmokeInterface getInstSmokeParam "R1" "PDM"]
$lParam SetPeak 50
}
proc ::smokePostPDML { pInstName } {
puts "PostPDML"
}
proc ::runCustomTest { pInstName } {
puts "setting up custom test"
set param $::pAASmokeInterface addInstSmokeParam "R1" "MYSMKTEST"
####### Get the peak, average, rms values from the dat file #######
####### Get the max op value from the instance parameters #########
####### Perform user defined algorithms for derating values #######
####### Set the smoke output values back to smoke results of the tool ######
$param SetMaxOpValue 400
$param SetPeak 200
$param SetAverage 210
$param SetRMS 220
$param SetPeakDerMax 300
$param SetAverageDerMax 290
$param SetRMSDerMax 290
}
RegisterAction "_cdnPspAAProcessDesignStart" "::pspTrue" "" "::processDesignStart" ""
RegisterAction "_cdnPspAAAnalysisComplete" "::pspTrue" "" "::saveDesign" ""
RegisterAction "_cdnPspAASmokeDerateReliability" "::pspTrue" "" "::smokeDerateReliability" ""
RegisterAction "_cdnPspAASmokeDerateTemp" "::pspTrue" "" "::smokeDerateTemp" ""
RegisterAction "_cdnPspAASmokePostPDM&TJ" "::pspTrue" "" "::smokePostPDM&TJ" ""
RegisterAction "_cdnPspAASmokePostPDM&TB" "::pspTrue" "" "::smokePostPDM&TB" ""
RegisterAction "_cdnPspAASmokePostPDML" "::pspTrue" "" "::smokePostPDML" ""
RegisterAction "_cdnPspAARunCustomSmoke" "::pspTrue" "" "::runCustomTest" ""
# Run smoke
set pAAInterface [pspAAcreateInterface]
$::pAAInterface openAAProfile
"D:/Cadence_lite/SPB_17.2/tools/pspice/capture_samples/advanls/bpf/bpf-
PSpiceFiles/SCHEMATIC1/SCHEMATIC1.aap"
set ::pAASmokeInterface [$pAAInterface getSmokeController]
$::pAAInterface getAdvGeneralSettings
$::pAAInterface getSmokeSettings
$::pAASmokeInterface runSmoke
{D:\Cadence_lite\SPB_17.2\tools\pspice\capture_samples\advanls\bpf\bpf-
```

```
PSpiceFiles\SCHEMATIC1\transient\transient.sim}
$::pAAInterface saveAAProfile ""
pspAAdeleteInterface $::pAAInterface
```