cadence®

Voltus-Fi Custom Power Integrity Solution XL User Guide

Product Version ICADVM20.1
October 2020

© 2014-2020 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

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

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 1-800-862-4522.

All other trademarks are the property of their respective holders.

Restricted Print Permission: This publication is protected by copyright and any unauthorized use of this publication may violate copyright, trademark, and other laws. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. This statement grants you permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used solely for personal, informational, and noncommercial purposes;
2. The publication may not be modified in any way;
3. Any copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement; and
4. Cadence reserves the right to revoke this authorization at any time, and any such use shall be discontinued immediately upon written notice from Cadence.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

Licensing Requirements 21
Related Documentation 22
Voltus-Fi Custom Power Integrity Solution Documentation 22
What's New and KPNS 22
Other Documentation 22
Additional Learning Resources 22
Video Library 22
Virtuoso Videos Book 23
Voltus-Fi-XL Videos 23
Rapid Adoption Kits 23
Help and Support Facilities 24
Customer Support 24
Feedback about Documentation 24
Typographic and Syntax Conventions 25

1
Voltus-Fi Licensing Information 27
Voltus-Fi-XL Compatibility with MMSIM/SPECTRE and EXT Product Versions 29
Voltus-Fi-XL Compatibility with Voltus IC Power Integrity Solution 29

2
Getting Started with Voltus-Fi-XL 31
Key Features of Voltus-Fi-XL 32
Installation Information 34
Running Voltus-Fi-XL in Batch Mode 34
Running Voltus-Fi-XL in GUI Mode 35
Starting the Virtuoso Design Environment 35
Opening the Schematic or Layout View 36
Launching Voltus-Fi-XL 36
Specifying Simulation Settings in ADE L/ADE XL 39
Specifying EMIR Analysis Settings in ADE L/ADE XL 39

© 2020 All Rights Reserved.

Generating Simulation Database in ADE L/ADE XL 42
Accessing Documentation and Help 42
Voltus-Fi-XL Console Description 43

3
Product Overview 45
Key Product Features and Description 45

4
Data Preparation 51
Voltus-Fi-XL Flow Overview 52
Spectre APS/XPS EMIR Flow Overview 52
Data Requirements for Voltus-Fi-XL 55
xDSPF Requirements 55
Simulation Requirements and Setup 56
Outputs of EMIR Analysis 89

5
EM Rules Specification 91
EM Rule File Requirements in Voltus-Fi-XL 92
Parameters for Limit-Based EM Analysis in Voltus-Fi-XL 93
Supported Current Density Keywords in Voltus-Fi-XL 95
EM Model Syntax 96
Rules for Defining EM Parameters in Voltus-Fi-XL 100
Rules for Specifying Via Area 112
Rules for Specifying the EM Analysis Type for Power and Ground Nets 113
EM Rule Selection Priority 115
Process 120
EM Data File in Voltus-Fi-XL 122

6
xDSPF Generation using Quantus 125
Generating xDSPF File 126

© 2020 All Rights Reserved.

Specifying Input Data Information 126
Specifying Process Node-Specific Information 128
Specifying Technology Data Information 128
Specifying Output Information 129
Extract xDSPF Form Description 131
Customized Extract Commands File 138
Voltus-Fi Quantus Log File 139
Voltus-Fi Extract xDSPF GUI Last File 139

7
Static Current Analysis 141

Static Current Analysis 142
Static EMIR Flow in Spectre 142
Static Current Analysis Overview 142
Static Current Analysis Flow 144
Data Requirements for Static Current Calculation 144
Setting Up and Running Static Current Analysis 144
Static Current Analysis Report 153

8
IR Drop Analysis Results 155

Overview 158
Batch Mode Support for IR Drop Analysis 158
Signal Net IR Drop Analysis 158
Power Gate Support 159
Viewing the Power-Up Summary Report 161
Static EMIR Analysis 162
Static Power Grid Solver 163
Solid Shape Highlighting 165
Quantus Shape Database-Based Flow 165
Support for Specifying Cut-Layers for Metal Layers while Highlighting Solid Shapes 166
Displaying Finer Color Gradient for Selected Layers in IR and EM Plots 168
Specifying Multiple Simulation Result Files for EMIR Analysis 169
GUI Support for Specifying Multiple Bin Files in Different Directories 169
GUI Support for Specifying Multiple Bin Files in the Same Directory 170

© 2020 All Rights Reserved.

Batch Mode Support for Specifying Multiple Bin Files 171
Viewing the IR Drop Analysis Results 172
Displaying and Querying EMIR Results 184
Viewing EMIR Violations in Annotation Browser 185
Querying EMIR Results 189
Specifying the Display Options for IR/EM Plots 190
Types of IR Drop Analysis Plots 215
IR, IRAVG – IR Drop Plots 216
RC, RCAVG, and RCRMS – Resistor Current Plots 217
IV – Transistor Voltage Plot 219
TC, TCAVG, TCRMS – Tap Current Plots 220
POWER – Resistor Power Plots 222
PI – Powergate Current Plot 224
PV – Powergate Voltage Plot 227
REffective – Effective Resistance Plot 229
Plotting the Least-Resistive Path 230
Batch Mode Support for LRP Analysis 231
Batch Mode Support for LRP Analysis 231
GUI Support for LRP Analysis 232
Calculating Effective Resistance between any Two Nodes on a Net 247
Batch Mode Support for Calculating Effective Resistance 247
GUI Support for Calculating Effective Resistance 248
Generating EMIR Analysis Reports in Voltus-Fi-XL 252
Reports Generated Using the EMIR Control File 252
Format of IR Reports 253
Default Reports Generated for EMIR Analysis 254
RLRP Report 259
Reports Generated for Multiple Bin Files for Different Nets from the Same Testbench . 260
Reports Generated for Multiple Bin Files from Different Testbenches in a Design . 260
Reporting a List of Devices from a Subnode 266
Use Model 266
Sample Command File and Results File 267
Reporting On-Resistance (RON) Values 269
Overview 269
Generating the RON Report 269

© 2020 All Rights Reserved.

Output 269

9
EM Analysis Results 273

Overview 274
Types of EM Analysis 275
Average (avg) Analysis 275
Root Mean Square (rms) Analysis 275
Peak (max) Analysis 275
AC-Peak (AC Peak) Analysis 275
Avg-Abs (avgabs) 278
Design Resistor EM Current Analysis 279
Viewing the EM Analysis Results 279
Displaying Current Directions for EM Plots 289
Reporting the Direction of Currents in the EM Result File 290
Generating EM Reports in Voltus-Fi-XL 291
Types of EM Analysis Plots 292
RJ JMAX – Current Density/Limit and J_MAX – Current Density Plots 293
JAVG – Current Density/Limit and J_AVG Current Density Plots 294
JABSAVG – Current Density/Limit and J_ABSAVG – Current Density Plots 296
JACRMS – Current Density/Limit and J_ACRMS – Current Density Plots 298
JACPEAK – Current Density/Limit and J_ACPEAK – Current Density Plots 300

10
Self-Heating Effect Analysis 303

Overview 305
Self-Heating Effect Analysis 305
Thermal-Aware DC EM Analysis 306
Hierarchical SHE Analysis 306
Data Requirements 307
Compatible Software Versions 308
Flow Overview 308
Performing the Analysis 309
Running Pegasus or PVS LVS 310
Running Quantus Parasitic Extraction 310

© 2020 All Rights Reserved.

Running Post-Layout Simulation using Spectre APS 311
Viewing the Analysis Plots in Voltus-Fi-XL 312
Types of Plots 313
Thermal-DeltaT Regions 314
Thermal-DeltaT Tiles 315
RJ_SH - Current Density/Limit - max (SH) 316
JAVG_SH - Current Density/Limit - avg (SH) 316
JACRMS_SH - Current Density/Limit - rms (SH) 317
Re-Evaluation of EM Limit Based on dTemp 317
EMIR Control File Options 318
Batch Mode Support for Running SHE Analysis and Thermal-Aware DC EM Analysis 326
Command for Encrypting Data in param.sh File 329
Generating Reports 330

11
ESD Analysis 335

Overview 336
Specifying ESD Analysis Settings 337
Opening ESD Form 337
Specifying Mandatory Inputs 338
Specifying Optional Inputs 339
Loading DSPF File 339
Selecting Instances for Analysis 339
Specifying Advanced Settings 341
xDSPF File Optimization 344
Generating Optimized xDSPF File 344
ESD Effective Resistance Checks 346
Effective Resistance Checks Flow 346
Key Features of ESD Effective Resistance Checks 346
Prerequisites for Effective Resistance Checks 349
Running Effective Resistance Checks 350
ESD Current Density Checks 354
ESD Current Density Checks Flow 354
Key Features of ESD Current Density Checks 354
Types of Analysis for ESD Current Density Checks 357

© 2020 All Rights Reserved.

Prerequisites for ESD Current Density Checks . . . 360
Setting Quantus for xDSPF Generation . . . 362
ESD Modeling . . . 363
Specifying ESD Models . . . 365
ESD Testbench . . . 366
Specifying ESD Testbench . . . 367
Specifying In-House Testbench . . . 367
Specifying Tool-Generated Testbench . . . 368
ESD Zaps . . . 369
Creating Custom ESD Zaps . . . 370
Running Current Density Checks . . . 371
Running Current Density Checks for DC Mode . . . 373
Running Current Density Checks for CDM Mode . . . 374
Running Current Density Checks for HBM Mode . . . 375
Performing Simulation . . . 377
Loading the Simulation Results . . . 379
Viewing the ESD Current Reports . . . 379
Viewing Waveforms for Dynamic Simulation . . . 380
Viewing the Simulation Log Information . . . 381
Viewing the Device Ron Information . . . 382
Rule-Based ESD Checks . . . 383
ESD Rule File . . . 384
Running Rule-Based ESD Checks . . . 388
Batch Mode Support for ESD Analysis . . . 390
Batch Mode Support for Effective Resistance Checks . . . 390
Batch Mode Support for Current Density Checks . . . 390
Batch Mode Support for Rule-Based Checks . . . 391
Using Distributed Processing in Voltus-Fi-XL . . . 393
Outputs of ESD Analysis . . . 393
Effective Resistance Check Reports . . . 393
Current Density Check Reports . . . 398
Rule-Based Check Reports . . . 400
Summary Report . . . 406
Sample ICT File with EM Limits . . . 410
ESD Form Description . . . 411

© 2020 All Rights Reserved.

ESD Form - ESD Setup Tab 411
Advanced Settings Form - Run Settings Tab 413
Advanced Settings Form - Simulation Settings Tab 414
ESD Form - ESD Optimizer Tab 415
ESD Form - ESD Model Tab 415
ESD Form - ESD Run Tab 418
ESD Form - ESD Rule Check Tab 422

12
Structural Analysis 425
Structural Analysis in Voltus-Fi-XL 426
Checks Performed for Structural Analysis in Voltus-Fi-XL 426
Top Via Missing Check in Voltus-Fi-XL 427
Bottom Via Missing Check in Voltus-Fi-XL 429
Missing Via Connections Check in Voltus-Fi-XL 429
Via Coverage Ratio Check in Voltus-Fi-XL 431
Skewed Via Ratio Check in Voltus-Fi-XL 432
Running Structural Analysis in Voltus-Fi-XL 434
Results of Structural Analysis in Voltus-Fi-XL 436
Show Unconnected from Top 436
Show Unconnected from Bottom 437
Bottom / Top Via Ratio 438
Missing Via Connections 439
IR/EM Results Form — Structural Analysis Tab 441

13
What-If EMIR Analysis 445
Overview 446
ECO Commands Supported in Voltus-Fi-XL 447

© 2020 All Rights Reserved.

Performing What-If (ECO) Analysis in Voltus-Fi-XL 448
Pre-Filling the Run Simulation Form 456
Output 457
Options for Performing What-If (ECO) Analysis 458
Layout Modification Options 466
Adding Pins and Modifying the Layout 466
Deleting Pins and Modifying the Layout 468
Adding Resistors and Modifying the Layout 469
Deleting Resistors and Modifying the Layout 470
Adding Vias and Modifying the Layout 470
Deleting Vias and Modifying the Layout 471
Design Rule Checking 472

14
Power-Grid View Generation 473

What is Power-Grid View? 474
PGV Generation in Voltus-Fi-XL 474
Data Requirements and Flow for PGV Creation 474
Open Access (OA)-Based Library Generation 478
Automatic Generation of Layer Map Files 479
Automatic Generation of the LEF layer map File 479
Generating Power-Grid Views in Voltus-Fi-XL 481
Troubleshooting Information for PGV Generation in Voltus-Fi-XL 485

15
Hierarchical EMIR Analysis 487

Hierarchical EMIR Analysis in Voltus-Fi-XL 489
The High-Level Flow 490
Compatible Software Versions 492
Data Requirements for the Hierarchical EMIR Flow 492
Types of Block-Level xDSPF – Detailed, Port, and Via Snapped 493
PWL Current Files for Top-Level Simulation 494
Hierarchical EMIR Analysis Flow in GUI Mode 496
Opening the Hierarchical EMIR Forms 496
Creating the Block PGVs and xDSPF 496

© 2020 All Rights Reserved.

Generating the Top-Level xDSPF File . . . 500
Running the Top-Level Simulation . . . 508
The Hierarchical EMIR Analysis Batch Mode Flow . . . 512
set_mmsim_pgv_nets . . . 513
create_pgv . . . 514
extract_xdspf . . . 517
set_block_data . . . 519
run_top_simulation . . . 521
The Hierarchical EMIR Analysis Batch Mode Automation Flow . . . 524
Block PGV Generation . . . 524
Block PGV Inputs for Top-Level Extraction and Simulation . . . 524
The Top-Level Extraction and Simulation Run . . . 527
Multi-Mode Support . . . 529
A Sample Script for Automated Batch Mode Flow . . . 529
Analyzing the EMIR Results for Hierarchical Analysis . . . 532
Loading the IR/EM Results for the Hierarchical Flow in the GUI . . . 532
Viewing Plots for Top and Block-Level Results . . . 534
Analyzing the EMIR Results . . . 536
Sample File Formats for Hierarchical EMIR analysis . . . 538
Sample Block-Level xDSPF Files . . . 538
A Sample Top-Level xDSPF file . . . 541
A Macro LEF File . . . 543

16
Batch Mode Execution . . . 545

Batch Mode Support in Voltus-Fi-XL . . . 547
Command File Support . . . 547
Configuration File Support . . . 547
extract_xdspf . . . 548
ground_pins . . . 559
power_pin . . . 559
run_static_simulation . . . 560
setup_simulation . . . 563
load_ir_results . . . 564
print_ir_report . . . 564

© 2020 All Rights Reserved.

print_rlrp_report . . . 567
pin_2_pin_res . . . 568
print_subnode_to_tap_path_report . . . 570
load_em_results . . . 572
print_em_report . . . 573
load_view . . . 576
print_thermal_report . . . 577
analyze_esd_generic . . . 579
analyze_esd_current_density . . . 582
analyze_esd_network . . . 586
esd_optimizer . . . 590
setMultiCpuUsage . . . 592
setDistributeHost . . . 596
create_pgv . . . 600
update_pgv . . . 607
set_mmsim_pgv_nets . . . 609
report_vfi_XL_ron . . . 611
search_res_id . . . 612
res_id_info . . . 614
search_node_id . . . 618
node_id_info . . . 619

17
Environment Variables . . . 621

Environment Setting (cdsenv) Variables . . . 622
Extract xDSPF Form Variables . . . 625
foundry . . . 625
vfiAutoChkBox . . . 625
vfiBlocking . . . 626
vfiCouplingmode . . . 626
vfiColoring . . . 627
vfiCompression . . . 627
vfiCustom . . . 627
vfiDatatype . . . 628
vfiExtractmode . . . 628

© 2020 All Rights Reserved.

vfiGndNet 629
vfiInputdir 629
vfiMaxviarrcount 630
vfiMulticpu 630
vfiNetsfile 631
vfiOutdir 631
vfiOutdspf 631
vfiPgnets 632
vfiPinorder 632
vfiProcess 633
vfiReduction 633
vfiRunname 634
vfiSignalem 634
vfiSubckt 635
vfiSubCondModel 635
vfiSubNode 635
vfiViaArrayspacing 636
vfiViaCnt 636
vfiViaSpacingEdit 637
vfiViaSpacingEditChk 637
vfiTechCorner 638
vfiTechLib 638
vfiTechName 639
vfiTechType 639
vfiTemperature 640
IR/EM Results Form Variables 641
vfiDfiiLayerMapFile 641
vfiElementCount 641
vfiEMOnlyIctFile 642
vfiIRThreshold 642
vfiLayermapFile 642
vfiLayerIndex 643
vfiNoOfThreads 644
vfiQrcRunDir 644
vfiQrcRunName 644
vfiResultDirectoryIr 645

© 2020 All Rights Reserved.

vfiSetShrinkFactor 645
vfiTechFile 646
vfiTypeCheckboxMax 646
vfiTypeCheckboxAvg 647
vfiTypeCheckboxAvgAbs 647
vfiTypeCheckboxAcpeak 648
vfiTypeCheckboxRms 648
vps_results__plots_distributeRange 649
vps_results_plots_OutofRange 649
Shell Environment Variables 651
VFI_LIC_QUEUE 652
VOLTUSFI_DELTA_T 652
VOLTUSFI_DYNAMIC_AC_PEAK 652
VOLTUSFI_SPLIT_ACDC_RULES 652
VFI_USE_OLD_SHAPE_DB 653
VFI_POINT_QUERY 653
VFI_ENABLE_INTERSECTING_SHAPE 653
VFI_SPLIT_NON_RECTANGULAR_SHAPE 654
vfi_dfii_unmap_skip 654
vfi_ignore_conf_file 654
CDS_MMSIM_VOLTUSFI_ROOT 654

18
Variables 657
Overview 658
Using the set_variable Command in the Voltus-Fi-XL Command File 658
set_variable 658
Using the emirutil Command in the EMIR Control File 659
Setting the Configuration Variable in Batch Mode 659
Using the Variables Form in the Voltus-Fi-XL GUI 660
applyRThreshold 662
connectingODLayers 662
consolidatedReport 662
coupledSH 662
deltaT 663

© 2020 All Rights Reserved.

deltaTLayers 663
disableODSH 663
disableHRSH 664
disableLayoutScale 664
dynamicACPeak 664
emLimitScale 665
emVariableValues 666
enableContextAwareSrmsRelaxation 666
enableDesignResistorPeakFactor 667
enableDesignResistorScaling 667
enablePWLDensityInterpolation 667
enableReportScaleParam 668
enableRMSHotSpotWaiving 668
enableSHHeatSink 669
extendHeatSinkRange 669
extendHRRange 669
extendJouleHeating 669
extendODRange 670
extendMetalRange 670
extendPWLMatch 670
extendSHRegion 670
geounit 670
geounit_wl 670
geounit_xy 671
layermap 671
mdLayers 672
mergeMDLayers 672
multiplePeakPercentThreshold 672
optimisticEMRuleSelection 672
powerRailGroundNets 673
powerRailRules 673
powerRailSupplyNets 673
process_scale 674
prioritizedPwlRulesLayers 674
removeBulkShorting 675
reportUnmatchedRuleResistor 675

© 2020 All Rights Reserved.

rmsMetalLineNumberPower . . . 675
rmsMetalLineNumberSignal . . . 676
rmsMultiplierPin . . . 676
rmsMultiplierSignal . . . 676
shEMEffect . . . 676
shParamFile . . . 676
shXMLFile . . . 677
simpleSingleViaDefinition . . . 677
splitACDCRules . . . 677
Tj . . . 677
Tambient . . . 677
Tlife . . . 678
toggleNetFrequency . . . 678
toggleNetFrequencyFile . . . 678
useEncryptedWidth . . . 679
useRecoveryFactor . . . 679
viaCountFromResistance . . . 679

19
SKILL Functions . . . 681

Overview . . . 682
Voltus-Fi SKILL Functions . . . 682
vsaCreateMarker . . . 684
vsaOpenLayout . . . 685
vsaLoadNets . . . 686
vsaLoadSecondaryLayers . . . 687
vsaSetLayersToMergeDuringResultsLoading . . . 688
vsaSetInputType . . . 689
vsaSetEMIRConfig . . . 691
vsaSetEMResultsFile . . . 692
vsaSetEMLayerMapFileName . . . 693
vsaSetEMTechFileName . . . 694
vsaSetEMTypes . . . 695
vsaSetDfiiLayerMapFileName . . . 696
vsaShowAllResistors . . . 697

© 2020 All Rights Reserved.

vsaSetQRCData 698
vsaSelectFailedNets 699
vsaSetEMOnlyICTFile 700
vsaSetEMPlot 701
vsaLoadEM 703
vsaSetIRResultsFile 704
vsaSetIRThreshold 705
vsaSetIRPlot 706
vsaSetLRPShortedLayers 708
vsaLoadIR 709
vsaSetFinerDisplayOff 710
vsaSetParallelLoading 711

20
File Formats 713

Static Current Analysis Files 714
Activity File 715
Clock File 716
Value Change Dump (VCD) File 717
Power File 720
Rail Analysis Files 721
ICT File 722
EM Only ICT File 725
EM Data File 727
Layer Map File 729
Layer Map File 730
DFII Layer Map File 731
Self-Heating Effect Analysis File 733
shParamFile 733
Power-Grid View Creation Files 735
Format and Contents of Trigger File 735
LEF layer Map File 738
PGDB layer Map File 739
Common Command Language (CCL) File 741

© 2020 All Rights Reserved.

21
Voltus-Fi Form Descriptions 743
Create Power Grid Views Form 744
Hierarchical Top Simulation Form 746
Extract xDSPF Form 748
. 758

A
Third-Party DSPF 759
Troubleshooting Information Related to Third-Party DSPF File 759
Problem – Missing Number of Vias 759
Solution 760

B
Converting the EM Data File to ICT File 763
convert_emdatafile_to_ict 763
Supported EM Rules for Converting EM Data File into ICT File 765
Unsupported EM Rules for Converting EM Data File into ICT File 768

Index 771

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

Preface

The Voltus™-Fi Custom Power Integrity Solution-XL (Voltus-Fi-XL) product is used to perform transistor-level power integrity analysis, which includes multi-mode simulation (MMSIM) electromigration and voltage drop (EMIR) analysis for transistor designs.

The target designs for Voltus-Fi-XL include analog, analog/mixed-signal (AMS), and custom digital designs of large sizes that are created using Virtuoso.

This user guide is aimed at developers and designers of integrated circuits and assumes that you are familiar with:

- The Virtuoso design environment and application infrastructure mechanisms designed to support consistent operations between all Cadence® tools.
- The applications used to design and develop integrated circuits in the Virtuoso design environment, notably, the Virtuoso Layout Suite, and Virtuoso Schematic Editor.
- The Virtuoso design environment technology file.
- Component description format (CDF), which lets you create and describe your own components for use with Layout XL.

This preface contains the following topics:

- Licensing Requirements
- Related Documentation
- Additional Learning Resources
- Customer Support
- Feedback about Documentation
- Typographic and Syntax Conventions

Licensing Requirements

For information about the licenses required for running the Voltus-Fi product options, see the "Voltus-Fi Licensing Information" chapter.

© 2020 All Rights Reserved.

Related Documentation

For more information about related products, see the following documents. You can access these and other Cadence documents using the Cadence Help online documentation system.

Voltus-Fi Custom Power Integrity Solution Documentation

- *Voltus-Fi Custom Power Integrity Solution L User Guide*
- *Power IR/EM User Guide*

What's New and KPNS

- The following chapters in the *Virtuoso What's New* manual:
  - *Voltus-Fi Custom Power Integrity Solution L*
  - *Voltus-Fi Custom Power Integrity Solution XL*
- *Power IR/EM Known Problems and Solutions*

Other Documentation

- *Virtuoso Design Environment User Guide*
- *Virtuoso Analog Design Environment L User Guide*
- *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*
- *Virtuoso Layout Suite XL User Guide*
- *Pegasus Interactive User Guide*
- *Virtuoso IPVS User Guide*

Additional Learning Resources

Video Library

The Video Library on the Cadence Online Support website provides a comprehensive list of videos on various Cadence products.

© 2020 All Rights Reserved.

To view a list of videos related to a specific product, you can use the *Filter Results* feature available in the pane on the left. For example, click the *Virtuoso Layout Suite* product link to view a list of videos available for the product.

You can also save your product preferences in the Product Selection form, which opens when you click the *Edit* icon located next to *My Products*.

Virtuoso Videos Book

You can access certain videos directly from Cadence Help. To learn more about this feature and to access the list of available videos, see Virtuoso Videos.

Voltus-Fi-XL Videos

You can access the Voltus-Fi-XL videos at the below link:

Transistor Level Power Signoff with Voltus-Fi (IC618) (Video)

The above channel comprises a series of videos on the Voltus™-Fi Custom Power Integrity Solution-XL (Voltus-Fi-XL) product that is used to perform transistor-level power integrity analysis, which includes Spectre multi-mode simulation (SPECTRE) electro-migration and voltage drop (EMIR) analysis for transistor designs.

Rapid Adoption Kits

Cadence provides a number of Rapid Adoption Kits that demonstrate how to use Virtuoso applications in your design flows. These kits contain design databases and instructions on how to run the design flow.

The following RAK is available for Voltus-Fi-XL:

- Voltus-Fi EMIR Analysis Workshop

In addition, Cadence offers the following training course on Voltus-Fi-XL:

- Transistor Level Power Signoff with Voltus-Fi

To explore the full range of training courses provided by Cadence in your region, visit Cadence Training or write to training_enroll@cadence.com.

Note: The links in this section open in a separate web browser window when clicked in Cadence Help.

© 2020 All Rights Reserved.

Help and Support Facilities

Virtuoso offers several built-in features to let you access help and support directly from the software.

- The Virtuoso *Help* menu provides consistent help system access across Virtuoso tools and applications. The standard Virtuoso *Help* menu lets you access the most useful help and support resources from the Cadence support and corporate websites directly from the CIW or any Virtuoso application.
- The Virtuoso Welcome Page is a self-help launch pad offering access to a host of useful knowledge resources, including quick links to content available within the Virtuoso installation as well as to other popular online content.

  The Welcome Page is displayed by default when you open Cadence Help in standalone mode from a Virtuoso installation. You can also access it at any time by selecting *Help – Virtuoso Documentation Library* from any application window, or by clicking *Home* on the Cadence Help toolbar (provided you have not set a custom home page).

For more information, see Getting Help in *Virtuoso Design Environment User Guide.*

Customer Support

For assistance with Cadence products:

- Contact Cadence Customer Support

  Cadence is committed to keeping your design teams productive by providing answers to technical questions and to any queries about the latest software updates and training needs. For more information, visit https://www.cadence.com/support.
- Log on to Cadence Online Support

  Customers with a maintenance contract with Cadence can obtain the latest information about various tools at https://support.cadence.com.

Feedback about Documentation

You can contact Cadence Customer Support to open a service request if you:

- Find erroneous information in a product manual
- Cannot find in a product manual the information you are looking for
- Face an issue while accessing documentation by using Cadence Help

© 2020 All Rights Reserved.

You can also submit feedback by using the following methods:

- In the Cadence Help window, click *Feedback* and follow instructions.
- On the Cadence Online Support Product Manuals page, select the required product and submit your feedback by using the *Provide Feedback* box.

Typographic and Syntax Conventions

The following typographic and syntax conventions are used in this manual.

text	Indicates names of manuals, menu commands, buttons, and fields.
`text`	Indicates text that you must type just as it is presented. Typically used to denote command, function, routine, or argument names that must be typed literally.
`z_argument`	Indicates text that you must replace with an appropriate argument value. The prefix (in this example, *`z_`*) indicates the data type the argument can accept and must not be typed.
`|`	Separates a choice of options.
`{ }`	Encloses a list of choices, separated by vertical bars, from which you **must** choose one.
`[ ]`	Encloses an optional argument or a list of choices separated by vertical bars, from which you **may** choose one.
`[ ?argName` *`t_arg`* `]`	Denotes a *key argument*. The question mark and argument name must be typed as they appear in the syntax and must be followed by the required value for that argument.
`...`	Indicates that you can repeat the previous argument. Used with brackets to indicate that you can specify zero or more arguments. Used without brackets to indicate that you must specify at least one argument.
`,...`	Indicates that multiple arguments must be separated by commas.
`=>`	Indicates the values returned by a Cadence® SKILL® language function.

© 2020 All Rights Reserved.

/	Separates the values that can be returned by a Cadence SKILL language function.

If a command-line or SKILL expression is too long to fit within the paragraph margins of this document, the remainder of the expression is moved to the next line and indented. In code excerpts, a backslash (\ ) indicates that the current line continues on to the next line.

© 2020 All Rights Reserved.

1

Voltus-Fi Licensing Information

You need the following licenses to use the Voltus-Fi Custom Power Integrity Solution software:

License	License Number	Description
Virtuoso Framework License (running Voltus-Fi-XL in the GUI mode requires checkout availability of the following license sets)		
Cadence Design Framework II (DFII)	111	This license must be checked out at Virtuoso startup.
Virtuoso Analog Design Environment XL (ADE XL)	95210	This license is required to run Voltus-Fi-XL from the Virtuoso Analog Design Environment (ADE L/ADE XL).
Simulator License (you need one of the following licenses for performing simulation)		
Spectre Accelerated Parallel Simulator (Spectre APS) with Spectre Power	91400	This license must be purchased separately. The Spectre Power option enables EMIR analysis with the Spectre APS base product.
Spectre eXtensive Partitioning Simulator (XPS) with Spectre Power	91400	This license must be purchased separately. The Spectre Power option enables EMIR analysis with the Spectre XPS base product.
Voltus-Fi License (the Voltus-Fi Custom Power Integrity Solution software includes the following products. When you run a command to use a product, a license is checked out)		

© 2020 All Rights Reserved.

License	License Number	Description
Voltus-Fi Custom Power Integrity Solution-L (VTS-Fi-L)	VTS400	License string: `Voltus_Power_Integrity_Fi_L` Uses the extracted view to perform IR drop and Electromigration (EM) analysis. It uses the results of the simulation performed using the `vavo-db` or the `psf` data.
Voltus-Fi Custom Power Integrity Solution-XL (VTS-Fi-X)	VTS500	License string: `Voltus_Power_Integrity_Fi_XL` Performs multi-mode simulation (MMSIM) electromigration and voltage drop (EMIR) analysis for power and signal nets using the simulation database generated by Spectre APS or Spectre XPS. It is tightly integrated within Virtuoso. **Note:** With this license, you can also run Voltus-Fi Custom Power Integrity Solution-L (VTS-Fi-L).
Voltus-Fi Custom Power Integrity Solution-AA (VTS-Fi-AA)	VTS501	License string: `Voltus_Power_Integrity_Fi_AA` Supports the following advanced features: ■ Hierarchical EMIR Analysis ■ Hierarchical SHE Analysis
Beta License		
VPS_NG100	VPS_NG100	The following features are beta features and available only with this license: ■ EM analysis on the N3 process node ■ ESD analysis

© 2020 All Rights Reserved.

Voltus-Fi-XL Compatibility with MMSIM/SPECTRE and EXT Product Versions

The table below provides the mapping between the Voltus-Fi-XL product version and the corresponding EXT and MMSIM/Spectre versions that are compatible with it. It is recommended that you use these version specifications to avoid erroneous results.

Voltus-Fi-XL Version	EXT Version	MMSIM/SPECTRE Version
Voltus-Fi-XL ICADVM20.1	EXT 20.1x	SPECTRE 19.1.0 ISRx

Voltus-Fi-XL Compatibility with Voltus IC Power Integrity Solution

The table below provides the mapping between the Voltus-Fi-XL product version and the corresponding Voltus IC Power Integrity Solution (Voltus) version that is compatible with it. It is recommended that you use this version specification to avoid erroneous results.

Important

The power-grid view (PGV) generation flow—Voltus-Fi and Voltus—does not provide backward compatibility with the earlier Voltus versions. It is recommended that you use the version specifications provided below.

Voltus-Fi-XL Version	Voltus Version
Voltus-Fi-XL ICADVM20.1	Voltus IC Power Integrity Solution 19.14

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

2

Getting Started with Voltus-Fi-XL

Voltus-Fi-XL is used for transistor-level power-integrity analysis, which includes multi-mode simulation (MMSIM) electromigration and voltage drop (EMIR) analysis for transistor designs. The target designs include analog, analog/mixed-signal (AMS), and custom digital designs of large sizes that are created using Virtuoso.

This product is tightly integrated within Virtuoso. The results of the analyses are displayed on the Virtuoso layout. Voltus-Fi-XL also generates text reports of the analyses and lets you query the analyses results to view specific violations in the layout. This is used to identify and debug regions of high IR drop and EM violations in the design. After analyzing the EMIR results, you can create power-grid views (PGVs) of the design block, which can then be used in Voltus for mixed-signal analysis.

Voltus-Fi-XL uses the simulation database generated by simulators—Spectre Accelerated Parallel Simulator (APS) and Spectre eXtensive Partitioning Simulator (XPS)—as inputs to perform EMIR analysis.

Related Topics

- Key Features of Voltus-Fi-XL
- Running Voltus-Fi-XL in Batch Mode
- Running Voltus-Fi-XL in GUI Mode
- Accessing Documentation and Help

© 2020 All Rights Reserved.

Key Features of Voltus-Fi-XL

The following table lists the key features of Voltus-Fi-XL.

Feature	Description
Extraction	Supports the generation of the Detailed Standard Parasitic Format (xDSPF) file using Quantus. This file provides all the parasitic information of the design for performing simulation using Spectre.
Simulation-based analysis	Uses APS/XPS—direct or iterated method for EMIR analysis.
Support for multiple simulation results files for EMIR analysis	Supports the following: ■ Multiple bin files that contain simulation results for different nets from the same testbench ■ Multiple bin files that contain simulation results from different testbenches for the same design
Static current analysis	Evaluates IR drop and EM currents based on user-provided subcircuit instance current consumptions without running a transient or DC simulation.
Signal net and power-rail IR drop analysis	Supports the following: ■ Solid shape highlighting ■ Displaying and querying results ■ Analyzing power-gated designs ■ The following plot types: peak and average IR drop, transistor voltage, peak and average tap current, RMS tap current, powergate current, powergate voltage, peak, average, and RMS resistor current, and effective resistance plots

© 2020 All Rights Reserved.

Feature	Description
Signal net and power-rail EM analysis	Supports the following: ■ Solid shape highlighting ■ Displaying and querying results ■ All types of EM analysis: peak, average, rms, AC-peak, and absavg ■ Text-based ICT file for specifying EM rules ■ Advanced EM rules with `qrcTechFile` for 20nm and below technologies ■ The `emDataFile.txt` file for backward compatibility in 28nm and above technologies
Static and dynamic PGV generation	Creates PGVs of an analog design block that can be used in Voltus for mixed-signal analysis.
Advanced debug features	**Structural analysis** is a shape-based geometrical analysis that is used to quickly identify power-grid weaknesses in designs and mark them on the Virtuoso layout for you to view and debug. Structural analysis cuts down on the sign-off analysis time. The following checks are supported: ■ Missing Via analysis ■ Via coverage ratio ■ Skewed via ratio
	Effective resistance is calculated between any two nodes (pins, tap nodes, or subnodes), either on the same net or on different layers of the same net. This feature uses the DSPF file, used for Spectre simulation, and the SPGS feature of Spectre.
	Least-Resistive Path (LRP) analysis helps you identify weakly connected instances in the design during early stages of power planning. It displays the worst IR drop violations and plots the LRP on demand for any node.
	What-If analysis feature lets you analyze the impact of potential layout changes, without implementing these changes in the layout, and re-extracting the xDSPF file. These changes are also called the ECO changes.

© 2020 All Rights Reserved.

Installation Information

- Voltus-Fi-XL is launched within the Virtuoso Platform. To use this product, you must have the Virtuoso ICADVM20.1 release version.
- To check whether Voltus-Fi-XL is available in your Virtuoso Design Environment, type the following in the UNIX window, shell, or xterm:

  ```
  vfibatch -v
  ```

  This prints the product version available in your Virtuoso Design Environment.
- Voltus-Fi-XL is supported only on the Linux (only 64-bit) platform. To specify the 64-bit version, you need to set the `CDS_AUTO_64BIT` environment variable before starting the software. If you are using the `lnx86` operating system, verify that it supports 64-bit applications.
  - To run all applications in 64-bit mode, run the following command:

    ```
    setenv CDS_AUTO_64BIT ALL
    ```
  - To run specific applications in 64-bit mode, such as Spectre and Voltus-Fi-XL, and all other applications in 32-bit mode, run the following command:

    ```
    setenv CDS_AUTO_64BIT spectre:vfibatch
    ```

Related Topics

- Running Voltus-Fi-XL in Batch Mode
- Running Voltus-Fi-XL in GUI Mode

Running Voltus-Fi-XL in Batch Mode

You can run Voltus-Fi-XL as a batch command, running it from Virtuoso in non-graph mode. Batch mode is used when the simulation database is already generated, and you want to print the IR/EM text reports without viewing the plots on the layout.

The batch mode flow of Voltus-Fi-XL supports both the configuration file or the `conf` file and the command file, `cmd_file`.

To run Voltus-Fi-XL in batch mode, do one of the following:

- Use the following command to run Voltus-Fi-XL from the UNIX window, shell, or xterm:

  ```
  vfibatch -cmd <cmd_file>
  ```


© 2020 All Rights Reserved.

Where, `<cmd_file>` is a file with a list of batch commands that define the input data, control the execution of the analysis, and specify the output location.

Note: You can also use the `emirreport` command, from the simulation hierarchy, to generate textual and HTML reports for IR and EM analysis.

- Use the following command to run Voltus-Fi-XL by specifying the EMIR configuration file:

  ```
  vfibatch -c <emir conf> -db <simulation directory path> -outdir
  <output directory>
  ```

  You can specify the control file options in the configuration file using the `emirutil` command.

Related Topics

- Running Voltus-Fi-XL in GUI Mode
- Batch Mode Execution

Running Voltus-Fi-XL in GUI Mode

The flow to run Voltus-Fi-XL in GUI mode comprises the following steps:

1. Starting the Virtuoso Design Environment
2. Opening the Schematic or Layout View
3. Launching Voltus-Fi-XL

Starting the Virtuoso Design Environment

To start the Virtuoso design environment, do the following:

- Type `virtuoso &` in the UNIX window, shell, or xterm.

  The Command Interpreter Window (CIW) opens. You interact with the design environment from the CIW.

© 2020 All Rights Reserved.

Opening the Schematic or Layout View

You need to open the schematic or layout view of your design in the Virtuoso Schematic Editor or Virtuoso Layout Suite, respectively. To do this, use either the *File* menu or the *Tools* menu of the CIW.

File Menu of CIW

1. Choose *File – Open*.

   The Open File form is displayed.

2. In the *File* group box, select a *Library*, select a cell name from the *Cells* list or directly type the cell name in the *Cell* field, and select a *View* (either *schematic* or *layout*).

   Alternatively, you can click *Browse* to open the Library Browser – Open File window and select your design.

3. In the *Application* group box, select an item from the *Open with* drop-down list.

4. Select one of the *Open for* options: *edit* or *read*.

5. Click *OK*.

Tools Menu of CIW

1. Choose *Tools – Library Manager.*

   The Library Manager form is displayed.

2. To select a library and its corresponding cell and view, click the item names in the order: *Library*, *Cell*, and *View*. Select *schematic* or *layout* in the *View* section.

3. Double-click the view to open the design in the Virtuoso Schematic Editor or Virtuoso Layout Suite.

Launching Voltus-Fi-XL

You can run Voltus-Fi-XL from the Virtuoso Design Environment in two ways. You can run it either from the Virtuoso Layout Suite using the plug-in option, or from the Virtuoso Analog Design Environment (ADE L/ADE XL) after adding the DSPF file and performing postlayout simulation with EMIR analysis.

From Virtuoso Layout Suite

1. Choose *Launch – Plugins – Voltus-Fi-XL*.

   The Voltus-Fi-XL console opens.

© 2020 All Rights Reserved.

From ADE L/ADE XL

The steps for launching Voltus-Fi-XL using ADE L are detailed below. The same steps can be used to launch Voltus-Fi-XL using ADE XL.

1. From Virtuoso Schematic Editor or Virtuoso Layout Suite, choose *Launch – ADE L.*

   The ADE L window opens.

2. Set up and run simulation, and generate the simulation database.

3. In the main menu of the ADE L window, choose *Results – EM/IR Data – Layout Analysis*.

4. In the Select Layout View form, select the cell name from the *Cell* drop-down list and click *OK*.

   The following windows open:

   - Voltus-Fi-XL console
   - IR/EM Results form of Voltus-Fi-XL

© 2020 All Rights Reserved.

Voltus-Fi-XL Console

Related Topics

- Running Voltus-Fi-XL in Batch Mode
- Batch Mode Execution
- Voltus-Fi-XL Console Description
- Launching Voltus-Fi-XL from Virtuoso ADE Explorer Environment (video)
- Launching Voltus-Fi-L/XL from Virtuoso Layout Suite for EMIR Analysis (video)

© 2020 All Rights Reserved.

Specifying Simulation Settings in ADE L/ADE XL

Before you run simulation, you need to specify the simulation settings in ADE L/ADE XL. The steps required in ADE L are detailed below. The same steps can be used in ADE XL.

To specify simulation settings in ADE L, do the following:

1. In the main menu of the ADE L window, choose *Setup – Simulation/Directory/Host* and specify the *Simulator* as *spectre*.
2. Choose *Setup – High Performance Simulation* and ensure that *APS* and its *Post-Layout Settings* group box options are selected.
3. Choose *Setup – Model Libraries* and load the appropriate model files.
4. Choose *Setup – Simulation Files* and in the Simulation Files Setup form, specify the *Definition Files* and *Parasitic Files (DSPF)*.

   Note: The Definition File contains the UTI commands. This file is required for generating power-grid views in Voltus-Fi-XL.

Related Topics

- Running Voltus-Fi-XL in GUI Mode
- Specifying EMIR Analysis Settings in ADE L/ADE XL
- Generating Simulation Database in ADE L/ADE XL

Specifying EMIR Analysis Settings in ADE L/ADE XL

Before performing EMIR analysis in Voltus-Fi-XL, you need to specify the EMIR analysis settings in ADE L/ADE XL. The settings required for static and dynamic EMIR analysis are different. The steps required in ADE L are detailed below. The same steps can be used in ADE XL.

Setting up Static EMIR Analysis

Static EMIR analysis estimates the IR and EM current density without performing a circuit simulation based on the user-specified current estimates in the static current file. To set up static EMIR analysis, do the following:

1. In the main menu of the ADE L window, choose *Setup – EM/IR Analysis*.

   The Voltus-Fi XL Analysis Setup form opens.

© 2020 All Rights Reserved.

2. In the *Analysis* tab, select the *Type* of analysis as *Static*.
3. In the *Net Selection* group box, click *Select* to choose the *Net Name* for the nets to be analyzed in the schematic. In the schematic, select an instance. Selecting an instance chooses all the nets of the instance with the wild card "*".
4. Specify the *Static Current File*.
5. Select *avg* current analysis for both *IR Drop Analysis* and *EM Current Analysis*.

   Note: By default, IR drop is calculated only for the power nets and not for the signal nets. For EM analysis, both the power and signal nets are considered.
6. Click *Add/Modify* to add the net and analysis combination statements to the summary information in the EMIR configuration file.
7. In the *Solver* tab, ensure that the *Speed* is set to *default*.
8. In the Option tab, select an option from the Advanced Option drop-down list or specify the required option in the Additional Option field.

   Note: The DSPF file included through the *Simulation Files* submenu of ADE L is automatically populated in the *DSPF File Checking group* box.
9. Click *Run* to invoke the spfchecker.
10. Select the *EMIR Analysis for Static* check box at the bottom of the form.
11. Click *OK* to save the static EMIR analysis settings for the run.

Setting up Dynamic EMIR Analysis

Dynamic EMIR analysis requires a DC or transient simulation. To set up dynamic EMIR analysis, do the following:

1. In the main menu of the ADE L window, choose *Setup – EM/IR Analysis*.

   The Voltus-Fi XL Analysis Setup form opens.
2. In the *Analysis* tab, select the *Type* of analysis as *Dynamic*.
3. In the *Net Selection* group box, click *Select* to choose the *Net Name* for the nets to be analyzed in the schematic. In the schematic, select an instance. Selecting an instance chooses all the nets of the instance with the wild card "*".
4. Select the analysis types from available options: *IR Drop Analysis*, *EM Current Analysis*, and *Advanced IR Drop Analysis*.

   Note: Select the *EM Current Analysis* options, *max* and *avg*, if you want to perform AC-Peak analysis later in Voltus-Fi-XL.

© 2020 All Rights Reserved.

5. Click *Add/Modify* to add the net and analysis combination statements to the summary information in the EMIR configuration file.
6. In the *EM Rule Check Setup* group box, specify the *EM Tech File*.
7. In the *Solver* tab, select *direct* as the *Solver Method*.
8. In the *Time Window Setup* group box, select an option to specify EMIR analysis duration.

   Full Transient is selected by default, which means the EMIR analysis is run over the entire simulation period. You can also select *Time Window* and provide the start and stop times.
9. In the *Option* tab, select an option from the Advanced Option drop-down list or specify the required option in the *Additional Option* field.

   Note: The DSPF file included through the *Simulation Files* submenu of ADE L is automatically populated in the *DSPF File Checking group* box.
10. Click *Run* to invoke the spfchecker.
11. Select the *Enable EMIR Analysis in Transient or DC Simulation* check box at the bottom of the form.
12. Click *OK* to save the dynamic EMIR analysis settings for the run.

The *Summary Information* group box at the bottom of the Voltus-Fi XL Analysis Setup form is updated as you make changes in the form or click *Add/Modify* where applicable. The summary section contains the final set of commands in the EMIR configuration file. This file is later used by Spectre APS for performing EMIR analysis.

Related Topics

- Performing Static EMIR Analysis in the Virtuoso ADE Explorer Environment (video)
- Setting Up and Running Dynamic EMIR Analysis in Virtuoso ADE Explorer (video)
- Running Voltus-Fi-XL in GUI Mode
- Specifying Simulation Settings in ADE L/ADE XL
- Generating Simulation Database in ADE L/ADE XL

© 2020 All Rights Reserved.

Generating Simulation Database in ADE L/ADE XL

After specifying simulation and EMIR analysis settings, you need to run a simulation in ADE L/ADE XL to generate the simulation database.

To do this:

- In the main menu of the ADE L/ADE XL window, choose *Simulation – Netlist and Run*.

  You should use this option when you run the simulation for the first time and also when you have edited your design. Using this option ensures that your design, the ADE setup, and the output netlist for simulation are synchronized. A progress bar appears at the bottom of the simulation window displaying the status of netlisting and simulation runs. You can view the details of the run in the UNIX window, shell, or xterm.

Related Topics

- Running Voltus-Fi-XL in GUI Mode
- Specifying Simulation Settings in ADE L/ADE XL
- Specifying EMIR Analysis Settings in ADE L/ADE XL

Accessing Documentation and Help

You can access the Voltus-Fi-XL documentation and Cadence Help using one of the following methods:

From the Command Prompt

1. Change to the following directory: `installation_dir`/tools/bin
2. Enter the following command:

   ```
   ./cdnshelp
   ```

   After launching Cadence Help, press `F1` or choose *Help – Contents* to display the help page for Cadence Help.

From the Voltus-Fi-XL GUI

Do one of the following:

- Click *Help* in the Voltus-Fi-XL console.
- Click *Help* on the bottom-right corner of any form.

© 2020 All Rights Reserved.

Clicking *Help* opens the Voltus-Fi-XL user guide entry for the form in the Cadence Help window.

From Cadence Online Support

- Select Cadence Online Support on the Virtuoso *Help* menu.

  The Cadence Online Support website is opened in your browser. You are required to have a Cadence Online Support account to access these materials.

Related Topic

- Voltus-Fi-XL Console Description

Voltus-Fi-XL Console Description

The following main menus of Voltus-Fi-XL are available. In addition to the menus, there is a *Stop Process* button in the console. This button is enabled when you run a process. Click this button to stop a process while it is running.

Menu	Description
Extraction	Opens the Extract xDSPF form that lets you generate an xDSPF file.
Simulation	Opens the Setup and Run Current Analysis form that lets you set up and run the static current analysis mode for peak current analysis.
IR/EM Analysis	Opens the following forms: ■ IR/EM Results: Lets you view the IR drop and EM analysis results. ■ Create Power Grid Views: Lets you create PGVs. ■ Connect and vfi: Lets you connect to the port and opens the vfi form using which you can view the IR drop and EM analysis results.

© 2020 All Rights Reserved.

Menu	Description
Hierarchical EMIR	Opens the following forms for performing hierarchical EMIR analysis: ■ Create Power Grid Views: Lets you create block-level PGVs and xDSPF files. ■ Extract xDSPF: Lets you generate the top-level xDSPF file. ■ Hierarchical Top Simulation: Lets you perform top-level simulation.
ESD Analysis	Opens the ESD form that lets you perform electrostatic discharge (ESD) analysis.
Help	■ Opens the *Voltus-Fi Custom Power Integrity Solution XL User Guide* and the Voltus-Fi Custom Power Integrity Solution XL What's New chapter in the *Virtuoso What's New* document. ■ Provides the version information of Voltus-Fi-XL.

Related Topics

- Key Features of Voltus-Fi-XL
- Accessing Documentation and Help

© 2020 All Rights Reserved.

3

Product Overview

The Voltus-Fi-XL product is used for transistor-level power-integrity analysis, which includes multi-mode simulation (MMSIM) electromigration and voltage drop (EMIR) analysis for transistor designs. The target designs include analog, analog/mixed-signal (AMS), and custom digital designs of large sizes that are created using Virtuoso.

This product is tightly integrated within Virtuoso. The results of the analyses are displayed on the Virtuoso layout. Voltus-Fi-XL also generates text reports of the analyses and lets you query the analyses results to view specific violations in the layout. This is used to identify and debug regions of high IR drop and EM violations in the design. After analyzing the EMIR results, you can create power-grid views (PGVs) of the design block, which can then be used in Voltus for mixed-signal analysis.

Voltus-Fi-XL uses the simulation database generated by simulators—Spectre® Accelerated Parallel Simulator (APS) and Spectre® eXtensive Partitioning Simulator (XPS)—as inputs to perform EMIR analysis.

A brief description of the capabilities of the two simulators is provided below.

- Spectre APS is a new-generation SPICE simulator that provides high performance, high capacity circuit simulation with full Spectre accuracy. APS achieves maximum simulation performance by enabling multi-threading on multi-core and multi-CPU shared memory systems.
- Spectre XPS is a new-generation transistor-level circuit simulator that emphasizes high simulation performance and large simulation capacity, and addresses the design and verification needs of full-chip low-power designs at advanced process nodes.

Note: For detailed information about Spectre APS/XPS, see “Introducing the Spectre Circuit Simulator” in *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

Key Product Features and Description

The following table lists the key features of Voltus-Fi-XL.

© 2020 All Rights Reserved.

Table 3-1 Product Features – Voltus-Fi-XL

Feature	Description
Extraction	Supports the generation of the Detailed Standard Parasitic Format (xDSPF) file using Quantus. This file is provides all the parasitic information of the design for performing simulation using Spectre. For details, see the "xDSPF Generation using Quantus" chapter.
Simulation-based Analysis	Uses APS/XPS—direct or iterated method for EMIR analysis. For details of the two methods, see Spectre APS/XPS EMIR Flow Overview in the "Data Preparation" chapter.
Multiple Simulation Results Files for EMIR Analysis	Supports the following: ■ Specifying multiple bin files that contain simulation results for different nets from the same testbench. ■ Specifying multiple bin files that contain simulation results from different testbenches for the same design For details, see the "IR Drop Analysis Results" chapter.
Static Current Analysis	A static EMIR analysis evaluates IR drop and EM currents based on user-provided subcircuit instance current consumptions without running a transient or DC simulation. For details, see the "Static Current Analysis" chapter.

© 2020 All Rights Reserved.

Viewing IR Drop Analysis Results	Signal net and power-rail IR drop analysis includes support for the following: ■ Solid shape highlighting ■ Displaying and Querying Results ■ Analyzing power-gated designs ■ The following plot types; peak and average IR drop, transistor voltage, peak and average tap current, RMS tap current, powergate current, powergate voltage, peak, average, and RMS resistor current, and effective resistance plots For details, see the "IR Drop Analysis Results" chapter.
Viewing EM Analysis Results	**Signal net and power-rail EM analysis** includes support for the following: ■ Solid shape highlighting ■ Displaying and Querying Results ■ All types of EM analysis: peak, average, rms, AC-peak, and absavg ■ Text-based ICT file for specifying EM rules ■ Advanced EM rules with `qrcTechFile` for 20nm and below technologies. ■ The `emDataFile.txt` file is supported for backward compatibility in 28nm and above technologies. For details, see "EM Analysis Results" chapter.
PGV Generation	**Static and Dynamic PGV generation** feature lets you create power-grid views (PGVs) of an analog design block that can be used in Voltus for mixed-signal analysis. For details, see "Power-Grid View Generation" chapter.

© 2020 All Rights Reserved.

Advanced Debug Features	**Structural analysis** is a shape-based geometrical analysis that is used to quickly identify power-grid weaknesses in designs and mark them on the Virtuoso ® layout for you to view and debug. Structural analysis cuts down on the sign-off analysis time. The following checks are supported: ■ Missing Via analysis ■ Via coverage ratio ■ Skewed via ratio For details, see "Structural Analysis" chapter.
	Calculating Effective Resistance between any two nodes (pins, tap nodes, or subnodes), either on the same net or on different layers of the same net. This feature uses the DSPF file, used for Spectre simulation, and the SPGS feature of Spectre. For details, see "IR Drop Analysis Results" chapter.
	Least-Resistive Path (LRP) analysis helps you identify weakly connected instances in the design during early stages of power planning. It displays the worst IR drop violations and plots the LRP on demand for any node. For details, see "IR Drop Analysis Results" chapter.
	What-If Analysis feature lets you analyze the impact of potential layout changes, without implementing these changes in the layout, and re-extracting the xDSPF file. These changes are also called the ECO changes. For details, see "What-If EMIR Analysis" chapter.
	Self Heating Effect (SHE) analysis flow visualization and the thermal-aware DC EM flow visualization. You can view the results of the these analyses by plotting them on the Virtuoso layout to show the worst deltaT violations. You can then debug your design to reduce the self-heating effect. For details, see "Self-Heating Effect Analysis" chapter.

© 2020 All Rights Reserved.

	ESD (electrostatic discharge) analysis lets you check whether every power or ground pin or bump has an ESD device along the potential ESD paths or not. In addition, it analyzes the device placement to ensure that the device does not violate the effective resistance limit. For details, see "ESD Analysis" chapter.
Viewing Results in vfiplot	The **vfiplot utility** is used to view the results of the IR drop and EM analysis that is performed in Voltus-Fi-XL. The key benefits of viewing the results of IR drop and EM analysis in vfiplot are as follows: ■ No dependency on the Virtuoso layout ■ Increased speed of visualization For details, see "Viewing Results in vsaplot or vfiplot" chapter in the *Voltus-Fi Custom Power Integrity Solution L User Guide*.

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

4

Data Preparation

- Voltus-Fi-XL Flow Overview on page 52
  - Spectre APS/XPS EMIR Flow Overview on page 52
- Data Requirements for Voltus-Fi-XL on page 55
  - xDSPF Requirements on page 55
  - Simulation Requirements and Setup on page 56
  - EMIR Control File Options Supported in Voltus-Fi-XL on page 57
- Outputs of EMIR Analysis on page 89

© 2020 All Rights Reserved.

Voltus-Fi-XL Flow Overview

The Voltus-Fi-XL-Spectre® Accelerated Parallel Simulator (APS)/eXtensive Partitioning Simulator (XPS) flow is a two-step flow.

Step1: Simulation is run using Spectre APS/XPS and the simulation database is generated. For details of the Spectre APS/XPS flow, see Spectre APS/XPS EMIR Flow Overview.

Step2: The simulation output generated in the above step is provided as input to Voltus-Fi-XL, which performs the following electromigration and voltage drop (EMIR) analyses:

- Power IR/EM analysis
- Signal IR/EM analysis
- Structural analysis

Voltus-Fi-XL generates text reports of IR drop and EM violations and displays them as plots on the Virtuoso® layout. It also provides batch mode support for IR/EM analysis and report generation.

Note: Voltus-Fi-XL supports both static and dynamic EMIR analysis.

Spectre APS/XPS EMIR Flow Overview

With MMSIM/SPECTRE12.1, Spectre APS and Spectre XPS, Cadence delivers a new EMIR solution. This dynamic power EMIR and signal EM capability uses a new technology, and is designed to provide higher capacity and better performance compared to existing EMIR solutions. Within this flow, Spectre APS can be used for high accuracy EMIR analyses; while Spectre XPS can be deployed for high performance and high capacity EMIR simulation.

In an EMIR flow, a circuit is evaluated together with the parasitic resistor and capacitor network, which models the IR drop or the EM effect. There are two general approaches of solving such a problem.

1. **Direct EMIR analysis**: When high accuracy is needed, a brute-force simulation of the entire system – circuit plus parasitic resistances and capacitances – can be performed to accurately calculate EMIR of any net. This approach is called the “one-step” method, where the EMIR simulation performance and capacity is subject to the limitation of the circuit simulator being used.
2. **Iterated EMIR analysis**: To conduct EMIR simulation on circuits with much larger power and signal nets, within a shorter time, there is an alternative approach that involves decoupling the nonlinear circuit simulation from the linear RC net analysis. You can iterate

© 2020 All Rights Reserved.

the linear RC net analysis by modifying the layout, however, the nonlinear circuit simulation is done only once. This approach is called the “two-step” method. The decoupling of the linear RC nets from the nonlinear circuit is not mathematically equivalent to the original design and certain inaccuracy is introduced but it provides the benefit of simulation performance and capacity.

The above two approaches are depicted in the figure below.

Figure 4-1 Direct and Iterated EMIR Analysis Methods

For high accuracy EMIR analysis of small design blocks, or designs with smaller numbers of RC nets, the direct EMIR analysis method is recommended, where Spectre APS and Spectre XPS SPICE mode are fully supported. Spectre XPS does not support direct EMIR analysis.

To gain higher performance and higher capacity on medium to large designs, the iterated EMIR analysis method is recommended. Spectre XPS is used for these large-scale simulations as it is able to take full advantage of the FastSPICE technology. Both, Spectre APS and Spectre XPS SPICE mode support iterated EMIR analysis.

The Spectre APS/XPS EMIR flow requires a complete testbench that contains Detailed Standard Parasitic Format (DSPF) files – with parasitic and instance sections describing the circuit to be analyzed – and stimuli, device models, and so on. The presence of a `+emir` statement on the Spectre command line enables the EMIR analysis during circuit simulation. The details of EMIR analysis, including the type of analyses to be performed, nets to be analyzed, and the output to be generated, are specified using an EMIR configuration file.

© 2020 All Rights Reserved.

Using either direct EMIR analysis or iterated EMIR analysis, the circuit simulation is performed first with various voltages and currents being calculated. Next, a standalone tool, `emirutil`, is called to post-process the simulation results and generate IR and EM reports. By default, this postprocessing step is invoked automatically.

The output of the EMIR analysis is an EMIR text report, which lists the EM and/or IR information of the specified nets. In addition, an OA database is generated to enable the visualization of the EMIR results on the Virtuoso layout using Voltus-Fi Custom Power Integrity Solution tool.

This flow is shown in the below figure.

Figure 4-2 Voltus-Fi-XL – Spectre APS/XPS Data Flow

© 2020
All Rights Reserved.

For details of Spectre APS/XPS EMIR flow and how to activate the EMIR analysis, see the EMIR Flow Setup section in the "Postlayout Simulation" chapter of the *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

Data Requirements for Voltus-Fi-XL

The following inputs are required for EMIR analysis in Voltus-Fi-XL:

- A simulation output directory that contains the EMIR database
- The `qrcTechFile` with EM Models for EM rules, text-based ICT file, or the `emDataFile.txt` file with EM rules
- Layer map file – this file is required to map the layer names in the simulation result file and the technology file

  Note: This file is required only if the layer names in the two files are different.
- APS/XPS-to-DFII layer map file for solid shape display and structural analysis
- Net names (optional)
- Information about the type of analysis to be performed (peak, avg, rms, absAvg)

The above data requirements are categorized under the following heads and explained in detail in subsequent sections:

- xDSPF Requirements
- Simulation Requirements and Setup
- Outputs of EMIR Analysis

xDSPF Requirements

The third-party netlist file or xDSPF is used to provide all the parasitic information of the design for performing simulation using the Spectre APS/XPS simulators. The xDSPF must include the following information:

- All power ground nets extracted
- Layer information with each resistor
- x, y co-ordinates of the parasitic resistors, and
- Decoupled RC of all signal nets

© 2020 All Rights Reserved.

DSPF files are created using the parasitic extraction tools and their content and format is heavily dependent on the extraction tool and the extraction runset. Often simulation problems occur due to problems in the DSPF file. To ensure that the DSPF file is compatible with the simulation requirements use the Standard Parasitic Format (SPF) Checker utility before using the file in the MMSIM/SPECTRE EMIR flow.

The SPF Checker utility analyzes the DSPF file, reports problems that may cause simulation problems, and creates an EMIR conf file with recommended mapping statements.

For details of how to use the SPF Checker, see the SPF Checker section in the “Postlayout Simulation” chapter of *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

Note: Ensure that there is no pin on the substrate shorting in the xDSPF. Set the extractor accordingly.

Note: If you are using third-party DSPF file, you may encounter some problems while performing the EMIR analysis. For possible issues and how to resolve them, see Third-Party DSPF.

Note: Voltus-Fi-XL supports the generation of the xDSPF file using Quantus both from the GUI and in batch mode. For details of generating the xDSPF file from within Voltus-Fi-XL GUI, see “xDSPF Generation using Quantus” chapter. For details of batch mode command used for generating the xDSPF file, see extract_xdspf in the “Batch Mode Execution” chapter.

Simulation Requirements and Setup

For viewing IR drop and EM analysis results in Voltus-Fi-XL, the simulation database generated using Spectre APS/XPS simulators is loaded into Voltus-Fi-XL. This is shown in Figure 4-2 on page 54 .

For EMIR analysis, ensure the following while performing simulation.

- Create a complete simulation testbench with the DSPF files containing the post-layout data of the design, including fingered devices and net parasitics. Use the `dspf_include` statement to read the DSPF content, as shown below.

  ```
  dspf_include "sram.spf" (Spectre syntax)
  .dspf_include "sram.spf" (SPICE syntax)
  ```

  Note: Though `include/.include` statements are also supported, it is recommended to use `dspf_include/.dspf_include` statements, which automatically handle duplicated subcircuit definitions and port order differences.

© 2020 All Rights Reserved.

Important

Use the SPF Checker utility before using the file in the MMSIM/SPECTRE EMIR flow.

- Define the EMIR configuration using the EMIR control file `(emir.config)`, and include the control file with the `+emir` statement on the Spectre command line, as shown below.

```
% spectre +aps +emir=emir.conf input.scs (APS solver)
% spectre +xps +emir=emir.conf input.scs (XPS solver)
```

Note: The supported EMIR control file options are provided in Supported EMIR Control File Options on page 58.

- Before plotting EMIR results, ensure that the simulation result file is available in the output directory of the simulation results.

  The naming convention for the simulation results file is; `*.emir#_bin.` For example, the result file could be named, `xps.emir0_bin.` This file is required for displaying IR and EM plots and generating text reports in Voltus-Fi-XL.

  Note: In the APS direct method of simulation, no intermediate database files are created. However, in the iterated method, EMIR files have the emirtap file extension. The 1st stage EMIR circuit simulation creates the tap device voltage profiles that are stored in the pwl files, and used in the 2nd stage. The 2nd stage RC network solver writes all IR drop and EM current values into the binary database with the extension `emir0_bin`. This binary database is used by `emirutil/emirreport` for creating the text/HTML/CSV reports, and by Voltus-Fi-XL to visualize the results in the layout.

```
Direct method: design.emir0_bin, design.rpt_ir, design.rpt_em
Iterated method: design.emirtap.emir0_bin, design.emirtap.rpt_ir,
design.emirtap.rpt_em
```

  All intermediate and output files are written into the Spectre output (raw) directory.

  For large designs, you can compress the intermediate (`pwl, pdb`) and the EMIR database files (`emir0_bin`) by specifying the "`eisopt zipfiles=2`" option in the EMIR conf file.

EMIR Control File Options Supported in Voltus-Fi-XL

The following table summarizes the EMIR control file options supported in Voltus-Fi-XL. These are specified in the configuration file using the `emirutil` command. The following table summarizes the supported EMIR control file options:

© 2020 All Rights Reserved.

Table 4-1 Supported EMIR Control File Options

Keyword	Option Set	Explanation	Default Value
net	`name=[instance1.net1instance1.net2....]`	Defines the nets for which the analysis is performed. instance defines the instance of the subcircuit containing the net. net defines the net name inside the subcircuit instance as it is defined in the `DSPF *| NET` definition. If the DSPF file is included on the top level, the instance name is not required. Wildcarding is supported for net names but not instance names.	`none`
	`analysis=[imax iavg iavgpos iavgneg iavgabs iacpeak irms vmax vavg]`	peakEM, avgEM, avgEM for i>0, avgEM for i<0, avgabsEM, rmsEM, peakIR, avgIR **Note**: To perform AC-Peak analysis in Voltus-Fi-XL, ensure that the `imax` and `iavg` options are specified in the control file.	`iavg, vmax`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	`pwrgate=[powernet1 powernet2...]`	Enables power gate handling. You need to specify the power supply net driving the power gate and the internal power supply net driven by the powergate. If one power supply net drives multiple power gates, then you need to specify one statement with all internal power supply nets. For example:
		
		If the internal and external net names are as follows and XTOP is the top cell name:
		
		`pair1 ->[VDD VSS]`
		`pair2 ->[VDDM VDDM1]`
		`pair3 ->[VDDM VDDM2]`
		
		Then the following settings can be specified where the second setting will not override the first:
		
		`pwrgate=[XTOP.VDD XTOP.VSS]`
		
		`pwrgate=[XTOP.VDDM XTOP.VDDM1 XTOP.VDDM2]`
			`none`

	`analysis=[sigvmax sigvavg]`
	`reftype=[avg max]`
	
	`[findsrc=yes]`
		`sigvmax` - max signal net IR drop.
		
		`sigvavg` - avg signal net IR drop.
		
		`avg` - reference voltage for IR drop is the average voltage of all subnodes at every time point.
		
		`max` - reference voltage for IR drop is the maximum voltage of all subnodes at every time point.
		
		`findsrc=yes` - Use voltage from subnode which is connected to vsource. The feature automatically traces through the design or source resistors.
			`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`analysis=[ipwc] pwc_threshold=1e-6`	Enables pulse-wise EM current calculation for violation visualization. The `pwc_threshold` option defines the current threshold for finding the start of the pulse. Any value below `pwc_threshold` is considered as zero. The `emirutil_dump_pwc_info=1` option generates an ascii report in the `<name>.pwc0_raw` file for debugging purpose only.	`none`
	`vref`	Defines the optional voltage reference value for `vmax` and `vavg` analyses.	`none`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	For versions before MMSIM/SPECTRE15.1 ISR2, the syntax is as follows:
	`design_res_models=[name=<cell_name> w=<width parameter name> l=<length parameter name> layer=<layer name> unit=<multiplier to get length/width * geounit in um>]`
	For MMSIM/SPECTRE15.1 ISR2 and later versions, the syntax is as follows:
	`design_res_models=[name=<cell_name> w=<width parameter name> l=<length parameter name> layer=<layer name> unit=<multiplier to get length/width in m>]`
		Specifies the design resistor models for which EMIR analysis will be performed.
		Note: The design resistor models are specified for cells and not for the instances of the cell.
		For example, for the design resistor provided in the xDSPF file as:
		`XR1 XR1#PLUS XR1#MINUS r_user wr=1e-06 lr=5e-06`
		In this example, width (`wr`) and length (`lr`) are in meters.
		The design resistor models will be specified in the following manner:
		Before MMSIM/SPECTRE15.1 ISR2,
		`design_res_models=[name=r_user w=wr l=lr layer=mt1 unit=1e6]`
		MMSIM/SPECTRE15.1 ISR2 and later:
		`design_res_models=[name=r_user w=wr l=lr layer=mt1 unit=1]`
		where,
		`r_user` is the design resistor model name
		`lr` is the length parameter name specified in the xDSPF
		`wr` is the width parameter name specified in the xDSPF
		`mt1` is the layer for the EM rules. The tool will apply `mt1` layer rules from the ICT file for this resistor
			`none`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

		Before MMSIM/SPECTRE15.1 ISR2
		`1e6` is the unit that is calculated as design resistor `W/L` in netlist `* unit/geounit`. For example:
		For `lr=5e-06` and `unit=1e6`, the length value of design resistor in `um` is as follows:
		`l=5e-06*1e06=5um`
		MMISM15.1ISR2 and later
		`1` is the unit that is calculated as design resistor `W/L` in netlist `* unit`. For example:
		For `lr=5e-06` and `unit=1`, the length value of design resistor in meters is as follows:
		`l=5e-06*1=5e-06m`	

`solver`	`method=[auto | direct | iterated | profonly | iteronly]`	`auto` - direct for APS, iterated for XPS (default)| `direct` - forced direct method, not available for XPS `iterated` - forced iterated approach, available for APS and XPS `profonly` - iterated flow - circuit profile generation `iteronly` - iterated flow - RC network iteration	`auto`
	`speed=[1|2|3|4|5|6|7|8]`	Defines the speed/accuracy trade-off for PN and SN RC network solver. `1` - highest accuracy, lowest speed `8` - lowest accuracy, highest speed	`5`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`inputwf=file_name`	Specifies the waveform file that contains the results from circuit simulation when `method=iteronly` is used.	`none`
`static`	`ifile="filename"`	Enables static EMIR analysis. The file defines the subckt instance port currents. **Note**: Static EMIR analysis cannot be combined with the dynamic EMIR analysis.	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
emirutil	`techfile="emDataFile"`	Specifies the EM rule file in the emdatafile, qrctechfile, or ictfile format. Layer names that are not case-sensitive are supported, allowing easier match between the DSPF and `emData` file. The EM rule file can be predefined in the configuration file using the `EMTECHFILE` environment variable or in the `.cdsinit` file as follows: `envSetVal("spectre.envOpts" "emTechFile" string <path to the file>` The rule file defined in the configuration file has a higher priority followed by the rule file specified in `.cdsinit`. Alternatively, you can use the `emdatafile="emDataFile"` option to specify the techfile in the emdata format, `qrctechfile="qrcTechFile"` to specify the Quantus tech files, and `ictfile="ictfile"` to specify the file in the ICT format. The EM rule file in emdatafile format can be predefined using the EMDATAFILE environment variable, or emRuleFile in `cdsenv`. The EM rule file defined in `cdsenv` has a higher priority. The `layermap` file provides the mapping between the xDSPF and the layout database. It is applicable only for `qrcTechfile` and `ictfile`.	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`EMOnlyICTFile="emOnlyICTFile"`	Specifies the process and EM models to be used for EM analysis. **Note**: This file can only be provided with the `qrcTechFile` flow.	`none`
	`ir_in_em=[true | false]`	When set to `true`, reports the maximum IR drop value for each metal resistor/via in the EM current report. It adds a "Max IR" column in the EM report. **Note:** This value is the higher IR drop value of the two nodes of the resistor.	`false`
	`tap_model_report=yes | no`	When set to `yes`, the software reports the device model names for all the tap devices in the dynamic and static analysis report files – `*.rpt_ir` and `*.spgs`.	`none`
	`layerSetupFile=`*`filename`*	Specifies the name of the layer setup file to be used for the Voltus-Fi Quantus shape server flow. By default, the Voltus-Fi Quantus shape server flow reads the "`layer_setup`" file from the path specified in the "`_qrc_tech_dir`" file of the `qrc run dir`. Use this option to specify a file other than the default file. For example: `emirutil layerSetupFile=layer_setup_1`	`none`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`mdLayers="mdlayer1 mdlayer2 mdlayer#"`	Specifies the list of metal diffusion layers that need to be combined for EM analysis. For example, to calculate the distance between terminal vias (`Lv2v`), for thermal analysis (SHE) and so on. The default value is shown below. `emirutil mdLayers="MD_STI MD_OD MD_STI_SRM MD_OD_SRM M0_OD M0_STI MD_STI_CPP85 MD_OD_CPP85"`	`MD_STI MD_OD MD_STI_SRM MD_OD_SRM M0_OD M0_STI MD_STI_CPP85 MD_OD_CPP85`
	`mergeMDLayers=[true | false]`	Enables combining of metal diffusion layers that are specified using the variable, `mdLayers`, for calculating the effective Blech length. When set to `false`, no merging of metal diffusion layers is done for the Blech length calculation. `emirutil mergeMDLayers=true`	`true`
	`mt=8`	Specifies the number of CPUs used by clients for multithreading. `emirutil mt=8`	`8`
	`postTclCmdFile=[conf.tcl]`	Specifies the Tcl file that contains user-specified commands for querying specific resistor and node information and creating customized EM and IR drop analysis reports. For details about the commands that can be written in the file (`conf.tcl`), see the “Batch Mode Execution” chapter.	`none`
	`layermapfile="mapfile"`	Specifies the layer map file that provides the mapping between the layer names in the simulation database to the layer names in the technology file.	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`layermap=[(rcxLayer=rcx_layer_name1 ictLayer=ict_layer_name1) (rcxLayer=rcx_layer_name2 ictLayer=ict_layer_name2)]`	Specifies the mapping between the layer names in the simulation database to those specified in the technology files (qrcTechFile and ICT file) for the EM analysis flow. It maps the layer names in the xDSPF file to the ICT layer names. `rcxLayer` is the layer name in the simulation database. `ictLayer` is the layer name in the ICT EM file. Example: `emirutil layermap=[(rcxLayer=M1 ictLayer=T1) (rcxLayer=M2 ictLayer=T2)]` **Note:** This option is used if the `layermapfile` option is not available.	`none`
	`autorun=[true | false]`	`true` - run `emirutil` automatically to generate a text report. `false` - manually run `emirutil`.	`true`
	`reportCurrentAmp=[true | false]`	`true` - reports the current of the resistor in ampere (A). `false` - reports the current of the resistor in milliampere (mA).	`true`
	`report=[text | html | csv]`	Defines the type of report created by `emirutil`. `text` - creates a report in text format. `html` - creates the report in HTML format. `csv` - creates the report in CSV format. **Note**: Multiple entries are supported.	`text`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`report_voltage=[true | false]`	When set to `true`, IR reports display the absolute voltage value instead of the IR drop value and in the GUI, the Virtuoso layout displays the absolute voltage plots instead of the voltage drop plots. `emirutil report_voltage=true`	`false`
	`notation=[s | e]`	Notation for the text and html reports. `e` - engineering scale number (for example, 5.02m) `s` - scientific notation (for example, 5.02e-3)	`e`
	`sort_by_net=yes|no`	Report IR and EM results per net, or all nets. `yes` - report EMIR results per net `no` - combine EMIR results or all nets into one report.	`yes`
	`power_gate_extra_report`	When specified, adds additional information in the `rpt_pwg` file about power gates. This option is used only for power-gated designs.	`none`
	`filter_ir_threshold=`*`value`*	Defines the IR voltage drop threshold for the node being reported. `filter_ir_threshold=0.01` reports all nodes with IR drop above `0.01V`.	`no filter`
	`filter_ir_percent=`*`value`*	Defines the percentage of nodes being reported starting from the biggest IR drop. For example, `filter_ir_percent=5` reports the top 5 percent IR drop nodes in the report.	`no filter`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`filter_em_threshold=value`	Defines the EM analysis threshold value for the resistors being reported. The value for `filter_em_threshold` is the J/Jlimit value. `filter_em_threshold=1` reports only those resistors that have a J/Jlimit value above 1. **Note**: Specifying this option is equivalent to filtering all passed resistors.	`none`
	`geounit=value`	Defines the scaling of all DSPF geometry parameters (`W, L, X, Y)`. It applies only to the `*|NET` section, and not to the `instance` section.	`1um`
	`geounit_XY`	Defines the scaling of DSPF geometry parameters X and Y. It applies only to the `*|NET` section, and not to the `instance` section. **Note**: If the `geounit_XY` and `geounit` options are specified together, the `geounit_XY` option takes precedence over the `geounit` option.	`1um`
	`geounit_WL`	Defines the scaling of DSPF geometry parameters W and L. It applies only to the `*|NET` section, and not to the instance section. **Note**: If the `geounit_WL` and `geounit` options are specified together, the `geounit_WL` option takes precedence over the `geounit` option.	`1um`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`enableDesignResistorPeakFactor=[true | false]`	Specifies the addition of the factor, $1/(r)^{1/2}$, in the current limit calculation for the AC-Peak analysis of the design resistor. The software, by default, calculates the current limit for the AC-Peak analysis of the design resistor using the following equation: $\mathbf{I}_{peak_ac(limit)} = \mathbf{I}_{peak_dc(limit)}$ To apply the factor, $1/(r)^{1/2}$, in your calculation, set this variable to `true`. The current limit will then be calculated in the following manner: $I_{peak_ac(limit)} = I_{peak_dc(limit)}/(r)^{1/2}$	`false`
	`enableDesignResistorScaling=[true | false]`	Enables the scaling of the width and length values based on the `layout_scale` factor specified in the technology file before the EM rules are applied for EM analysis of design resistors. By default, the scaling of design resistors is enabled.	`true`
	`enableResShape true | false`	When set to `true`, enables tagging of shapes on the layout based on full resistor shapes and terminal nodes. This option is enabled by default. To disable tagging of shapes, set the variable to false. `emirutil enableResShape false`	`true`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`enableRMSHotSpotWaiving=[true | false]`	Enables the RMS EM hot spot waiving analysis. By default, this feature is disabled. **Note:** This variable is applicable only to the RMS EM hot spot waiving feature, which is a limited-access feature. To use this feature, contact your Cadence representative and explain your usage requirements.	`false`
	`extendPWLMatch=[true | false]`	When the EM rules are specified in terms of PWL and the width of metal or the area of via does not match with any width/area of the PWL, the EM limit will not be calculated for this metal resistor/via. When set to `true`, the software interpolates the limit value for the `resistor/via` from the selected rule PWL as follows: minimum of PWL i^{th} value * (metal's width or via's area) / (PWL i^{th} width or PWL `area`).	`false`
	`enablePWLDensityInterpolation=[true | false]`	Enables the current density calculation from the given PWL using current density and width, or current density and area linear interpolation for either the metal's width or the via's area. The calculation is only done when the metal's width or the via's area does not match the PWL width or area values. `emirutil enablePWLDensityInterpolation=true`	`false`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`viaCountResistance=[true | false]`	This option is set to `true` when third-party DSPF with missing via count ($N) is used for EM analysis. For more information, see the Solution section.	`false`
	`process_scale=`*`value`*	Specifies the value by which the width and length values will be multiplied before the EM rules are applied.	`1.0`
	`prioritizedPwlRulesLayers=[all | none |` *`list_layers`*`]`	Specifies the list of layers for which area-based PWL search is performed first. When no match is found, then non-PWL rules are used. The default value of this variable is `"all"`, which means that this behavior is enabled for all layers. To enable the behavior for specific layers, you can specify a list of layers. To disable this behavior for all layers, set the value of this variable to `none`. In the GUI, you can set this variable in the Variables form. Example rules: `em_jmax_dc_avg 10 jmax_factor 90 2.384 95 2.078` - non-PWL baseline rule `em_jmax_dc_avg PWL 0.3582 0.001444 0.5694 0.003364 0.7164 0.0038 jmax_factor 90 2.384` - PWL rule When both the above rules are specified, the second rule is given priority over the first.	`all`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`via_count=[useR | useArea]`	Defines the method used for calculating the via count. `useR` - calculation based on via resistance `(emdatafile_via_resistance/dspf_via_resistance)`. `useArea` - calculation based on via area `(dspf_via_area/emdatafile_via_area)`.	`useR`
	`idirn=[true|false]`	Prints the current direction in the EM report (*Current Direction* column). For metal resistors, the current direction is printed. That is, *W* means that the current is flowing from east to west. For via resistors, the direction from/to layer is printed.	`false`
	`recovery_factor`	Defines the recovery factor for iavg calculation with recovery factor. If specified, then `iavg=max(|iavgpos|,|iavgneg|) - recovery_factor*min(|iavgpos|,|iavgneg|)`. If not specified, then `iavg=iavgpos+iavgneg`	
	`Tlife="hours" | "years"`	Specifies the lifetime for which the EM analysis will be performed. You can specify the value in either `years` or `hours`. For example: `emirutil Tlife="20000hours"`	`"10 years"`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	`emLimitScale=`*`scale_factor_value`* | *`"scale_factor_netfile"`*	Scales the EM limits of the parasitic resistors in the design before generating the EM reports. You can either apply the same scale factor to all the nets or specify different scale factors for different nets.
		
		To apply the same scale factor to the parasitic resistors of all the nets in the design, provide the value of the scale factor.
		
		`emirutil emLimitScale=`*`scale_factor_value`*
		
		For example,
		
		`emirutil emLimitScale=0.04`
		
		Note: This is applicable to `avg`, `avg abs`, and `rms` analyses. For the `avg` and `avg abs` analyses, the limit is directly multiplied by this factor. For rms analysis, the limit is multiplied by the root of this factor.
		
		To apply different scale factors to different nets in the design, provide the name of the file that contains information about the net names and the corresponding scale factors to be applied to these nets.
		
		`emirutil emLimitScale="`*`scale_factor_netfile`*`"`
		
		For example,
		
		`emirutil emLimitScale="scaleLimitFile"`
			`none`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`rmsMetalLineNumberPower=`*`number`*	Specifies the value of power on the metal lines for RMS EM analysis of power nets. If this option is not specified, the default value of `0` is considered, which means the software uses the `em_dc_rms_metal_line_num` value specified in the ICT file. If, however, this option is specified, it overrides the `em_dc_rms_metal_line_num` value specified in the ICT file.	`0`
	`rmsMetalLineNumberSignal=`*`number`*	Specifies the value of power on the metal lines for RMS EM analysis of signal nets. If this option is not specified, the default value of `0` is considered, which means the software uses the `em_ac_rms_metal_line_num` value specified in the ICT file. However, if this option is specified, it overrides the `em_ac_rms_metal_line_num` value specified in the ICT file.	`0`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`print_em_report`	Defines the EM analysis report file to be generated in the Voltus-Fi-XL flow. `filename`: create report with specified filename `net`: create report for specified nets - all power nets (`all_power`), all signal nets (`all_signal`), all nets (`all_nets`), or net (`netname`). `threshold` - create report for resistors or nodes that have a threshold value of EM ratio above the specified value `type` - create report for specified analysis type; `javg, jmax, jabsavg, jacpeak, jacrms, rc, rcavg, rcrms, tc, tcavg,` or `tcrms.` `rule` - specifies the custom EM rule for which the report should be generated	
	`extendedreport=true | false`	Specifies that the following information should be included in the EM analysis report file: ■ `Td`: time duration in micro second or total 'On Time' period ■ `dutyR`: duty ratio ■ I_{limit}: current limit	`false`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`extendedRMSReport=true | false`	When set to `true`, the following parameters are reported in the RMS EM report: ■ `rmsT`: Delta temperature value ■ `Lv2v`: Distance between terminal vias ■ `rmsMetalLineNumRelaxFactor`: specifies the rating factor corresponding to the number of power on metal lines In addition, the following is added in the header section of the report: ■ `rmsMetalLineNumber`: Specifies the value of power on the metal lines for RMS EM analysis	`false`
	`reportUnmatchedRuleResistor=true | false`	When set to `true`, specifies that while generating the EM analysis reports, the resistors that do not match any EM rule for the specified analysis in the ICT file should be reported in the `"#.rpt_unmatch"` file.	`true`
	`reportAllResistor=true | false`	When set to `true`, reports all resistors of the specified nets in the `"#.rpt_all"` file. **Note:** If both, `reportUnmatchedRuleResistor` and `reportAllResistor`, are set to `true`, the `reportAllResistor` setting gets priority.	`false`
	`consolidatedReport=true`	Specifies that consolidated EM and IR reports for multiple bin files will be generated in Voltus-Fi-XL.	`false`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	`worstResult=true`	Specifies that the worst result will be reported in the consolidated EM and IR reports. This option is enabled by default when the `consolidatedReport` parameter is set to `true`.
		For example,
		`consolidatedReport=true`
		`worstResult=true`
		Note: To view all the results in the consolidated report, set this option to `false`.
			`true`

	`layer_filter=[layer1 layer2...]`	Specifies the layers that are to be filtered out from the EMIR analysis reports. For example, `emirutil layer_filter=[contodcont]` **Note:** If you have specified this option and also set the `reportAllResistor` option to `true`, the resistor information for the specified layers will be filtered out from the reports.	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	`violation_columns=[`*`column_names`*`]`	Specifies the columns to be displayed in the Violation Browser. You can specify multiple columns separated by a space from the following list of columns:
		`LENGTH_COLUMN`
		`RESISTANCE_COLUMN`
		`VIA_AREA__COLUMN`
		`NEEDED_NUM_VIA_COLUMN`
		`NET_COLUMN`
		`WIDTH_COLUMN`
		`MIN_WIDTH_COLUMN`
		`DENSITY_COLUMN`
		`CURRENT_LIMIT_COLUMN`
		`CURRENT_COLUMN`
		`PEAK_TIME_COLUMN`
		`TAP_NAME_COLUMN`
		For example,
		`emirutil violation_columns=[LENGTH_COLUMN RESISTANCE_COLUMN]`
			`All columns are displayed`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`disableLayoutScale=true`	When set to `true`, disables the layout scale factor defined in the ICT file for Voltus-Fi-XL EM analysis. The layout_scale factor is the ratio of the scaled dimensions divided by the drawn dimensions. The ICT file is defined based on the scaled dimensions. Quantus applies the layout_scale factor to the input design during extraction so that the design dimensions and the ICT file dimensions are consistent.	`false`
	`simpleSingleViaDefinition=true | false`	The keyword, `single` is used to differentiate between a single square cut via and square via array. It is applicable to square via, with width equal to '`vc_width/min_width`', only. To support the `single` keyword for all via sizes, set the `simpleSingleViaDefinition` parameter to `true`.	`false`
	`cdf_percentage=value`	The variable holds the value which is mapped to the `cdf_percentage` keyword in the EM rules specified in the ICT file or the technology file. The value of the `cdf_percentage` should be between 0 and 100. For example, `emirutil cdf_percentage="10.0"`	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`device_temp_file=`*`filename`*	Specifies the file that includes information about resistor names and their corresponding temperature values. The EM analysis for these resistors is performed at their corresponding temperatures. For example, `emirutil device_temp_file=resistor_temp.txt`	`none`
	`toggleNetFrequency="(`*`net1, frequency1`*`) (`*`net2, frequency2`*`) (`*`net#, frequency#`*`)"`	Specifies the frequency of the nets so that the tool can derive the `Td` (total on-time period) for multiple clock cycles. You can specify frequencies for different nets as shown below. Wild cards are also supported. For example, `emirutil toggleNetFrequency="(CLK*,2.3e9) (NET1,4.6e9)"`	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	`toggleNetFrequencyFile=`*`filename`*	Specifies the name of the file that contains the net names and their corresponding frequency values. This file is used to specify the frequency of nets to compute the `Td` (total on-time period) for multiple clock cycles. This file contains net names separated by commas and the frequency values.
		
		For example,
		
		`emirutil toggleNetFrequencyFile=freq_file`
		
		A sample `freq_file` is shown below.
		`VDD, 2e08`
		`D*, 4e08`
		`GSS, 3e08`
			`none`

	`multiplePeakPercentThreshold=`*`value`*	Specifies the threshold percentage for the peak current above which the `Td` (total on-time period) for a particular peak is added to the final `Td`. For more information, see Types of EM Analysis. **Note:** This variable is applicable only when the `dynamicACPeak` analysis is set to `multiPeak`. For example, `emirutil multiplePeakPercentThreshold=7`	`5`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`deltaT=`*`temp_value`*	Specifies the maximum rise in temperature in degree Celsius, caused due to Joule heating. It must be a positive value. If the rise in temperature is not specified, Voltus-Fi-XL considers a default temperature of 5 degrees Celsius. This is an ideal condition wherein the design must be EM free. **Example**: `emirutil deltaT=4.0` This option can also be specified in the Variables form and in the command file by using the `set_variable` command.	`5.0 degree Celsius`
	`deltaTLayers=[(layer=`*`layer_name`* `deltaT=`*`temp_value`*`) (layer=`*`layer_name`* `deltaT=`*`temp_value`*`)]`	Specifies layer-specific deltaT values. For layers that do not have deltaT value specified for them, global deltaT value will be used. **Example**: `emirutil deltaTLayers=[(layer=m1 deltaT=5) (layer=m2 deltaT=25)]` In the above example, the `deltaT` value for `m1` layer is `5` degree Celsius while that for `m2` layer is `25` degree Celsius. This option can also be specified in the Variables form and in the command file by using the `set_variable` command.	`none`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	`resistor_filter=[`*`resistor_names`*`]`	Filters the specified resistors in the plot. You can specify multiple resistors separated by a space. These resistors will not be displayed in the plot. Wildcards are supported.
		
		Example:
		
		`emirutil resistor_filter=[rx ry]`
		
		or
		
		`emirutil resistor_filter=[rx*]`
			`All resistors are displayed`

	`exclusion_resistor_filter=[`*`resistor_names`*`]`	Filters all resistors in the plot except those that are specified. This option is used to view plots for specific resistors. Wildcards are supported.
		
		Example:
		
		`emirutil exclusion_resistor_filter=[rx ry]`
		
		or
		
		`emirutil exclusion_resistor_filter=[rx*]`
			`All resistors are displayed`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`removeBulkShorting=true | false`	When set to `true`, this option removes bulk-shorting in the design. Bulk-shorting in the design results in an increase in the EM analysis runtime. This variable is only used as a troubleshooting option. Example: `emirutil removeBulkShorting=true`	`false`
	`emVariableValues=[(variable=configA value=#) (variable=configB value=#)]`	Specifies the EM variable values for current density limit calculation in the EMIR configuration file. Example: `emirutil emVariableValues=[(variable=configA value=20) (variable=configB value=10)]` Where the values of configA and configB are 20 and 10, respectively. **Note:** The EM variable values passed through the configuration file will get priority over the value of same variable specified in the techfile.	`none`

© 2020
All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`enableContextAwareSrmsRelaxation=true | false`	When set to `true`, this option enables the context-aware RMS Relaxation flow. Example: `emirutil enableContextAwareSrmsRelaxation=true` **Note:** This flow is foundry-specific. For more information, contact your Cadence representative.	`false`
`time`	`window=[start1 stop1 start2 stop2 ...]`	Time window to which the EMIR analysis is applied. Multiple none-overlapping windows are supported.	`[0 tend]`
`spf`	`aliasterm="alias library_terminal name"`	Alias used when DSPF instance terminal names do not match the cell terminal name in the libraries.	`none`
`spgs`	`net=[net1 net2 …]`	Enables static power grid solver, and defines power nets to be analyzed.	`none`
	`pwrgate=[powernet1 powernet2 …]`	Defines power gates to be analyzed in static power grid solver.	`none`
	`rshort=value`	Resistors with R<rshort are shorted during static power grid analysis.	`0`
	`tap2sub net=<net_name> include=[<tap_name>] exclude=[<tap_name>]`	Converts tap to subnode.	`none`
	`tap2pin net=<net_name> include=[<tap_name>] exclude=[<tap_name>]`	Converts tap to pin.	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`sub2tap net=<net_name> include=[<node_name>] exclude=[<node_name>]`	Converts subnode to tap.	`none`
	`sub2pin net=<net_name> include=[<node_name>] exclude=[<node_name>]`	Converts subnode to pin.	`none`
	`pin2sub net=<net_name> include=[<pin_name>] exclude=[<pin_name>]`	Converts pin to subnode.	`none`
	`pin2tap net=<net_name> include=[<pin_name>] exclude=[<pin_name>]`	Converts pin to tap.	`none`

Important

In earlier releases, Voltus-Fi-XL 12.1 ISR 10 and before, the `geounit*` option was specified as follows:
`geounit* = 1e6` – when values in DSPF were in meter
`geounit* = 1` – when values in DSPF were in um

From 12.1 ISR 11 release onwards, the behavior of the geounit* option is modified as follows to synchronize it with emirutil:
`geounit* = 1` – when values in DSPF are in meter
`geounit* = 1e-6` – when values in DSPF are in um

© 2020 All Rights Reserved.

Note: Every `net` statement only supports one name and one analysis definition. If different analyses are required, use multiple `net` statements.

The following example performs the EMIR analysis on the `X1` instance. EM and IR analyses are performed on net `VDD` and `VSS` to measure `vmax` and `iavg`. EM analysis is performed on all other nets to calculate the `irms` value. The Current Density limits for the EM report are read from file `./em_dir/qrcTechFile.txt`. In this release, only one instance is allowed for EMIR analysis per Spectre run.

Example EMIR control file (emir.config)

```
net name=[X0.VDD X0.VSS] analysis=[vmax vavg]
net name=[X0.*] analysis=[iavg imax] emirutil techfile = qrcTechfile
layermapfile = contactmapfile


emirutil deltaT=10.0
emirutil dbu=200
emirutil dynamicACPeak=false
emirutil applyRThreshold=0.05
emirutil rmsMultiplierPin=1.0
emirutil rmsMultiplierSignal=1.0
emirutil useEncryptedWidth=true
net design_res_models = [name=resTwoTerm   l=lr   w=wr   layer=mt2 unit=1e6]
solver method=[iterated]
```

Note: Multiple net names must be specified in different lines in the control file to avoid any errors. This is shown in the above example.

For details on how to specify variables in the EMIR control file, see the "Variables" chapter.

© 2020 All Rights Reserved.

Outputs of EMIR Analysis

As output, Voltus-Fi-XL produces the following:

- Power-rail and signal IR drop and EM analysis result plots displayed on the Virtuoso layout
- Power-rail and signal IR drop and EM analysis result reports
- Structural analysis results displayed on the Virtuoso layout
- Power-grid views

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

5

EM Rules Specification

- EM Rule File Requirements in Voltus-Fi-XL
- Parameters for Limit-Based EM Analysis in Voltus-Fi-XL
- Supported Current Density Keywords in Voltus-Fi-XL
- EM Model Syntax
- Rules for Defining EM Parameters in Voltus-Fi-XL
- EM Rule Selection Priority
- Process
- EM Data File in Voltus-Fi-XL

© 2020 All Rights Reserved.

EM Rule File Requirements in Voltus-Fi-XL

Before running EMIR analysis in Voltus-Fi-XL, you need to specify the EM reliability rules. EM rules are defined in the technology files. The Voltus-Fi - Spectre EMIR solution supports the following technology file formats:

- The Quantus technology file (`qrcTechFile`)
- The ICT, ICT-EM (EM Rules only) file
- The EM data file (`emDataFile`).

Important

The `emDataFile` support is provided only for backward compatibility. It is recommended that you convert your `emDataFile` into an ICT file to specify the EM rules for performing EM analysis. For details of how to convert the `emDataFile` into an ICT file, see .

The ICT file contains the process information for the design and information about the EM rules. This is a text-based file that can be edited using any text editor. You can either specify the ICT file directly as an input for EM analysis or use the information in the ICT file to update the `qrcTechFile`.

Related Topics

Converting the EM Data File to ICT File, Parameters for Limit-Based EM Analysis in Voltus-Fi-XL, Supported Current Density Keywords in Voltus-Fi-XL

© 2020 All Rights Reserved.

Parameters for Limit-Based EM Analysis in Voltus-Fi-XL

Voltus-Fi-XL supports parameters for only limit-based EM analysis. The following parameters can be used to specify EM rules in the ICT EM file. The EM rules might also be embedded in the ICT file or the `qrcTechFile`.

Parameter Name	Description
`em_vcwidth` <*value*>	Optional parameter for `via` layers that defines the minimum size via that is used by the software to identify single square via.
`em_W_n`	Optional parameter that specifies the wire width in microns below which a line is considered narrow. All lines are considered wide unless you specify otherwise using this parameter **Note:** It helps to select the rule between `em_jmax_*_w` and `em_jmax_*_n`. If `w < em_W_n` , then `em_jmax_*_n` is selected, otherwise `em_jmax_*_w` selected.
`em_jmax_dc_rms_metal_line_factor`	Defines the relationship between the value of power on metal lines and the `Irms` rating values for DC RMS analysis.
`em_jmax_ac_rms_metal_line_factor`	Defines the relationship between the value of power on metal lines and the `Irms` rating values for AC RMS analysis.
`em_ac_rms_context_aware_pitch_num`	Specifies the pitch count values for RMS relaxation for AC RMS analysis in the context-aware RMS Relaxation flow. For example, `em_ac_rms_context_aware_pitch_num 5` **Note:** The context-aware RMS Relaxation flow is foundry-specific. For more information, contact your Cadence representative.
`em_dc_rms_context_aware_pitch_num`	Specifies the pitch count values for RMS relaxation for DC RMS analysis in the context-aware RMS Relaxation flow. For example, `em_dc_rms_context_aware_pitch_num 5` **Note:** The context-aware RMS Relaxation flow is foundry-specific. For more information, contact your Cadence representative.

© 2020 All Rights Reserved.

`em_jmax_ac_rms_context_aware_factor`	Specifies the benefit factor for RMS relaxation for AC RMS analysis in the context-aware RMS Relaxation flow. For example, `em_jmax_ac_rms_context_aware_factor 1.5 srms_0>= 0.008 srms_1> 0.0004` Where, `srms_0` is the worst `srms` value in the neighboring area within the context-aware region of the parasitic. `srms_1` is the second worst `srms` value in the neighboring area within the context-aware region of the parasitic. **Note:** The context-aware RMS Relaxation flow is foundry-specific. For more information, contact your Cadence representative.
`em_jmax_dc_rms_context_aware_factor`	Specifies the benefit factor for RMS relaxation for DC RMS analysis in the context-aware RMS Relaxation flow. For example, `em_jmax_dc_rms_context_aware_factor 1.5 srms_0>= 0.008 srms_1> 0.0004` Where, `srms_0` is the worst `srms` value in the neighboring area within the context-aware region of the parasitic. `srms_1` is the second worst `srms` value in the neighboring area within the context-aware region of the parasitic. **Note:** The context-aware RMS Relaxation flow is foundry-specific. For more information, contact your Cadence representative.

© 2020 All Rights Reserved.

`em_segment_length`	A process definition parameter that determines how to calculate the length of a wire segment. This parameter determines whether the segment length is of the type perimeter, centerline, or total. The difference between the different length types is as follows: ■ `perimeter`: specifies that the length calculation is along the wire's outer edge boundary. This value is used for 28nm process nodes. ■ `centerline`: specifies that the length is extended to the edge of the segment along the centerline path. This value is used for 20nm process nodes. ■ `total`: specifies that the segment length is calculated by adding the path lengths of all the resistors in the segment.
`em_recover`	Optional parameter that specifies the recovery factor used for calculating the average current density with recovery. You can set the recovery factor to -1.0 when you want to account for the average of the absolute current (avg(abs(current))) so that the negative current does not cancel the positive current values in arriving at the average current result. *Default*: 1 *Range*: A positive integer or floating point number

Related Topics

Supported Current Density Keywords in Voltus-Fi-XL, EM Rule File Requirements in Voltus-Fi-XL

Supported Current Density Keywords in Voltus-Fi-XL

The table below lists the Current Density (JMAX) keywords that can be specified in the ICT EM file.

Keyword	Description
`em_jmax_dc_avg*`	Optional keyword that specifies the DC AVG analysis in Voltus-Fi-XL
`em_jmax_ac_avg*`	Optional keyword that specifies the AC AVG analysis in Voltus-Fi-XL

© 2020 All Rights Reserved.

`em_jmax_dc_peak*`	Optional keyword that specifies DC PEAK analysis in Voltus-Fi-XL
`em_jmax_ac_peak*`	Optional keyword that specifies AC_PEAK analysis in Voltus-Fi-XL
`em_jmax_dc_rms*`	Optional keyword that specifies DC RMS analysis in Voltus-Fi-XL
`em_jmax_ac_rms*`	Optional keyword that specifies AC RMS analysis in Voltus-Fi-XL
`em_jmax_ac_rec`	Optional keyword that specifies the maximum recovery AC current analysis in Voltus-Fi-XL.

Where star '*' implies that '`_w`' and '`_n`' rules are also supported.

For example,

- `em_jmax_ac_peak_w` is an optional parameter that is essentially the same as `em_jmax_ac_peak`, with the exception that it is specified for wide wires (as defined by `em_W_n`).
- `em_jmax_ac_peak_n` is an optional parameter that is essentially the same as `em_jmax_ac_peak`, with the exception that it is specified for narrow wires (as defined by `em_W_n`).

Related Topics

EM Rule File Requirements in Voltus-Fi-XL, Parameters for Limit-Based EM Analysis in Voltus-Fi-XL

EM Model Syntax

```
em_model {
# Width Independent parameters are not affected by width or area
[em_vcwidth <value>]
[em_W_n <value>]
[em_recover <value>]
# Parameters applying specifically to narrow or wide wires
[em_jmax_ac_avg_n <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_ac_avg_w <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_ac_peak_n <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_ac_peak_w <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
```


© 2020 All Rights Reserved.

```
[em_jmax_ac_rms_n <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_ac_rms_w <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_ac_rec_n <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_ac_rec_w <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_dc_avg_n <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_dc_avg_w <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_dc_peak_n <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_dc_peak_w <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_dc_rms_n <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[em_jmax_dc_rms_w <value>]
   [jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
# Width dependent parameters for a conductor or diffusion layer
[em_jmax_ac_avg <value> | <value_1> <width_1> [...] | EQU <fn(E)> ]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[via_range <value>]
[em_jmax_ac_peak <value> | <value_1> <width_1> [...] | EQU <fn(E)> ]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[via_range <value>]
[em_jmax_ac_rms <value> | <value_1> <width_1> [...] | EQU <fn(E)> ]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[via_range <value>]
[em_jmax_ac_rec <value> | <value_1> <width_1> [...] | EQU <fn(E)> ]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[via_range <value>]
[em_jmax_dc_avg <value> | <value_1> <width_1> [...] | EQU <fn(E)> ]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>....]
[via_range <value>] [L < | <= | > | >= <value in microns>]
```


© 2020 All Rights Reserved.

```
   [W < | <= | > | >= <value in microns>]
[em_jmax_dc_peak <value> | <value_1> <width_1> [...] | EQU <fn(E)> ]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[via_range <value>] [L < | <= | > | >= <value in microns>]
   [W < | <= | > | >= <value in microns>]
[em_jmax_dc_rms <value> | <value_1> <width_1> [...] | EQU <fn(E)> ]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>....]
[via_range <value>] [L < | <= | > | >= <value in microns>]
   [W < | <= | > | >= <value in microns>]
# Area dependent parameters for a via layer
[em_jmax_ac_avg <value> | <value_1> <area_1> [...] | EQU <fn(E)>]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[current_direction up|down|both] [via_range <value>]
[em_jmax_ac_peak <value> | <value_1> <area_1> [...] | EQU <fn(E)>]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[current_direction up|down|both] [via_range <value>]
[em_jmax_ac_rms <value> | <value_1> <area_1> [...] | EQU <fn(E)>]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[current_direction up|down|both] [via_range <value>]
[em_jmax_ac_rec <value> | <value_1> <area_1> [...] | EQU <fn(E)>]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.....]
[current_direction up|down|both] [via_range <value>]
[em_jmax_dc_avg <value> | <value_1> <area_1> [...] | EQU <fn(E)>]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>....]
[current_direction up|down|both] [via_range <value>]
[L < | <= | > | >= <value in microns>]
   [W < | <= | > | >= <value in microns>]
[em_jmax_dc_peak <value> | <value_1> <area_1> [...] | EQU <fn(E)>]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>....]
[current_direction up|down|both] [via_range <value>]
[L < | <= | > | >= <value in microns>]
   [W < | <= | > | >= <value in microns>]
[em_jmax_dc_rms <value> | <value_1> <area_1> [...] | EQU <fn(E)>]
```


© 2020 All Rights Reserved.

```
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.......]
[current_direction up|down|both] [via_range <value>]
[L < | <= | > | >= <value in microns>]
   [W < | <= | > | >= <value in microns>]
}
```


© 2020 All Rights Reserved.

Rules for Defining EM Parameters in Voltus-Fi-XL

An example of the EM rule for defining the `em_jmax_dc_avg` parameter is provided below:

```
[em_jmax_dc_avg <value> | <value_1> <area_1/width_1> [...] | EQU <fn(E)>]
[jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]]
[jmax_life <life1> <scale1> [<life2> <scale2>.......]]
[current_direction up | down | both]
[via_range <value>]
[conditions]
[single]
[power_rail/power_grid]
[priority==<priority no.>]
[device==”<device model names>”]
[sub_conductor=="<subconductor names>"]
[color=="<list of color no.>"]
[mask==<mask no.>]
```

All other EM parameters are defined with the same structure but apply to different characteristics of the `em_model` (peak current versus average for instance). The following rules apply to all the limit-based `em_model` parameters:

- **Single Value Current Density**

  Specifying a single value or an equation is an enhancement for width or area dependent parameters (such as `em_jmax_ac_avg`).

  Note: For narrow/wide wire parameters (such as `em_jmax_ac_avg_n` or `em_jmax_ac_avg_w`) you can only specify a single value. It will not accept either a PWL table or an equation.

- **Piece-wise Linear (PWL) Interpolation**

  In a width-dependent model for a conductor layer, or for an area-dependent model for a via layer, PWL interpolation uses a series of data points that are points on a graph of X versus Y, where X can be either a value or an expression and Y is a value. For example, current density value and area `<value1> <area1>`, current density value and width `<value1> <width1>`, current density expression and area `<expression1> <area1>`, or current density expression and width `<expression1> <width1>`.

  The following syntax are supported:

  ```
  em_jmax_dc_avg PWL <value1> <area1/width1> <value2> <area1/width1>
  em_jmax_dc_avg PWL (<value1> <area1/width1> <value2> <area1/width1>)
  em_jmax_dc_avg PWL (<expression1> <area1/width1> <expression2> <
  area1/width1>)
  ```


© 2020 All Rights Reserved.

For example:

```
em_jmax_dc_avg PWL 0.3582 0.001444 0.5694 0.003364 0.7164 0.0038
em_jmax_dc_avg PWL (0.3582 0.001444 0.5694 0.003364 0.7164 0.0038)
em_jmax_dc_avg PWL (0.05*0.5 0.002 0.1*0.5 0.004 0.2*0.5 0.008)
```

Equations in EM Model

The EQU keyword denotes the definition of an equation for current density.

An EOL or one of the qualifiers (`jmax_factor`, `current_direction`, or `single`) marks the end of an equation specifying the Current Density limit. If the equation resolves into two separate equations, then this will result in an error.

For EQU, the following operators (not case-sensitive) are supported:

- `SQRT`
- `LOG`
- `EXP`
- `* / + -`
- `()`
- `e`
- `^`
- `expr1? expr2: expr3`

Where, `expr1` is a conditional expression. If `expr1` evaluates to `true`, then `expr2` is evaluated, else `expr3` is evaluated.

For example:

```
Life_factor= ((A < 5 and B > 2)? 3:4)

em_jmax_dc_avg EQU 3.0*w * Life_factor jmax_factor 50 1.1 110 1.0
125 0.92 L > 5
```

Note: Square brackets '`[]`' are not supported. A comma ',' implies multiple equations, and results in an error. The period '.' is ambiguous and should not be used to indicate multiplication (use * instead).

Spaces are permitted to make an equation more readable.

Variables within equations will be populated by the analysis tool at run-time. Use the `em_variables` parameter in the process definition to globally declare variables that are used within the various `em_models`. List of supported variables are provided below:

© 2020 All Rights Reserved.

- `a:` area of the resistor
- `w:` width
- `l:` length
- `La:` length of pseudo via
- `La_x:` length of pseudo via along the x-axis
- `La_y`: length of pseudo via along the y-axis
- `N:` number of vias
- `delta_T`: rise in temperature in degree Celsius
- `r:` duty ratio
- `Tref`: refers to `em_tref`, the nominal temperature for EM analysis defined in the process section of the ICT file. If `em_tref` is not defined, then default value(110C) is set.
- `cdf_percentage`: This variable lets you use the value specified using the keyword `cdf_percentage` in the emir config file or command file, in the EM rules. For example,

```
conductor "m1" {
    min_spacing         0.3
    min_width           0.3
    height              1.05
    thickness           0.53
    resistivity         0.1
    gate_forming_layer  false
    thermal_conductivity 0.39
    em_model {
      cxyz=exp(normsinv(cdf_percentage*10.0))
      em_jmax_dc_avg  EQU 2*w*cxyz L >  5
      em_jmax_dc_avg  EQU 3*w*cxyz L <= 5
    }
}
```

- `Tlife`: This variable provides the value of lifetime at which EM analysis is performed. This value can be specified in hours or years.

  The value of lifetime can be specified in the config file as follows:

```
emirutil Tlife = "20000 hours"
```

  It can also be specified in techfile in the process block section as follows:

© 2020 All Rights Reserved.

```
em_lifetime 20000
```

Note: The lifetime value specified in the techfile gets priority over the value specified in the config file. If the value is not specified in either of these files, the default value of "`10 years`" is used.

- `jmax_factor_multiplier`: This variable provides the value of jmax_factor for the resistor.
- `jmax_life_multiplier`: This variable provides the value of jmax_life for the resistor.

  For example:

```
derate = jmax_factor_multiplier * jmax_life_multiplier
em_jmax_dc_avg  EQU 3.0*derate   jmax_factor   50 1.1  110  1.0  125 0.92
L >  5
```

  The equation definition ends when reaching the end of line (EOL) or another keyword (`jmax_factor, current_direction`, or `single`).

The following are supported for equations in the ICT EM file:

- **Support for min max functions**: The min and max functions accepts two arguments:

```
min(a,b) => It returns the smallest value of a and b
max(a,b) => It returns the largest value of a and b
```

  Nested min and max functions are also supported as shown below:

```
min(a, max(b, c))
max(max(a, b), c)
min(min(a, b), max(c, d))
max(min(a, b), max(c, d))
```

- **Support for defining the normsinv function**: The `normsinv` function accepts a single argument. It returns the value by taking the inverse normal distribution of the fraction `x`. The value of `x` must always be between 0 to 1.

```
normsinv(x)
```

- **Support for defining variables:** Variables can be defined to be used in equations specifying the current density limit. An example of how these are specified in the ICT file is shown below. In the below equation, s1, s2, s3 and s4 are variables.

```
em_model {
    s1 = 2.0*w
    s2 = s1*1.0
    s3 = 3.0*w
    s4 = s3
```


© 2020 All Rights Reserved.

```
    em_jmax_dc_avg EQU s2*4 jmax_factor 50 1.1 110 1.0 125 0.92 L> 5
    em_jmax_dc_avg EQU s4*2 jmax_factor 50 1.1 110 1.0 125 0.92 L<=5
}
```

Note: A variable needs to be defined before it is used.

- **Support for keyword "cond" and logical operator "not"**: The keyword "cond" can be used to group multiple conditions logically for EM rules. for example:

```
em_jmax_dc_avg EQU 0.1*w     cond (L <=0.024 and L>=0.026)
em_jmax_dc_avg EQU 0.2*w     cond (((L == 0.020) or (L == 0.030))and (W <
0.01))
em_jmax_dc_avg EQU 0.2*w     cond ( not(L == 0.020 or L == 0.030))
```

- **Support for jmax_factor_dc_avg and jmax_factor_ac_avg**: These are specified for defining the jmax_factor information at the layer level.

  jmax_factor_dc_avg: specifies the jmax_factor information for the DC avg EM rules at the layer level.

  jmax_factor_ac_avg: specifies the jmax_factor information for the AC avg EM rules at the layer level.

  For example:

```
conductor "M1" {
   em_model {
          jmax_factor_dc_avg 85 3 90 2 100 1 110 0.6 120 0.2 125 0.1
          jmax_factor_ac_avg 85 3 90 2 100 1 110 0.6 120 0.2 125 0.1

          em_jmax_dc_avg EQU 2 * w     L < 0.5
          em_jmax_ac_avg EQU 2 * w     L < 0.5
          em_jmax_dc_avg EQU 1.5 * w   L >= 0.5
          em_jmax_ac_avg EQU 1.5 * w   L >= 0.5
    }
}
```

  Note: For EM rules that have jmax_factor construct, jmax_factor will be calculated from the rule jmax_factor information instead of the layer-level construct jmax_factor_dc_avg or jmax_factor_ac_avg.

- **Support for jmax_life_dc_avg** and **jmax_life_ac_avg**: These are specified for defining the jmax_life information at the layer level.

  jmax_life_dc_avg: specifies the jmax_life information for the DC avg EM rules at the layer level.

© 2020 All Rights Reserved.

jmax_life_ac_avg: specifies the jmax_life information for the AC avg EM rules at the layer level.

When the lifetime has an exact match in the PWL, jmax_life gets the corresponding value from the PWL. If not, then the expression in the below syntax is evaluated to get the value of jmax_life.

```
jmax_life_dc_avg <life1> <scale1> <life2> <scale2> <life3> <scale3>
jmax_life_equ <expression>
```

```
jmax_life_ac_avg <life1> <scale1> <life2> <scale2> <life3> <scale3>
jmax_life_equ <expression>
```

For example:

```
jmax_life_dc_avg PWL 43800 1.8 52560 1.5  61320 1.3 70080 1.2  78840 1.1
87600 1 131400 0.6 175200 0.5  jmax_life_equ  (10/Tlife)^0.9


jmax_life_ac_avg 43800 1.8 52560 1.5  61320 1.3 70080 1.2  78840 1.1  87600
1 131400 0.6 175200 0.5  jmax_life_equ  (10/Tlife)^0.9
```

Note: For EM rules that have jmax_life construct, jmax_life will be calculated from the rule jmax_life information instead of the layer-level construct jmax_life_dc_avg or jmax_life_ac_avg.

Order Dependency

The actual Current Density limit (value, piece-wise linear or PWL pairs, or equation) should follow immediately after the Current Density keyword (`em_jmax_dc_avg`). Any of the qualifiers (`jmax_factor`, `current_direction`, or `single`) will come after that. The order of the qualifiers is not important.

Units of Values

The unit should be specified in accordance with the setting defined by the `em_conductor_unit`, `em_via_unit`, and `em_via_area_unit` parameters defined in the process definition (see Process). Units are specified as:

- For a single value or an equation:
  - `A/cm`^{2} or `mA` for metal layers (width based)
  - `A/via` or `mA` for via layers (per via)
- For a PWL table:
  - `A/cm`^{2} or `mA`, with width specified in microns for metal layers
  - `A/cm`^{2} or `mA`, with area specified in square microns for via layers

JMAX Factor

```
jmax_factor <temp1> <scale1> [<temp2> <scale2> ...]
```


© 2020 All Rights Reserved.

`jmax_factor` is the optional scaling factor to use at different temperatures compared to the reference temperature (defined by `em_tref` in the process definition). The temperature for `jmax_factor` should be specified in degrees Celsius.

Scaling factor is a positive integer: `>1 to scale up, <1 to scale down, or 1 for no scale effect`.

If the specified temperature, `T`, falls between the defined minimum and maximum temperatures, the software will calculate a new scale factor (RT) using the formula, `TTF=A*J^(-n)*exp(Ea/kT)`.

For example:

```
RT = RT1*exp((Ea/kT)*(1/T-1/T1))
```

where

```
Ea/kT=(T1*T2/(T1-T2))*ln(RT2/RT1)
```

Note:

- The temperatures T, T1, and T2 mentioned in the above formula are in Kelvin.
- If the specified temperature, `T`, is either below or above the defined minimum and maximum temperatures, the software will use the minimum and maximum temperature for calculating the scale factor, respectively.
- When setting scale factors for multiple temperatures, they should be specified in an ascending sequence to enable interpolation.

Format of the Conditions

Condition = `[L/W/Lu/Wu/Lb/Wb/Lv2v/Sv/Td/r/a/N < | <= | > | >= | == | != <value in microns/micro seconds> | ground_net | supply_net | Vu_current_direction {up | down}] | [Itolerance < | <= | > | >= | == | != <value in percentage> ]`

Conditions = `[condition]*`

The `L` parameter specifies the length-based jmax values in microns.

The `W` parameter specifies the width-based jmax values in microns.

`L` and `W` are applicable for both metals and vias.

`Lu, Lb, Wu,` and `Wb` are applicable for vias only.

`Lv2v` and `Itolerance` are applicable only for the `power_rail` rule.

Where,

© 2020 All Rights Reserved.

`L` = length
`W` = Width
`Lu` = Upper metal length
`Lb` = Bottom metal length
`Wu` = Upper metal width
`Wb` = Lower metal width
`Lv2v` = Distance between terminal vias
`Sv` = Distance between the two top vias enclosing a metal/via. For metals, it indicates the distance between the immediate top vias across a metal. For via, it indicates the distance between the immediate top vias across a particular via.
`Vu_current_direction {up | down}` = Current direction of the top vias across a metal/via. For metal, it indicates the current direction of the immediate top vias across a metal layer. For via, it indicates the current direction of the immediate top vias across a particular via.
`supply_net` = specifies that the EM rule is applicable only to the supply net
`ground_net` = specifies that the EM rule is applicable only to the ground net
`Itolerance` = Difference in the current values of terminal vias. For example, if a difference of up to 5 percent is acceptable, then specify '`Itolerance <= 0.05`'.
`Td` = Time duration in micro second or total ‘On Time” period
`r` = Duty ratio
`a` = Area of the single cut in the via or in the via array
`N` = Number of vias in the via array

Voltus-Fi-XL supports metal length/width rules above and below vias, where, the dimensions to be checked for the rules are different for the metal above and the metal below. In this case, `Lu/Wu` refers to dimensions of the metal above the via and `Lb/Wb` refers to dimensions of the metal below the via.

■ Single

The keyword, `single` can be specified to differentiate between a single square cut via and square viaarray. The following example shows how to use the `single` keyword:

```
em_jmax_dc_avg PWL 0.022  0.01  jmax_factor 105 1.1  110 1.0 115
0.9 120 0.8 125 0.7 130 0.6 140 0.5 150 0.4  current_direction up
Lb > 4 Wb >= 0.05  current_direction up single
```

■ JMAX Life

The keyword, `jmax_life` provides the ability to set the scaling factor that applies to the Current Density limits for different lifetimes. The syntax is as follows:

```
jmax_life <life1> <scale1> [<life2> <scale2>.......]
```

Note: The software will take the unit of lifetime from the `em_lifetime_units` parameter specified in the process section of the ICT file. In case the specified `life` value does not match any of the `life` values provided using `jmax_life`, the software

© 2020 All Rights Reserved.

performs linear interpolation to calculate the `Tlife` factor.

- **rating_factor**

The keyword, `rating_factor` specifies the temperature derating factor for different lifetimes using the rating factor table. The table is used along with the global lifetime factor table that lists the lifetime variables for different lifetimes. These variables can be used in the EM equation. For example,

```
em_model {
        GLOBAL_LIFETIME_FACTOR  =   (
        ( LIFETIME    HOURS   LF0 LF1 )
        ( 3_YEAR      26280   5.0 6.0 )
        ( 4_YEAR      35040   7.0 8.0 )
        ( DEFAULT    100000   9.0 10.0 )
        )
        RATING_FACTOR   =   (
        ( LIFETIME    100 105 110 115 120 125 130 140 150 )
        ( 3_YEAR  2.0 1.7 1.3 1.0 0.9 0.8 0.7 0.6 0.5 )
        ( 4_YEAR  2.0 1.6 1.4 1.2 1.0 0.4 0.3 0.2 0.1 )
        ( DEFAULT 1.8 1.7 1.6 1.4 1.1 0.9 0.5 0.4 0.3 )
        )
           em_jmax_dc_avg  EQU 2.0*w*1.0*LF0   rating_factor L >  5
           em_jmax_dc_avg  EQU 2.0*w*3.0*LF1   rating_factor
  }
```

Where,

The different lifetimes, `3 years`, `4 years`..., are specified in the `rating_factor` table and the lifetime variables for these lifetimes, `LFO`, `LFI`, are specified in the global lifetime factor table.

Note: The rating factor is only applied to EM rules that have the `rating_factor` keyword.

- **Length**

```
[L < | <= | > | >= <value in microns>]
```

The L parameter specifies the length-based jmax values in microns.

- **Width**

```
[W < | <= | > | >= <value in microns>]
```

Similar to the L parameter, the W parameter specifies the width-based jmax values in microns.

© 2020 All Rights Reserved.

Current Direction

`current_direction [up | down | both]` can be specified for via current direction. The syntax of `current_direction` is:

- `up` - means the direction of the current is from bottom to top and specifies to use the length and width of the metal below `Lb/Wb` for the VIA Jmax factor
- `down` - means the direction of the current is from top to bottom and specifies to use the length and width of the metal above Lu/Wu for VIA Jmax factor
- `both` - This rule is applicable only when the direction of the current is uncertain. This rule is unlikely to be applied since the software calculates the current direction

Via Range

```
via_range <value>
```

`via_range` is an optional parameter which can be used with the via layers. The value of `via_range` denotes the number of vias for which the calculated limits are valid.

`via_range` is specified as a positive integer.

Power Rail/Power Grid

The keyword `power_rail` or `power_grid` can be specified to enable the particular EM rule for power-rail analysis. When power-rail analysis is enabled, an EM rule with the `power_rail` or `power_grid` keyword will have a higher priority than any other rule.

```
em_jmax_dc_avg PWL 0.022 0.01 jmax_factor 105 1.1 110 1.0 115
0.9 120 0.8 125 0.7 130 0.6 140 0.5 150 0.4
Lb > 4 Wb >= 0.05 current_direction up
power_rail  Lv2v <= 5 Itolerance <= 0.05
```

ApplyR

The keyword `applyR` can be used to modify the threshold value of ratio “r” in AC-Peak analysis. The default value of this parameter is `0.05`. Modifying this parameter is useful in scenarios where the value of “`r`” is very small. A small “`r`” value results in an unreasonable increase in the `Ipeak_ac` limit. You can avoid such scenarios by resetting the value of “r” in the ICT file. For example:

```
em_jmax_ac_peak EQU 4.0 r < 0.01 applyR 0.01
```

Priority

Specifies the priority of a rule. It is an integer number. Currently the only value that is supported is `1`. The syntax is as follows:

```
priority==<integer no.>
```


© 2020 All Rights Reserved.

Note: The rule with `priority==1` is only enabled when the variable '`powerRailRules=n10_special`' is set and when the `power_grid` and `power_rail` keywords are also specified.

- **Device**

  This keyword is used to specify different EM rules for device resistors, which are subsets of other layers. The syntax is as follows:

  ```
  device=="device model names"
  ```

  For example, the following keyword specifies that the particular EM rule is applicable only to device with the model names, "`devRA devRB`".

  ```
  device=="devRA devRB"
  ```

- **Subconductor**

  This keyword is used to specify different EM rules for subconductors, which are subsets of poly or other layers. The syntax is as follows:

  ```
  sub_conductor=="subconductor names"
  ```

  For example, the following keyword specifies that the particular EM rule is applicable only to subconductor layers, `pploy` and `gpoly`.

  ```
  sub_conductor=="ppoly gpoly"
  ```

- **bridge_via**

  This keyword is used to specify that the particular EM rule will apply only to bridge vias.

- **Color**

  This keyword is used to specify that the particular EM rule is applicable only to the resistor with the specified color number.

  For example,

  `color=="2 4 5"` specifies that the particular EM rule is applicable only to the resistor with property, `$M=2 OR $M=4 OR $M=5`.

- **Masks**

  This keyword is used to specify that the particular EM rule is applicable only to the resistor with the specified mask number.

  For example,

  `mask==2` specifies that the particular EM rule is applicable only to the resistor with property, `$M=2`.

© 2020 All Rights Reserved.

■ hi_em

This keyword is used to specify that the particular EM rule is applicable only to the design resistor with the specified value of `hi_em` parameter.

For example,

`hi_em==1` specifies that the particular EM rule is applicable only to the design resistor with the property, `hi_em=1`.

Note: The `Sh` keyword is not supported in the EM rules condition. It is ignored if specified in the EM rules condition inside the techfile.

© 2020 All Rights Reserved.

Rules for Specifying Via Area

You can specify PWL in either one of the following ways:

- Provide PWL for a specific area:

  ```
  em_jmax_* PWL Value_1 area_1 … Value_N areaN
  ```

  For example,

  ```
  em_jmax_* PWL 1.3 2.0 1.7 3.0
  ```

- Provide PWL for a specified area range:

  ```
  em_jmax_* EQU <equation with 'a'> conditions
  ```

  For example,

  ```
  em_jmax_* EQU (1.3 * a)/2.0  a>=0.0 a<2.5
  em_jmax_* EQU (1.7 * a)/3.0  a>=2.5 N>=3
  em_jmax_* EQU (1.9 * a)/3.0  a>=2.5 N<3
  ```


© 2020 All Rights Reserved.

Rules for Specifying the EM Analysis Type for Power and Ground Nets

Set the variable, `splitACDCRules`, to `true` for specifying different rules for EM analysis of power and signal nets. You can set this variable in the following ways:

- `setenv VOLTUSFI_SPLIT_ACDC_RULES true`

  `set_variable splitACDCRules true`
  When this variable is specified using the set_variable command, it takes priority over the setenv variable.

- Set the variable, `splitACDCRules` to `true` in the Variables form. You can open this form from the EM tab of the IR/EM Results form.

The table below shows the rules for power and signal nets both when the variable is set and not set.

Table 5-1 Rules for Power and Signal Nets

splitACDCRules is set to false	**splitACDCRules is set to true**	
Power and Signal Nets	**Power Nets**	**Signal Nets**
■ `em_jmax_dc_peak` rules are used when `MAX` analysis is specified ■ `em_jmax_ac_peak` rules are used when `ACPEAK` analysis is specified **Note:** If `ACPEAK` analysis is specified but the `em_jmax_ac_peak` rules are not specified in the technology file, then `em_jmax_dc_peak` rules are used for the `ACPEAK` analysis.	`em_jmax_dc_peak`	`em_jmax_ac_peak`

© 2020 All Rights Reserved.

splitACDCRules is set to false	splitACDCRules is set to true	
em_jmax_dc_avg	em_jmax_dc_avg	em_jmax_ac_avg
em_jmax_ac_rms	em_jmax_dc_rms	em_jmax_ac_rms

© 2020 All Rights Reserved.

EM Rule Selection Priority

This section details the order of priority in which the EM rules are applied by Voltus-Fi-XL based on the specified keywords. The order of priority of keywords in descending order is provided below:

1. device --> **highest priority**
2. sub_conductor
3. bridge_via
4. priority
5. power_grid/power_rail
6. color
7. mask
8. current_direction
9. Conditions
10. Area matching in case of PWL
11. Base rule
12. Optimistic/pessimistic rules --> **lowest priority**

These are detailed below.

- `device`

  When a device is specified in the EM rule file, the software first looks for the `device` keyword and applies the EM rules to the specified device.

  For example, for a device with model name, `devRA`, if the following rules are specified in the ICT file:

  ```
  1. em_jmax* …   device=="devRA"
  2. em_jmax* …   device=="devRB"
  3. em_jmax* …
  ```

  Then the software follows the order of priority provided below:

  - Match rule 1 because it has a matching `device` keyword
  - Match rule 3 if rule 1 does not match
  - Skip rule 2 because it is for device, `devRB`

© 2020 All Rights Reserved.

- `sub_conductor`

  When a sub conductor is specified in the EM rule file, the software first looks for the `sub_conductor` keyword and applies the EM rules to the specified sub conductor.

  For example, for a resistor on layer, `ppoly`, if the following rules are specified in the ICT file:

  ```
  1. em_jmax* …   sub_conductor=="ppoly"
  2. em_jmax* …   sub_conductor=="gpoly"
  3. em_jmax* …
  ```

  Then the software follows the order of priority provided below:

  - Match rule 1 because it has a matching `sub_conductor` keyword
  - Match rule 3 if rule 1 does not match
  - Skip rule 2 because it is for sub conductor, `gpoly`

- `bridge_via`

  When a `bridge_via` is specified in the EM rule file, the software first looks for the `bridge_via` keyword and applies the EM rules having `bridge_via` keyword for the bridge via. In case, the via is not bridge via, rules having `bridge_via` keyword will be ignored.

  For example, for a bridge via, if the following rules are specified in the ICT file:

  ```
  1. em_jmax* … bridge_via
  2. em_jmax* …
  ```

  Then the software follows the order of priority provided below:

  - Match rule 1 because it is a bridge via
  - Match rule 2 if rule 1 does not match

  In case, the via is not bridge via, then the software will match rule 2.

- `priority`

  When a priority is specified in the EM rule file, in the following manner:

  ```
  1. em_jmax* …   priority==1
  2. em_jmax* …
  ```

  Then the software follows the order of priority provided below:

  - Match rule 1 when `powerRailRules=n10_special` is set
  - Match rule 2 if rule 1 does not match

© 2020 All Rights Reserved.

- `power_rail/power_grid`

  When the `power_rail/power_grid` keyword is specified in the EM rule file, in the following manner:

  ```
  1. em_jmax* …   power_grid
  2. em_jmax* …
  ```

  Then the software follows the order of priority provided below:

  - Match rule 1 when the power grid analysis is enabled
  - Match rule 2 if rule 1 does not match

- `color`

  When the `color` keyword is specified in the EM rule file, in the following manner:

  ```
  1. em_jmax* …   color==”1 3 5”
  2. em_jmax* …
  ```

  For example, if a resistor has color number 5, specified using `$M=5`, then the software follows the order of priority provided below:

  - Match rule 1 for the specified color number
  - Match rule 2 if rule 1 does not match

- `mask`

  When the `mask` keyword is specified in the EM rule file, in the following manner:

  ```
  1. em_jmax* …   mask==2
  2. em_jmax* …
  ```

  For example, if a resistor has mask number 2, specified using `$M=2`, then the software follows the order of priority provided below:

  - Match rule 1 for the specified mask number
  - Match rule 2 if rule 1 does not match

- `current_direction`

  When the `current_direction` keyword is specified in the EM rule file, in the following manner:

  ```
  1. em_jmax* …   current_direction up
  2. em_jmax* …   current_direction down
  3. em_jmax* …
  ```

  Then, for a resistor with `current_direction up`, the software follows the order of priority provided below:

© 2020 All Rights Reserved.

- Match rule 1 because it has a matching `current_direction` keyword
- Match rule 3 if rule 1 does not match
- Skip rule 2 because the current direction is not matching

- Conditions

  When conditions are specified in the EM rule file, in the following manner:

```
1. em_jmax* …   L <=5
2. em_jmax  …   L > 5
3. em_jmax* …
```

  Then the software follows the order of priority provided below:

  - Match rules 1 and 2 because conditions are specified
  - Match rule 3 if rules 1 and 2 are not matching

- Area Matching from the PWL

  When the PWL for specific via areas are specified in the EM rule file, in the following manner:

```
1. em_jmax* …   PWL VAL_1   AREA_1   VAL_2   AREA_2    L <=5
2. em_jmax  …   PWL VAL_3   AREA_3   VAL_4   AREA_4   L <=5
```

  Then the software follows the order of priority provided below:

  - Match the exact areas provided in the rule file
  - The software does not interpolate for unmatched areas (other than `AREA_1`, `AREA_2`, `AREA_3`, and `AREA_4` in above example)

- Base Rule Selection

  When the PWL for specific via area is specified in the EM rule file, **without any conditions** in the following manner:

```
em_jmax* …   PWL VAL_1   AREA_1   VAL_2   AREA_2
```

  Then the software follows the order of priority provided below:

  - Match the exact areas from the base rule
  - The software does not interpolate for unmatched areas (other than `AREA_1` and `AREA_2` in above example)

- Optimistic/Pessimistic Rule Selection

© 2020 All Rights Reserved.

When multiple rules are selected after all conditions match, the software selects the optimistic/pessimistic value based on whether the variable, `optimisticEMRuleSelection`, is set to `true` or `false`.

By default, the software selects the pessimistic value.

For example, when the following rule set is provided:

```
1. em_jmax*   EQU   1.5    …
2. em_jmax*   EQU   3.5    …
3. em_jmax*   EQU   2.5    …
```

- Rule 1 is selected by default because it is more pessimistic as compared to rules 2 and 3
- When the variable, `optimisticEMRuleSelection,` is set to `true`, then rule 2 is selected because it is more optimistic as compared to the other rules
- Rule 2 is skipped because it is neither the most pessimistic nor the most optimistic

Example

Consider the two scenarios provided below for EM rule selection.

The given rule set is as follows:

```
1. em_jmax*… current_direction up L>=5
2. em_jmax*… current_direction up L<5
3. em_jmax*… current_direction down L>=5
4. em_jmax*… L<5
```

Case 1: EM rule selection for a resistor with current direction "`up`" and with `L==3` will be as follows:

- The tool will select rules 1 and 2 because they match the current direction "`up`"
- Of the two selected rules, the tool will discard rule 1 because of the non-matching condition, `L>=5` and it will select rule 2 because of the condition, `L==3`, which matches the condition `L<5`

The final rule selection in Case 1 will be rule 2.

Case 2: EM rule selection for a resistor with current direction "`down`" and `L==2` will be as follows:

- The tool will first select rule 3 as it matches the current direction "`down`"

© 2020 All Rights Reserved.

- The tool will then discard the selected rule 3 because of the non-matching condition, `L>=5`
- The tool will then select rule 4 from the remaining rules, as rule 4 does not have any current direction condition but it matches the condition, `L<5`

The final rule selection in Case 2 will be rule 4.

Process

```
process "name" {
# Specify silicon based erosion tables
use_silicon_density true
background_dielectric_constant value
layout_scale value
temp_reference value


# EM_Model parameters
[em_tref <value>]
[em_output_wlt <silicon|drawn>]
[em_variables <variable1> [<variable2> [ ...]]]
[em_conductor_unit A/cm^2 | mA/um | mA]
[em_via_unit A | mA ]
[em_via_area_unit A/cm^2 | mA ]
[em_lifetime_units hours | years]
}
```

The process command specifies key characteristics of the device, such as background dielectric constant, reference temperature, and overall scale.

Arguments

"name"	Specifies the name of the process.
`background_dielectric_constant` *value*	Specifies the dielectric constant for the region above the top or last dielectric layer. This field is required.
`layout_scale` *value*	Specifies the ratio of the scaled dimensions divided by the drawn dimensions. The ICT file is defined based on the scaled dimensions.

© 2020 All Rights Reserved.

`temp_reference` *`value`*	Specifies the reference temperature in degrees Celsius.
EM_Model parameters	
	The process definition parameters described below are provided to establish variables or properties that apply to all em_models in the ICT file, across all layers. All process parameters for EM models are optional.
`em_tref` *`<value>`*	Specifies nominal temperature for EM analysis specified in Celsius.
`em_output_wlt [silicon | drawn]`	This determines whether the width and length parameters are written out as drawn dimensions or silicon dimensions.
`em_variables` *`<variable1> [<variable2> [...]]`*	Globally declares variables used inside the EM models. Examples of variables are: `delta_T`, `w`, `l`. The analysis tool will calculate the value for the variable as it applies to a specific em_model and apply that value during EM analysis. Variables can be declared as a string, with a space between variables. There is no default variable defined.
`em_conductor_unit [ A/cm^2 | mA/um | mA]`	Specifies the units to use for current density limits on conductor layers. The choices are amps per square centimeter (`A/cm^2`), milliamps per micron (`mA/um`), and milliamps (`mA`). Most processes use equation to specify current limit in `mA` (default value). The selection is not case-sensitive.
`em_via_unit [ A | mA ]`	Specifies the units to use for current density limits on via layers on a simple per via basis. The choices are amps (`A`) or milliamps (`mA`). The default is `A`. The selection is not case-sensitive.
`em_via_area_unit [A/cm^2 | mA]`	Specifies the units to use for piecewise linear (PWL) current density limits for via layers using density/area value pairs. The choices are amps per square centimeter (`A/cm^2`) or milliamps (`mA`). The default is `A/cm^2`. The selection is not case-sensitive.

© 2020 All Rights Reserved.

EM Data File in Voltus-Fi-XL

An EM data file specifies the technology information, such as current density limits, and provides a mapping between the layers for highlighting. The EM data file syntax is case-sensitive. The file must start with parenthesis.

Important

EM data file support in Voltus-Fi-XL is provided only for backward compatibility. It supports all the rules that were supported by APS EMIR. Any new rules are supported in the ICT file.

Following is a sample EM data file.

```
(
; this file is case sensitive
routingLayers = ("Poly" "Metal1" "Metal2" "Metal3" "Metal4" "Metal5" "Metal6")
viaLayers = ("Cont" "Nimp" "Pimp" "Via1" "Via2" "Via3" "Via4" "Via5")
viaWidthList = (("Nimp" 0.2) ("Pimp" 0.2) ("Cont" 0.2) ("Via1" 0.2)
("Via2" 0.2) ("Via3" 0.2) ("Via4" 0.2) ("Via4" 0.2))
xrefLayers = (
( "POLYcont" ("Cont" "Cont"))
( "NSDcont" ("Nimp" "Nimp"))
( "PSDcont" ("Pimp" "Pimp"))
( "poly" ("Poly" "Poly"))
( "mt1" ("Metal1" "Metal1"))
( "mt2" ("Metal2" "Metal2"))
( "mt3" ("Metal3" "Metal3"))
( "mt4" ("Metal4" "Metal4"))
( "mt5" ("Metal5" "Metal5"))
( "mt6" ("Metal6" "Metal6"))
( "Via2NoCapInd" ("Via2" "Via2"))
( "Via2" ("Via2" "Via2"))
)
avgCurrentDensSpecList = (
(nil layer "Poly" minW 0.0 maxW -1.0 minL 0 maxL 5 currentDensity ((1.0 , 110)))
(nil layer "Metal1" minW 0.0 maxW -1.0 currentDensity ((1.1 , 110)))
(nil layer "Metal2" minW 0.0 maxW -1.0 currentDensity ((1.2 , 110)))
(nil layer "Metal3" minW 0.0 maxW -1.0 currentDensity ((1.3 , 110)))
(nil layer "Metal4" minW 0.0 maxW -1.0 currentDensity ((1.4 , 110)))
(nil layer "Metal5" minW 0.0 maxW -1.0 currentDensity ((1.5 , 110)))
(nil layer "Metal6" minW 0.0 maxW -1.0 currentDensity ((1.6 , 110)))
```


© 2020 All Rights Reserved.

```
(nil layer "Cont" minW 0.0 maxW -1.0 res 1.0 currentDensity ((2.0 , 110)))
(nil layer "Nimp" minW 0.0 maxW -1.0 res 1.1 currentDensity ((2.1 , 110)))
(nil layer "Pimp" minW 0.0 maxW -1.0 res 1.2 currentDensity ((2.2 , 110)))
(nil layer "Via1" minW 0.0 maxW -1.0 res 1.3 currentDensity ((2.3 , 110)))
(nil layer "Via2" minW 0.0 maxW -1.0 res 1.4 currentDensity ((2.4 , 110)))
(nil layer "Via3" minW 0.0 maxW -1.0 res 1.5 currentDensity ((2.5 , 110)))
(nil layer "Via4" minW 0.0 maxW -1.0 res 1.6 currentDensity ((2.6 , 110)))
(nil layer "Via5" minW 0.0 maxW -1.0 res 1.7 currentDensity ((2.7 , 110)))
)
peakCurrentDensSpecList = (
(nil layer "Poly" minW 0.0 maxW -1.0 currentDensity ((1.1 , 110)))
(nil layer "Metal1" minW 0.0 maxW -1.0 currentDensity ((1.2 , 110)))
(nil layer "Metal2" minW 0.0 maxW -1.0 currentDensity ((1.3 , 110)))
(nil layer "Metal3" minW 0.0 maxW -1.0 currentDensity ((1.4 , 110)))
(nil layer "Metal4" minW 0.0 maxW -1.0 currentDensity ((1.5 , 110)))
(nil layer "Metal5" minW 0.0 maxW -1.0 currentDensity ((1.6 , 110)))
(nil layer "Metal6" minW 0.0 maxW -1.0 currentDensity ((1.7 , 110)))
(nil layer "Cont" minW 0.0 maxW -1.0 res 1.0 currentDensity ((2.1 , 110)))
(nil layer "Nimp" minW 0.0 maxW -1.0 res 1.1 currentDensity ((2.2 , 110)))
(nil layer "Pimp" minW 0.0 maxW -1.0 res 1.2 currentDensity ((2.3 , 110)))
(nil layer "Via1" minW 0.0 maxW -1.0 res 1.3 currentDensity ((2.4 , 110)))
(nil layer "Via2" minW 0.0 maxW -1.0 res 1.4 currentDensity ((2.5 , 110)))
(nil layer "Via3" minW 0.0 maxW -1.0 res 1.5 currentDensity ((2.6 , 110)))
(nil layer "Via4" minW 0.0 maxW -1.0 res 1.6 currentDensity ((2.7 , 110)))
(nil layer "Via5" minW 0.0 maxW -1.0 res 1.7 currentDensity ((2.8 , 110)))
)
```


© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

6

xDSPF Generation using Quantus

A Detailed Standard Parasitic Format (xDSPF) file is a third-party netlist file that is used to provide all the parasitic information of the design for performing simulations using the Spectre APS/XPS simulators.

The xDSPF file is created using an extractor. This file is then used as the input for performing a simulation. The simulation database that is generated is used to perform EMIR analysis in Voltus-Fi-XL. However, if you are using Quantus for extraction, you can also generate the xDSPF file from within Voltus-Fi-XL.

To create an xDSPF file, you can use either the GUI or batch mode. In batch mode, run the `extract_xdspf` command to create the file.

Related Topics

- Generating xDSPF File
- extract_xdspf (batch command)
- Extract xDSPF Form Description

© 2020 All Rights Reserved.

Generating xDSPF File

You can generate an xDSPF file using Quantus in Voltus-Fi-XL. This file is then used as the input for performing a simulation, and the generated simulation database is used to perform EMIR analysis in Voltus-Fi-XL.

Before generating the xDSPF file, ensure the following:

- The Quantus hierarchy is available as the software calls the Quantus binary from this path.
- The Pegasus, PVS, or Calibre run is performed successfully to generate the Pegasus, PVS, or Calibre database, which is the input database for generating the xDSPF file using Quantus.
- The SPF Checker utility is located at the MMSIM/Spectre installation directory. SPF Checker works only when the top-level Quantus run is successful.

To generate an xDSPF file, perform the following steps:

1. Specifying Input Data Information
2. Specifying Process Node-Specific Information
3. Specifying Technology Data Information
4. Specifying Output Information

Specifying Input Data Information

1. Choose *Extraction – Extract DSPF*.

   The Extract xDSPF form is displayed.

© 2020 All Rights Reserved.

2. Select one of the *Data Type* options: *PVS/Pegasus*, *Calibre*, or *Assura*.

© 2020
All Rights Reserved.

3. Specify the *Data Directory.*
4. Specify the *Run Name*.
5. Specify values for *Lib Name*, *Cell Name*, and *View Name*.

   These options are available only when *Assura* is selected as the data type.
6. Specify the *Post Processing Script*.

Specifying Process Node-Specific Information

1. Select a *Process Node*.
2. Select *Signal EM* if you want signal nets to be included in the xDSPF file.
3. Specify the power and ground nets in the *PG Net(s)* field.

   You can specify a value here only when the *Signal EM* option is deselected.
4. Specify the *Nets File*.

   If the Nets File is specified, the *Signal EM* and *PG Net(s)* options are ignored.
5. Select an *Extraction Type*.
6. Select a coupling mode in the *Cap Coupling Model* field.
7. Select the following check boxes, if required: *Parasitic RES Color*, *Compression*, *Reduce I-Cards*, *Comment Sub Ckt*, *Sub Conductor Model*, and *Filter RES Center*.
8. Select *Filter RES By Layer* if you want to specify the LVS layer name for which the minimum resistance value for the extracted parasitic resistors is to be applied.
9. Specify the reference node in the *Ref Node* field.
10. Specify the subnode identification character in the *Sub-Node Char* field.
11. Specify the *Fracture Via Count*.
12. Select *Via Array Spacing* and specify a value that will be used as the maximum distance criteria for grouping vias within the same array.
13. Specify the *Via Array Count*.

Specifying Technology Data Information

1. Specify either a *Technology Library File* or a *Technology Directory.*

© 2020 All Rights Reserved.

2. Specify the *Technology Name and Corner* if the *Technology Library File* is specified.
3. Specify the *Temperature* at which the extraction will be performed.
4. Specify the *No of CPUs* to be used during the Quantus extraction run.
5. Specify the *SubCkt Pin Order File.*
6. Specify the *Customized Extract Commands File* that contains user-defined commands. When you select *Add Extract Command,* the Voltus-Fi-generated extract commands override the user-defined extract commands.
7. Specify the *Parasitic blocking device cells file.*

Specifying Output Information

1. Specify the *Directory* in which the xDSPF file will be saved.
2. Specify the *xDSPF File Name*.
3. Select *Autorun Spfchecker* to automatically run the SPF Checker utility after the extraction run. This check box is selected by default.

After you specify the required values in the *Input*, *Process*, *Technology*, and *Output* group boxes of the form, click one of the following options to complete the task.

- Click *Generate CCL* to create the common command language (CCL) file or the Quantus command file that will be used to run Quantus.

  By default, `vfiQuantus-<cellname>.ccl` is created and the file opens in the Voltus-Fi-XL console. You can view the file and modify it, if required.

- Click *Generate and Run* to create the CCL file and run Quantus to create the xDSPF file in a single step. This is useful when you do not want to view the CCL file before running Quantus.

- Click *Run* to select the CCL file to use for running Quantus. This is useful if you have modified the auto-generated CCL file.

  The log is saved both in the `vfiQuantus-<cellname>.log` and in the Voltus-Fi-XL console.

Related Topics

- Extract xDSPF Form Description

© 2020 All Rights Reserved.

- extract_xdspf (batch command)
- How to Generate xDSPF Using Quantus in Voltus-Fi (video)
- Customized Extract Commands File
- Voltus-Fi Quantus Log File

© 2020 All Rights Reserved.

Extract xDSPF Form Description

Field	Description
Input	
Data Type	Lets you specify whether the input data will come from *PVS/Pegasus*, *Calibre*, or *Assura*. By default, *PVS/Pegasus* is selected.
Data Directory	Provides the path to the directory in which the input data is stored.
Run Name	Specifies the name of the PVS/Pegasus or Calibre run.
Lib Name	Specifies the name of the library that contains the layout cellviews. This option is available only when *Assura* is selected as the data type.
Cell Name	Specifies the name of the block or cellview that contains the layout cellviews. This option is available only when *Assura* is selected as the data type.
View Name	Specifies the name of the layout view. This option is available only when *Assura* is selected as the data type.
Post Processing Script	Specifies the name of the post-processing script file.
Hierarchical EMIR flow	Enables hierarchical EMIR analysis. **Note:** This field is specific to the hierarchical EMIR flow.
PGV Library Name	Specifies the block PGV library that is used for the block-level extraction performed for the hierarchical flow. **Note:** This field is specific to the hierarchical EMIR flow.

© 2020 All Rights Reserved.

Process

Process Node

Lets you specify the process node. The following options are available:

- *N3:* for N3 process nodes
- *N6/N5:* for N6 and N5 process nodes
- *N7:* for N7 process nodes
- *N10-PG*: for N10 process nodes with PG rules
- *N10*: for N10 process nodes
- *N16/N12*: for N16 and N12 process nodes
- *N20*: for N20 process nodes
- *N14*: for N14 process nodes
- *Others > =28*: for process nodes greater than N28

© 2020 All Rights Reserved.

Signal EM	Indicates that signal nets should be included in the xDSPF file.
	
	When *Signal EM* is selected, the following command is added to the Quantus command file or the common command language (CCL) file, to perform decoupled RC extraction (`rc_decoupled`) for all nets.
	
	Note: In `rc_decoupled` extraction, an RC network for the specified nets with both parasitic resistance and capacitance decoupled to ground is extracted.
	
	`extract \`
	`  -selection "all" \`
	`  -type "rc_decoupled"`
	
	When *Signal EM* is not selected, you need to provide the names of the power or ground nets on which `rc_decoupled` extraction will be performed. For all other nets, `c_only_decoupled` extraction is performed.
	
	Note: In `c_only_decoupled` extraction, only parasitic capacitance between the specified nets and adjacent nets decoupled to the specified cap ground is extracted.
	
	For example, when the following command is specified, `c_only_decoupled` extraction is performed on all nets:
	
	`extract \`
	`  -selection "all" \`
	`  -type "c_only_decoupled"`
	
	When the following command is specified, `rc_decoupled` extraction is performed on nets specified in the `vfiQrc_nets` file. For all other nets, `c_only_decoupled` extraction is performed.
	
	`extract \`
	`  -selection "nets_file ./vfiQrc_nets"`
	`  -type "rc_decoupled"`

PG Net(s)	Provides information about the power and ground nets that should be included in the xDSPF file. This field is enabled only when the *Signal EM* check box is deselected. If the *Signal EM* check box is not selected and *PG Nets* are not specified, a pop-up window opens asking for the list of power and ground nets.

© 2020 All Rights Reserved.

Nets File	Displays the names of the nets to be included in extraction. When the *Nets File* option is checked, the *Signal EM* and *PG Net(s)* options are ignored.
Extraction Type	Specifies the extraction type. ■ *RC*: Specifies that both parasitic resistors and capacitors will be extracted. ■ *R only*: Specifies that only parasitic resistors will be extracted. When *Extraction Type* selected is *R only*, the simulation time is short. While selecting the extraction type, consider the type of simulation being run. For example, to run a DC operating point (DCOP) simulation, select *R only* as the *Extraction Type* because the capacitance values are not required.
Cap Coupling Model	Specifies the coupling mode. ■ *De-coupled*: Specifies that decoupled capacitance extraction will be performed. ■ *Coupled*: Specifies that coupled capacitance extraction will be performed. By default, *De-coupled* mode is selected. While specifying the coupling mode, ensure that there is no `extract` command defined in the additional CCL file. The `extract` command in the additional CCL file will override the above options.
Parasitic RES Color	Enables color-aware EM analysis.
Compression	Generates xDSPF in a compressed format.
Reduce I-Cards	Enables the `reduce_i_cards` option in the CCL file that is used for generating the xDSPF file. If a net is “capacitance only”, it is qualified for reduced I card or instance card treatment. This means that when this option is selected, the instance cards and the corresponding dummy resistors are not printed in the results. **Note:** It is recommended not to select this option while performing extraction for the ESD analysis flow.

© 2020 All Rights Reserved.

Comment Sub Ckt	Comments out the top sub circuit statements from the xDSPF file.
Sub Conductor Model	Provides information about the parasitic resistor models for the subconductor layers that should be included during extraction. This is used to support different EM rules for poly layers during EM analysis.
Filter RES Center	Includes sub-nodes after eliminating resistors with small resistance values. When this option is selected, the survivor sub-nodes, among a group of sub-nodes that are formed by the filtered resistors, are chosen based on their proximity to the geometric center of the group. This field is specific to process nodes 5nm and below.
Filter RES By Layer	Lets you specify the LVS layer name for which the minimum resistance value for the extracted parasitic resistors is to be applied. If the layer name is not specified, the default value, `RH_TN_6 0`, is used. This field is specific to process nodes 5nm and below.
Ref Node	Specifies the reference node on which the extraction is to be performed. By default, `0` is specified.
Sub-Node Char	Indicates the subnode identification character. By default, `#` is specified.
Fracture Via Count	Indicates the number that will be used to divide pseudo vias into segments during extraction.
Via Array Spacing	Indicates the distance that will be used as the maximum distance criteria for grouping vias within the same array. This means that if the distance between vias is less than the specified maximum distance, they are grouped within the same array. The default value of the via array spacing is 0.35 times the via size for N10, N16, N12, and N20 processes. For other processes, the default is *auto*, in which case Voltus-Fi sets the via spacing automatically.
auto	Indicates that Voltus-Fi sets the via spacing automatically.
Via Array Count	Indicates the number of vias per side allowed in an array (without the per via layer count option). The default value is `4`. For example, if the via array count is set to 2, the number of vias allowed per array side will be 2.
Bounding box options	Sets the conductor bounding box and the via bounding box by layer.

© 2020 All Rights Reserved.

Technology	
Technology Library File	Displays the location of the technology file to be used for reading the technology data.
Technology Directory	Displays the name of the directory that contains the technology file.
Technology Name	Displays the name of the technology file to be used for extraction. This field is optional if the technology directory has been specified.
Corner	Displays the name of the technology corner to be picked from the library.

Temperature	Indicates the temperature at which the extraction will be performed.
	
	By default, a temperature of 25 degree Celsius is used. The temperature specified for extraction is optional. The options, temperature and `temp_coeff`, specified in the CCL file are mutually exclusive. The following scenarios hold true.
	
	■ If you specify a temperature for extraction in the Extract xDSPF form or by using the `-temperature` parameter of the `extract_xdspf` command in batch mode, the following option in the CCL file will be set to `false`:
	`-include_parasitic_res_temp_coeff "false" \`
	The software displays the following message:
	`User specified temperature will be used for extraction. The option include_parasitic_res_temp_coeff will be set as false.`
	
	■ If you do not specify a temperature for extraction, the following option in the CCL file will be set to `true`:
	`-include_parasitic_res_temp_coeff "true" \`
	The software displays the following message:
	`No user specified temperature value so the option include_parasitic_res_temp_coeff will be set as true`

No of CPUs	Displays the number of CPUs to be used during the Quantus extraction run.

© 2020 All Rights Reserved.

SubCkt Pin Order File	Provides the order of pins to be used during extraction.
	
	The pin order is specified in the xDSPF file using the `.SUBCKT` and `.ENDS` statements. You can use this option to provide a file with a different pin order that will override the order specified using the default `.SUBCKT` statement in the xDSPF file.
	
	Below are examples of pin orders in the xDSPF file before and after specifying the pin order file.
	
	Without the sub-circuit pin order file
	
	`.SUBCKT top In_001 VDD Out_001 VSS In_002 Out_002 In_003 Out_004 ...`
	`.ENDS`
	
	With the sub-circuit pin order file
	
	`.SUBCKT top VDD VSS In_001 Out_001 In_002 Out_002 In_003 Out_004 ...`
	`.ENDS`

Customized Extract Commands File	Lets you specify the file with user-customized extract commands.
Add Extract Command	Indicates that Voltus-Fi adds the extract commands to the CCL file. By default, when a `customextract` file is specified, all commands, including the extract commands, from that file are added to the Voltus-Fi-generated CCL file. However, when you select *Add Extract Command*, Voltus-Fi adds the extract commands to the CCL file, and all other commands specified in the `customextract` file are added to the CCL file.
Parasitic blocking device cells file	Lets you specify the file containing parasitic blocking device cell names, if any. Once a cell is declared a blocking cell, all the parasitic RCs inside the cell are not extracted. If the cell content overlaps top cell routing, then, the top cell routing's RCs in the overlapping regions are also not extracted.
Output	
Directory	Specifies the output directory in which the xDSPF file will be saved. If this file is not specified, a pop-up window opens asking for the name of the directory.

© 2020 All Rights Reserved.

xDSPF File Name	Specifies the name of the xDSPF file that will be generated. If you do not specify the filename, Voltus-Fi generates a file with the name, `<runName>.dspf`, by default.
Autorun Spfchecker	Runs the SPF Checker utility after the extraction run. SPF Checker checks the syntax of the DSPF/SPEF followed by opening the output log. It also compares the terminal names in the DSPF and the device models and populates the summary section with the recommended mapping statements or the `spf aliasterm` statements.
Generate CCL	Creates the CCL or the Quantus command file that will be used to run Quantus.
Generate and Run	Creates the CCL file and runs Quantus to create the xDSPF file in a single step.
Run	Selects the CCL file to use for running Quantus.
Reset	Displays all the default values.
Cancel	Closes the form without executing the command.
Save	Saves the xDSPF file.
Load	Loads the configuration file.
Help	Opens help information in Cadence Help.

Related Topic

- Generating xDSPF File

Customized Extract Commands File

The customized extract commands file is a file in which you can specify extract commands as per your requirements. When this file is specified, the Voltus-Fi-generated common command language (CCL) file will not have any extract commands, and the commands from the customized extract commands file will be added in the Voltus-Fi-generated CCL file.

A sample customized extract commands file is as follows:

```
extract \
   -selection "all" \
   -type "rc_coupled"
```


© 2020 All Rights Reserved.

```
extract \
   -selection nets_file "file.txt" \
   -type "none"
```

Note: You can create a `customextract` file by using either single or multiple extract settings. Ensure that the commands are in the Quantus CCL file syntax because this file is added as is in the Voltus-Fi-generated CCL file.

Related Topic

- Generating xDSPF File

Voltus-Fi Quantus Log File

The Voltus-Fi Quantus log file (`vfiQuantus-<cellname>.log)` stores all the common command language (CCL) commands and the info, warning, and error messages issued by Voltus-Fi during the extraction run. This file is saved in the output directory that is specified in the Extract xDSPF form.

With every successive extraction run, the name of the log file is updated to `vfiQuantus-<cellname>.log.(n-1)`. Here, n-1 is the run number. Therefore, the greater the numeric value in the filename, the more recent or latest that file is.

For example, after three extraction runs, the following log files are created:

- `vfiQuantus-<cellname>.log` (log file for the first run)
- `vfiQuantus-<cellname>.log.1` (log file for the second run)
- `vfiQuantus-<cellname>.log.2` (log file for the latest run)

Related Topic

- Generating xDSPF File

Voltus-Fi Extract xDSPF GUI Last File

The Voltus-Fi Extract xDSPF GUI last file (`.vfiExtractDSPFGuiLast`) is created by Voltus-Fi to save the settings of the Extract xDSPF form of the previous session. This file gets loaded automatically when the next session starts. If you do not want to load these settings, remove this file before starting the session. This file is created along with `cds.lib`.

© 2020 All Rights Reserved.

The values specified in the `.cdsinit` file override the values of the `.vfiExtractDSPFGuiLast` file of the previous session.

Related Topic

- Generating xDSPF File

© 2020 All Rights Reserved.

7

Static Current Analysis

- Static Current Analysis on page 142
- Static EMIR Flow in Spectre on page 142
- Static Current Analysis Overview on page 142
- Static Current Analysis Flow on page 144
  - Data Requirements for Static Current Calculation on page 144
  - Setting Up and Running Static Current Analysis on page 144
    - Setting Up and Running Peak Static Current Analysis on page 144
    - Setting Up and Running the Average Static Current Analysis on page 149
- Static Current Analysis Report on page 153

© 2020 All Rights Reserved.

Static Current Analysis

Static current analysis is performed in Voltus-Fi-XL using Thunder. It is performed before running a simulation in Spectre. The static current analysis output file, `static_current.txt`, is used as an input for performing a Spectre simulation. Following are the key aspects of this feature:

- Support for both, peak and average modes for static analysis
- Support for both, GUI and Tcl commands

Static EMIR Flow in Spectre

A static EMIR analysis allows the evaluation of IR drop and EM currents based on user-provided subcircuit instance current consumption w/o running a transient or DC simulation. The user-provided currents are distributed to the tap devices based on the W/L ratio of the devices in the design. The IR drop and EM current analysis is performed based on the current at each tap device.

The static EMIR analysis is enabled with the "`static ifile`" statement in the EMIR conf file.

```
net name=[I1.VDD I1.VSS] analysis=[vavg iavg]

static ifile="static_currents.txt"
```

Since the analysis is static, the IR drop `vmax` and `vavg`, and the EM current `imax`, `irms`, and `iavg` values will be the same, and just one can be selected in the statement.

For more information, see the "Static EMIR Analysis" section in the Postlayout Simulation chapter of the *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

Static Current Analysis Overview

Static current analysis can either be static peak current analysis (Ipeak) or static average current analysis (Iavg). Static peak current analysis is based on the peak saturation transistor currents. Because of simplicity and ability to find problems quickly in power grids, static current analysis is the most commonly performed analysis.

Static average current analysis or activity-based analysis resolves the distribution of currents on your power grid better than the static peak current analysis. The activity-based approach

© 2020 All Rights Reserved.

assumes that you have a mechanism such as a Verilog simulator to compute and report the relative activity of the nets in the design. These relative activities can be used in conjunction with net capacitances to estimate the average current load of each gate in the design. This form of analysis clearly provides more realistic current data than the Ipeak approach.

Definition of Activity, Duty Cycle, and Transition Density

Activity means the probability of all the signal nets in design switching from `0->1` or `1->0` in one clock cycle. For instance, if the activity of a net or instance is `0.1`, then the power engine assumes that the net or instance will switch from `0->1` or `1->0` once every ten clock cycles.

For the above diagram,

Activity = `(Number of (0->1 or 1->0) transitions/Number of clock cycles) =2/5=0.4`

Duty Cycle means the probability that a signal net has the value of `1`.

For example, if the signal of a net is 1 for 2ns in the total simulation time of `10ns`, then the duty cycle of net is `0.2`. The duty cycle of the signal in the previous diagram is `0.5` (2.5/5). However, if a signal is `Z` or `X` for some time and `0` for the rest of the time then the duty cycle of the signal is `0`.

Transition Density means the number of times the signal toggles from `0->1` or `1->0` in `1` second.

For the previous diagram, if you assume that one clock cycle is `4ns`, then:

Transition Density = `1e+08 (2/20ns)`

© 2020
All Rights Reserved.

Static Current Analysis Flow

In this section, the following topics will be covered:

- Data Requirements for Static Current Calculation
- Setting Up and Running Static Current Analysis

Data Requirements for Static Current Calculation

The following data is required to perform static current calculations:

- A Quantus-generated DSPF file
- SPICE model libraries and SPICE corners
- An activity, clock, or value change dump (VCD) file—These inputs are required for the average static current analysis

Setting Up and Running Static Current Analysis

The following topics are covered in this section:

- Setting Up and Running Peak Static Current Analysis
- Setting Up and Running the Average Static Current Analysis

Setting Up and Running Peak Static Current Analysis

Peak static current (Ipeak) analysis gives the peak current (saturated current) of every transistor connected to the power nets.

To set up and run the static current analysis mode for peak current analysis, perform the following steps:

- In the Virtuoso® console, choose *Simulation* and then *Setup and Run Simulation.* The Setup and Run Current Analysis form opens. The form has two tabs, *Basic* and *Advanced*. The *Basic* tab is shown below.

© 2020 All Rights Reserved.

Figure 7-1 Setting Up and Running the Peak Static Current Analysis – Basic Tab

On the *Basic* tab, provide the following information:

- Select *Peak* as the *Analysis Type*.
- The *Top Cell Name* is populated by default from the layout.
- Specify the *Net Names* for which you want to calculate the peak current.
- In the *Netlist Files* group box, specify the *Spice Netlist* file, which is the DSPF file.

  Note: The Spice netlist file can also be edited using the edit button provided next to this

© 2020
All Rights Reserved.

field.

- In the *Pins* group box, specify the *Power Pins* and *Gnd Pins* (ground pins) for which current analysis is to be performed. Specify the corresponding *Voltages* for the power pins.
- In the *Model Files* group box, choose the *Spice Models* and the corresponding *Spice Corners* used by the SPICE netlist. For this, perform the following steps:
  - Specify the path of the SPICE model files.
  - Specify the corresponding *Spice Corners* to be used from this model file.
  - Click *Add* to add the SPICE model – SPICE corner pair to the table.
  - Continue till all the models and their corresponding corners are specified.
  - Click *Remove* to remove a SPICE model – SPICE corner pair from the table.
- In the *Output* group box, specify the *Directory,* which is the output directory in which all the current data will be saved. By default, for every static current analysis run, the tool will save the results in the directory with the naming convention, "`vps_simulation_#`", where the number is increased incrementally for each run. For example, `vps_simulation_1`, `vps_simulation_2`, and so on.
- Select the *Advanced* tab. This form is shown below.

© 2020 All Rights Reserved.

Figure 7-2 Setting Up and Running the Peak Static Current Analysis – Advanced Tab

On the *Advanced* tab, provide the following information:

- Specify the *Hierarchical Separator.* This is required only if the netlist used is hierarchical.
- Specify the *Simulation Temperature*, in degree Celsius, at which to perform the simulation. By default, the simulation is performed at 25 degree Celsius.

  Note: The *Static Options* group box is disabled for the peak current analysis because

© 2020 All Rights Reserved.

these options are specified only for the average static current analysis.

- Specify the *Cmd Include File* that passes additional user commands to Thunder during static current analysis.
- Click *Run.* If any required input is missing in the form, a pop window opens to list all the required inputs. This is shown below.

Tcl Commands:

An example of the set of Tcl commands for the peak static current calculation is as follows:

```
setup_simulation\
    -spice_netlist  design.spice\
    -topcell design_name\
    -output_directory ./vps_simulation_1
spice_model\
    -spice_lib ./models/model1.l\
    -spice_corner {TT TT_dio}
spice_model\
    -spice_lib ./models/model2.l.special\
    -spice_corner {TT_sr}
power_pin\
```


© 2020
All Rights Reserved.

```
    -pin_name VDD\
    -voltage 0.9
ground_pins VSS
run_static_simulation\
    -method Ipeak\
    -nets {VDD VSS}
```

For details of the above commands, see the “Batch Mode Execution” chapter.

Setting Up and Running the Average Static Current Analysis

Average static current analysis is a method of analyzing a circuit to obtain the average current flowing through each tap point without using time-based simulation.

For the average (Iavg) static current calculation, the information in the *Basic* tab of the Setup and Run Current Analysis form remains the same as that filled in for the peak static current analysis, except for specifying the *Analysis Type*. This is shown below.

© 2020 All Rights Reserved.

Figure 7-3 Setting Up and Running the Average Current Analysis – Basic Tab

- Choose *Average* as the *Analysis Type*.
- Specify the *Spice Netlist* file, which is the DSPF file.
- Select the *Advanced* tab. This form is shown below.

© 2020
All Rights Reserved.

Figure 7-4 Setting Up and Running the Average Current Analysis – Advanced Tab

In the *Static Options* group box, provide the following information:

- Specify the dominant *Frequency* of the design.
- Specify the *Clock File.*
- Specify the *VCD File*. This file contains information about the number of transitions for each net.
- Specify a value for *Global Activity* in the design. This value represents the average number of times all the unset nodes switch in a clock cycle. Alternately, you can specify

© 2020 All Rights Reserved.

the path of the *Activity File*, which contains the data for the activity factor on any node of the circuit.

- Specify the *Avg Power (All Nets)* of the design. This is the average power value for all the nets. The default unit is `W` (Watts). If the value is in any other unit, you must specify the unit. For example, `20mW`. You can also specify the path of the *Avg Power File*, which contains the average power for specific sub circuits.
- Select *Net based average power* to specify different power values for different nets. When you check this option, a list box is enabled. The list box has two columns, "Net Name" and "Average Power in W". The "Net Name" column is already populated with the names of nets specified on the Basic tab. For each net, specify the power value in the "Average Power in W" column.

  Note: The *Avg Power (All Nets)* and *Net based average power* options are mutually exclusive.
- Click *Run.*

  Note: You can also create or edit the clock file, VCD file, activity file, and the average power file by using the edit buttons provided against each of these options in the form.

  For more information about the formats of above files, see Static Current Analysis Files in the "File Formats" chapter.

Tcl Commands:

The example Tcl command for the average static current analysis is as follows:

```
setup_simulation\
    -spice_netlist adc_sample_hold.spice\
    -topcell adc_sample_hold\
    -output_directory ./vps_simulation_1\
spice_model models/spectre/gpdk090.scs\
    -spice_corner {NN}
power_pin\
    -pin_name VDD\
    -voltage 2.5
ground_pins VSS
run_static_simulation\
    -method Iavg\
    -nets {VDD VSS}\
    -freq 333\
    -global_activity 0.25\
```


© 2020 All Rights Reserved.

```
-average_power 20mW
```

For details of the above commands, see the “Batch Mode Execution” chapter.

Static Current Analysis Report

The output of the static current analysis in Voltus-Fi-XL using Thunder is a text report that is saved in the output directory. The report is used by Spectre to run a simulation and create the simulation result file, which is then used for EMIR analysis and visualization in Voltus-Fi-XL. The default text report generated for both peak and average current analysis is `static_current.txt`.

Format:

```
Node Name          Static Current
```

Format Description:

`Node Name` specifies the name of the node for which the static current is being reported.

`Static Current` specifies the value of the static current on the specified node.

Sample Report

```
X0.AbcI2/avI33/avD30_32    9.83e-05
X0.AbcI2/avI30/avD30_35    9.83e-05
X0.AbcI86/avD27_135        1.96e-05
X0.AbcI86/avD27_134        1.96e-05
X0.AbcI86/avD27_133        1.96e-05
X0.AbcI86/avD27_122        1.96e-05
X0.AbcI86/avD27_121        1.96e-05
X0.AbcI86/avD27_120        1.96e-05
```


© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

8

IR Drop Analysis Results

- Overview on page 158
- Batch Mode Support for IR Drop Analysis
- Signal Net IR Drop Analysis on page 158
- Power Gate Support on page 159
- Static EMIR Analysis on page 162
- Static Power Grid Solver on page 163
- Solid Shape Highlighting on page 165
  - Quantus Shape Database-Based Flow on page 165
  - Support for Specifying Cut-Layers for Metal Layers while Highlighting Solid Shapes on page 166
- Displaying Finer Color Gradient for Selected Layers in IR and EM Plots on page 168
- Specifying Multiple Simulation Result Files for EMIR Analysis on page 169
  - GUI Support for Specifying Multiple Bin Files in Different Directories on page 169
  - GUI Support for Specifying Multiple Bin Files in the Same Directory on page 170
  - Batch Mode Support for Specifying Multiple Bin Files on page 171
- Viewing the IR Drop Analysis Results on page 172
- Displaying and Querying EMIR Results on page 184
  - Viewing EMIR Violations in Annotation Browser on page 185
  - Querying EMIR Results on page 189
  - Specifying the Display Options for IR/EM Plots on page 190
- Types of IR Drop Analysis Plots on page 215

© 2020 All Rights Reserved.

- IR, IRAVG – IR Drop Plots on page 216
- RC, RCAVG, and RCRMS – Resistor Current Plots on page 217
- IV – Transistor Voltage Plot on page 219
- TC, TCAVG, TCRMS – Tap Current Plots on page 220
- POWER – Resistor Power Plots on page 222
- PI – Powergate Current Plot on page 224
- PV – Powergate Voltage Plot on page 227
- REffective – Effective Resistance Plot on page 229

- Plotting the Least-Resistive Path on page 230
  - Batch Mode Support for LRP Analysis
  - GUI Support for LRP Analysis
    - The LRP Browser on page 233
    - Shorting Layers for LRP Analysis on page 239
    - Displaying LRP for the Node on the Selected Marker on page 241
    - Supporting LRP Analysis in Power-Gated Designs on page 243
- Calculating Effective Resistance between any Two Nodes on a Net on page 247
  - Batch Mode Support for Calculating Effective Resistance on page 247
  - GUI Support for Calculating Effective Resistance on page 248
- Generating EMIR Analysis Reports in Voltus-Fi-XL on page 252
  - Reports Generated Using the EMIR Control File on page 252
  - Default Reports Generated for EMIR Analysis on page 254
  - RLRP Report on page 259
  - Reports Generated for Multiple Bin Files from Different Testbenches in a Design on page 260
- Reporting a List of Devices from a Subnode on page 266
  - Use Model on page 266
  - Sample Command File and Results File on page 267

© 2020 All Rights Reserved.

- Reporting On-Resistance (RON) Values on page 269
  - Overview on page 269
  - Generating the RON Report on page 269
  - Output on page 269

© 2020 All Rights Reserved.

Overview

Voltus-Fi-XL uses the simulation database generated by Spectre APS/XPS simulators and displays the results of the IR drop analysis on the Virtuoso layout.

It also generates text reports of the analyses and lets you query the analyses results to view specific violations in the layout. This is used to debug the high IR drop regions in the design.

Batch Mode Support for IR Drop Analysis

Batch mode support is provided for loading IR drop analysis results and generating text reports. The following commands are used to load and print IR drop analysis results in Voltus-Fi-XL:

- load_ir_results

  Loads the IR drop analysis results

- print_ir_report

  Prints the IR drop analysis reports

The detailed description, syntax and examples of how to use these commands are provided in the load_ir_results section of the “Batch Mode Execution” chapter.

Signal Net IR Drop Analysis

By default, IR drop analysis can only be applied to power nets, which are nets driven by DC or constant voltage sources. The advanced feature, signal net IR drop analysis, lets you perform IR drop analysis on signal nets, which are nets that are not connected to any DC or constant voltage source.

In the signal net IR drop analysis, the maximum or average IR drop is reported using the `analysis=[sigvmax sigvavg]` statement in the EMIR control file (`emir.config`). The signal net IR drop is reported as the difference between the reference voltage and the voltage of the sub node (vref-vnode). For a particular signal net, the reference voltage for calculating the IR drop value at each sub node and time point is defined as the average voltage of all sub nodes (reftype=[avg]), the maximum voltage of all sub nodes (`reftype=[max]`), or the voltage of a user-specified pin (`reftype=[pin]`).

Alternatively, for nodes driven by voltage sources and the voltage of the net driven by the voltage source can be used as reference (`findsrc=yes`).

© 2020 All Rights Reserved.

The following is an example of the EMIR control file (`emir.config`):

```
net name=[X1.*] analysis=[sigvmax] reftype=[max] findsrc=yes
solver method=direct
```

In the above example, the signal net IR drop analysis is performed for all nets `X1.*`. The maximum IR drop for the sub nodes of each net is reported. For the IR drop analysis at each net and time point, the sub node with the highest voltage is picked as the reference voltage. For any net driven by a vsource, the voltage of the sub node driven by the vsource is used as reference.

When using `reftype=[max]` for a net with three sub nodes: `net1_1` (1V), `net1_2` (2V), and `net1_3` (3V), the detected reference voltage will be 3V, and the IR drop calculated will be 2V for `net1_1`, 1V for `net1_2`, and 0V for `net1_3`. When using `reftype=[avg]` the reference voltage will be 2V, and the IR drop calculated will be 1V for `net1_1`, 0V for `net1_2`, and -1V for `net1_3`.

In addition, if `net1_2` is driven by a `vsource` and `findsrc` has been specified as yes, the voltages of `net1_2`, at all time points, will be used as the reference voltages.

For details of the supported EMIR control file options that can be specified, see EMIR Control File Options Supported in Voltus-Fi-XL in the “Data Preparation” chapter.

Power Gate Support

Power networks may contain power gates which enable or disable the power supply in the circuit. These power gates split the power supply RC network into two parts. The RC network driving the power gate, and the RC network being driven by the power gate. This is shown in the figure below. The MMSIM EMIR analysis can handle such power gates.

© 2020 All Rights Reserved.

Figure 8-1 Power Gates

To invoke the power gating handling, add the power gate setting in the EMIR control file (`emir.config`), as shown below.

```
net pwrgate=[vdd vdd_int] analysis=[vavg]
```

Both, the power supply net (`vdd`) driving the power gates, and the internal power supply net (`vdd_int`) driven by the power gates, need to be specified in the net statement in any order. Wildcarding is supported for the internal power supply net(s), but not for the power supply net.

Next, the required analyses need to be defined, as shown above. The IR drop report includes both the power supply net in the usual report filename, for example, input.rpt_ir and the internal power supply net is reported in a file, such as `input.rpt_pwg` or `input.emirtap.rpt_pwg` for direct and iterated methods, respectively. As for the EM analyses, they are performed as usual, if specified.

In addition, an important parameter `Ton` is also be reported in the `.rpt_pwg` file. It reports the time taken to power up the terminal of the internal power supply net to 95% of the power supply level (VDD).

Note: The parameter `Ton` can be infinity if the internal power supply level does not reach 95% of VDD.

The subsequent sections cover the various options available in Voltus-Fi-XL for viewing the IR drop analysis results on the Virtuoso layout.

© 2020 All Rights Reserved.

Viewing the Power-Up Summary Report

Voltus-Fi-XL supports viewing of the power-up test summary report in the GUI. The feature is enabled only after the IR drop analysis results are loaded. For this, the *Power-up Report* button is provided on the IR tab of the IR/EM Results form.

When clicked, a view-only power-up summary report is displayed in the GUI. The report is available only if the following conditions are met:

- The `emirutil power_gate_extra_report` command is specified in the EMIR configuration file
- Spectre simulation is run with the above option and a `rpt_pwg` file is generated.

A sample power-up report is shown below.

Figure 8-2 Power-Up Summary Report

In addition to information about the type of simulation results, the report contains the following information:

- *NumPwgOn*: The number of power switches turned on during the simulation vs. the total number of power switches
- *LastPwgOn*: The name of the last power switch to be turned on in the simulation
- *Peak rush current (Imax)*: The peak rush current of the power-up net in the simulation

© 2020 All Rights Reserved.

- *Average wake up time*: The average wake-up time of the switch net in the design
- *Comment*: The link to the detailed power-up report
- Details of the wave form analysis reports

Static EMIR Analysis

A static EMIR analysis evaluates IR drop and EM currents based on user-provided subcircuit instance current consumptions without running a transient or DC simulation. The user-provided currents are distributed to the tap devices based on the W/L ratio of the devices in the design. The IR drop and EM current analysis is performed based on the value of current at each tap device.

The static EMIR analysis is enabled with the "`static ifile`" statement in the EMIR configuration file.

```
net name=[I1.VDD I1.VSS] analysis=[vavg iavg]
static ifile="static_currents.txt"
```

Since the analysis is static, the IR drop `vmax` and `vavg`, and the EM current `imax`, `irms`, and `iavg` values will be the same, and just one of these can be selected in the statement.

The subckt instance currents are defined (in A) in the "`static_currents.txt`" file with the subckt instance name, the subckt port name, and the value of current flowing in the port.

```
I1 VDD 0.001
I1.I2 VDD 0.005
I1.I2/I3 VDD 0.00001
I1.I2/I3/I4 VDD 0.000005
I1 VSS ?0.005
```

The accuracy of the static EMIR analysis depends greatly on the detailed current information in the `static_ifile`. It is highly recommended that you provide the current consumption for each subckt instance.

The results of the static EMIR analysis are written into the same IR drop and EM current text reports, and into the same binary EM database as the dynamic EMIR analysis. Static EMIR analysis cannot be combined with the dynamic analysis. You can perform either a static or a dynamic EMIR analysis at one time. When running the static EMIR analysis all other Spectre analyses like DC, TRAN and so on will be ignored.

The static EMIR analysis can also be applied to designs with power gates. In that case the `pwrgate` statement needs to be added to the EMIR configuration file as follows:

```
net name=[I1.VDD] analysis=[vavg iavg]
```


© 2020 All Rights Reserved.

```
static ifile="static_currents.txt"
net pwrgate=[I1.VDD I1.VDD_INT] analysis=[vavg iavg]
```

and the power gate current is required to be added to the "`static_currents.txt`" file as follows:

```
I1 VDD 0.001
I1 VDD_INT 0.0005
```

Note: To support users setting up the static EMIR flow, the static EMIR current file is automatically generated when running a dynamic EMIR analysis (file extension: static).

Static Power Grid Solver

The static power grid solver (SPGS) feature of Spectre can be used in Voltus-Fi-XL to calculate all pin-to-tap resistances based on the description of a DSPF file and the options set in the EMIR configuration file. The resistances calculated by SPGS are electrically equivalent resistances, and not the summation of resistors. The calculation assumes that all pins are connected together to form a global pin. After calculation, the resistance between the global pin and all taps is generated and listed based on the significance of their values.

This feature can also be used to calculate the pin to pin resistance between any two nodes of the same net. For details, see Calculating Effective Resistance between any Two Nodes on a Net.

The diagram below shows the definition of pin, sub nodes, and tap nodes in the DSPF file.

Figure 8-3 Pin, Sub Node, and Tap Node Information in the DSPF File

© 2020 All Rights Reserved.

```
*|NET vcc 0PF                                                    pll.spf
*|P (vcc_1 B 0 -1.441 419.479)
*|P (vcc B 0 2158.41 419.491)
*|I (xl107/mxP1:BULK xl107/mxP1 BULK B 0 2128.46 392.112)
*|I (xsvdc/mxP3@3:SRC xsvdc/mxP3@3 SRC B 0 1837.78 431.24)
*|I (xsvdc/mxP2:SRC xsvdc/mxP2 SRC B 0 1837.78 429.16)
*|S (vcc:2 40.43 193.665)
...
R35047 vcc_1 vcc_7 0.001 $l=0.004 $w=10 $lvl=195
...
```

The static EMIR analysis is enabled with the "`spgs net=...`" statement in the EMIR configuration file.

```
spgs net=[i1.vcc i1.vss]
spgs rshort=1e-12
spgs pwrgate = [i1.vcc i1.outp]
```

The net statement defines the power nets to be analyzed. Resistors in the power net can be optionally shorted with the `rshort` option. If the power net contains a power gate, then the `pwrgate` option needs to be defined in the configuration file.

The SPGS flow requires the DSPF file to be included in the Spectre input file with the `dspf_include` statement, as shown below.

```
dspf_include "pll.spf"
```

The power nets need to be connected to voltage sources, and device models need to be defined for the devices in the instance section in the DSPF file. For running the SPGS feature, Spectre is run as in the regular EMIR flow, as shown below.

© 2020 All Rights Reserved.

```
spectre +aps input.scs +emir=spgs.conf
```

The SPGS report provides a list of the pin to tap node resistors ordered highest to lowest, as shown below.

```
Count R          W/L   (x:y)       (x:y )in GDSII  name
#1    1920.2579 5.00  (1440:366) (1440:366)      xI107/mxP1:BULK
#2    1577.1482 10.00 (1440:476) (1440:476)      xsvdc/mxP3@3
#3    1353.6123 4.00  (1446:509) (1446:509)      xsvdc/mxP2:SRC
```

It is recommended to run the SPGS feature separately and not together with any other EMIR analysis. SPGS also supports the conversion between pin, tap node, and sub node. The following statements specified in the emir conf file convert the tap node, `MPM3@44:s` of net `I1.vdd` to a pin, and the subnode `VDD:5` to a tap node.

```
spgs tap2pin net=[i1.vdd] include=[MPM3@44:s]
spgs sub2tap net=[i1.vdd] include=[VDD:5]
```

All conversion options are described in the EMIR Control File Options Supported in Voltus-Fi-XL table in the “Data Preparation” chapter.

Solid Shape Highlighting

The rail analysis plots that are displayed on the Virtuoso Layout can be viewed as either stick diagrams or solid shape highlights. In the solid shape display, the plots highlight the full shape of the presistors, while in the stick diagram display, the presistors are connected with thin lines over the layout shapes.

The solid shape display is more useful for designers because it allows them to see the shapes that are failing the analysis. There are two flows for solid shape display. You can either specify the DFII layer map file or the Quantus run name for viewing solid shape display for plots. When you specify the Quantus run name, the plots are displayed based on the shape database generated by Quantus.

For details of how to specify the two options in the GUI, see Viewing the IR Drop Analysis Results.

Quantus Shape Database-Based Flow

To display plots using the shape database generated by Quantus, you must ensure the following:

- The Quantus command file or the CCL file should include the following command:

© 2020 All Rights Reserved.

```
extraction_setup -analysis em
```

For details, see the `extraction_setup` command in the “Quantus Command Files” chapter of the *Quantus Extraction Users Manual*.

- Quantus should be set in the path and its version should be compatible with the Voltus-Fi-XL version being used.

Support for Specifying Cut-Layers for Metal Layers while Highlighting Solid Shapes

In Voltus-Fi, cut-layers can be specified for metal layers for displaying the EMIR result plots. The cut-layers do not actually chop the shapes on the metal layers in the layout, but when plotting the results, while the shapes on the metal layers are highlighted, those on the specified cut-layers are not highlighted. This helps to view more accurate plots.

An example of a plot is shown below. In this plot, the shapes on metal layer, `M1` are highlighted while those on the cut-layer, `CM1` are excluded.

© 2020 All Rights Reserved.

Figure 8-4 Specifying Cut-Layers for Metal Layers

To let you specify cut-layers for metal layers, the following SKILL function is provided:

vsaSetLayersToMergeDuringResultsLoading

Use the SKILL function to specify the metal layer and the corresponding cut-layer as a pair. For example,

```
vsaSetLayersToMergeDuringResultsLoading("M1 CM1A")
```

where `CM1A` is the cut-layer for metal layer, `M1`.

© 2020 All Rights Reserved.

Note: The above SKILL function must be called before loading the results in Voltus-Fi.

Following are the key aspects of this feature:

- You can specify multiple pairs of metal and cut-layers. For this, call the function multiple times. Voltus-Fi honors all these calls. For example,

  ```
  vsaSetLayersToMergeDuringResultsLoading("M1 CM1A")
  vsaSetLayersToMergeDuringResultsLoading("PO CPO")
  ```

  where `CM1A` is the cut-layer for metal layer, `M1` and `CPO` is the cut-layer for layer, `PO`.

- You can specify multiple cut-layers for one metal layer. For example, if the following sequence of calls are made, Voltus-Fi honors both the calls.

  ```
  vsaSetLayersToMergeDuringResultsLoading("M1 CM1A")
  vsaSetLayersToMergeDuringResultsLoading("M1 CM1B")
  ```

  where, `CM1A` and `CM1B` layers are cut-layers for the same metal layer, `M1`.

Displaying Finer Color Gradient for Selected Layers in IR and EM Plots

In the DFII layer map flow, for solid shape display of IR/EM violations on the Virtuoso layout, Voltus-Fi-XL locates all the nodes lying on a shape and then colors the shape with the worst violation value among all resistors connected to these nodes.

However, sometimes resistors do not align well with the shapes used to display the violations. This may be because the shapes are large and have multiple nodes and resistors. Voltus-Fi supports a finer display of the color gradient for violations in different segments of such layers. The shape polygons are split into subpolygons and then the subpolygons are colored with the worst violation value for all resistors connected to the nodes in the split shape. This feature is turned on by default for all layers. However, you can select specific layers to view their finer gradient instead of viewing for all layers.

For this, specify the *DFII Layermap* file and then click the *Finer Gradient* button provided on the IR or EM tabs of the IR/EM Results form. In the *Finer Gradient Layer Selection* pop-up, click *Deselect All* and then select the PGDB layers for which you want to view finer gradient. For details of the steps involved, see Viewing the IR Drop Analysis Results.

Note: Perform this step before loading the IR or EM Results. After loading the results, when you click *Show Plot*, you can see the finer gradient for the selected layers.

A comparison of finer gradient display versus the original display is shown in below figure.

© 2020 All Rights Reserved.

Figure 8-5 Viewing the Finer Gradient Display

Original Display

Finer Gradient Display

Specifying Multiple Simulation Result Files for EMIR Analysis

While performing EMIR analysis in Voltus-Fi-XL, multiple bin files can be specified, both in the GUI mode as well as in batch mode. There are two different use models for specifying multiple bin files depending upon whether they are present in the same or different directories.

The following topics are covered in this section:

- GUI Support for Specifying Multiple Bin Files in Different Directories
- GUI Support for Specifying Multiple Bin Files in the Same Directory
- Batch Mode Support for Specifying Multiple Bin Files

GUI Support for Specifying Multiple Bin Files in Different Directories

To specify multiple bin files located in different directories, follow these steps:

© 2020 All Rights Reserved.

1. Open the IR/EM Results form.
2. Click the + button is provided for the *State Directory/Results File* field. This button is provided on both the IR and EM tabs of the IR/EM Results form.
3. The *Select MMSIM Results File* pop-up window opens. Use this to add multiple bin files for EMIR analysis. For details, see Viewing the IR Drop Analysis Results.

Figure 8-6 Specifying Multiple Simulation Result Files in Different Directories

GUI Support for Specifying Multiple Bin Files in the Same Directory

To specify multiple bin files located in the same directory, follow these steps:

1. Open the IR/EM Results form.
2. Click the browser button next to the *State Directory/Results File* field. The *Select MMSIM Results File* pop-up window opens.
3. Open the directory containing all the bin files. In the *Files of type* field, only the bin files will be visible. This is shown below.
4. Any sub-directories in this directory will also be visible and can be selected.
5. Select all the bin files that are required and click *Open*.

© 2020 All Rights Reserved.

Figure 8-7 Specifying Multiple Bin Files in the same Directory

Note: In the above flow, the + button located next to the *State Directory/Results File* field on the IR tab of the IR/EM Results form is not needed for selecting multiple bin files.

Batch Mode Support for Specifying Multiple Bin Files

In batch mode, different options of the `vfibatch` command are used to specify multiple bin files in different scenarios. The following use models are supported:

- **Specifying multiple bin files that contain simulation results for different nets from the same testbench**. This is useful in large designs, where simulation is run separately for different nets of the same testbench to improve the runtime. In such cases, you can create `emir_bin` for different nets in different bin files. In batch mode, the `-db` option of the `vfibatch` command is used to generate reports in this scenario.

  Consider an example where separate bin files are created for nets, `VDD` and `VSS`. The `emir.conf` files for the two nets will be as follows:

```
emir.conf.vdd net name=[X0.VDD] analysis=[iavg imax  sigvmax vmax] emirutil
techfile = qrcTechfile layermapfile = contactmapfile solver method=[iterated]
```

```
emir.conf.vss net name=[X0.VSS] analysis=[iavg imax  sigvmax vmax] emirutil
techfile = qrcTechfile layermapfile = contactmapfile solver method=[iterated]
```


© 2020
All Rights Reserved.

For details on how the EMIR analysis reports are generated for the above scenario in batch mode, see Reports Generated for Multiple Bin Files for Different Nets from the Same Testbench.

- **Specifying multiple bin files that contain simulation results from different testbenches for the same design**. You can specify multiple bin files that are located either in the same directory or in different directories. In batch mode, the `-db_path` and `-db_multi_path` options of the `vfibatch` command are used to generate reports in these scenarios, respectively. For details on how the EMIR analysis reports are generated for these scenarios in batch mode, see Reports Generated for Multiple Bin Files from Different Testbenches in a Design.

Viewing the IR Drop Analysis Results

To plot EMIR results in Voltus-Fi-XL, ensure the following:

- The `emir` mode is enabled while performing simulation.
- The simulation result file is available in the output directory of the simulation results. The naming convention of the simulation result file is, `*.emir#_bin`. For example, your result file could be named, `xps.emir0_bin`.

Follow these steps to view the IR drop analysis results.

- In the Voltus-Fi-XL console, choose *IR/EM Analysis – Rail Analysis Results.* The IR/EM Results form opens. This form is shown below.

  The IR/EM Results form has the following tabs: *IR, EM, Structural Analysis*, *LRP Browser, Pin2PinR*, and *What-If.*

  Click the IR tab to plot the IR drop analysis results.

© 2020 All Rights Reserved.

Figure 8-8 IR/EM Results Form – IR Tab

On this tab, perform the following steps:

- In the *Results* group box, specify the *State Directory* / *Results File* that stores the results of the simulation. The naming convention of the result file is `*.emir#_bin`.

© 2020 All Rights Reserved.

- Select the + button to specify multiple result files or bin files. For details, see Specifying Multiple Simulation Result Files for EMIR Analysis.
- Specify the *Shrink factor*, if any, by which the xDSPF was shrunk. This is used for flows where the xDSPF is generated with a shrunk technology but the layout remains on the original technology. For example, if the original xDSPF was shrunk by 80%, the shrink factor should be specified as 0.8. The shrink factor value is less than 1.
- Click *Customize Display* to select the nets for which you want to view the display. The Customize Display pop-up window opens. This helps save time as results are loaded only for the selected nets and not for all nets. You can also specify the *IR Threshold* value in this window. This is shown below.

Figure 8-9 Customize Display Form

The *Customize Display* button is enabled only before the results are loaded. It is disabled once the results are loaded.

- Specify either the *QRC Run* or the *DFII Layermap*. When either of these options is specified, the plots displayed in the layout show solid shape highlighting. For more information, see Solid Shape Highlighting.
  - Select *QRC Run.* Two text fields with browse buttons appear in the form. Click on the Browse buttons to open the pop-up windows, *Select QRC Run Directory* and

© 2020 All Rights Reserved.

Select QRC Run Name to specify the path to the QRC *Run Directory* and the QRC *Run Name*, respectively.

The display now generated will be based on the shape database generated by Quantus.

Note: When you switch between the IR and EM tabs, the status of the *QRC Run* radio button is maintained. This means that the Quantus run selections, run directory and run name fields, made on one tab remain populated for the other tab even after clearing the analysis results.

For more information about the setup requirements for the Quantus run flow, see Solid Shape Highlighting.

- Select the DFII layer map file, which is the APS/XPS-to-DFII layer map file. You can create a new layer map file or edit an existing file, using the edit button provided next to the field. This layer map file is the same file that is used in structural analysis. For more information about the file format and description, see DFII Layer Map File in the “FIle Formats” chapter.

  You can also specify this file by using the `vfiDfiiLayerMapFile` environment variable in either the `.cdsenv` or the `.cdsinit` file. For more information, refer to the Environment Variables chapter.

  The following considerations apply to the DFII layer map flow:

  - When the DFII layer map file is specified, the visibility of the DFII layers in the layout is in synch with the layers selected in the *Layers* group box or the layer selection window of the Display form.
  - When the correct mapping in the DFII layer map file is either not specified or if some layers are missing in this file then the resistors or nodes of those layers

© 2020 All Rights Reserved.

will not be highlighted in the layout because there will be no shapes attached to them.

- When this file is not specified, the pop-up-window shown below opens. If you click *Yes*, the plots are displayed using stick diagrams

- For the DFII layer map flow, click *Finer Gradient* to view the finer gradient for the selected layers. When you click this button, the *Finer Gradient Layer Selection* pop-up window opens. This is shown in below figure.

Figure 8-10 Selecting Layers for Viewing Finer Gradient

In this window, all the layers are selected by default. Click *Deselect All* and then select specific PGDB layers for which you want to view finer gradient and click *OK*. For details, see Displaying Finer Color Gradient for Selected Layers in IR and EM Plots.

- Click *Load Results* to load the results. You can click *Clear Results* to clear the results.
- Click *Power-up Report* to view additional information about power gates. This feature is only used in power-gated designs. If this report is not available, the software provides the following information in the *Input not available* pop-up window.

© 2020 All Rights Reserved.

Figure 8-11 Input not Available for Power-Up Report

For more information, see Viewing the Power-Up Summary Report.

- Once the results are loaded, the *Rail Analysis* cyclic field in the *Net Plot* group box becomes enabled. From the list, select the type of plot you want to view. The following options are available:
  - **IR – IR Drop**: analyzes and reports voltage drop
  - **IRAVG – IR Avg Drop**: analyzes and reports average voltage drop
  - **RC – Resistor Current**: analyzes and reports peak resistor currents
  - **IV – Transistor Voltage**: analyzes and reports the voltage drop at device nodes, also known as tap nodes
  - **TCRMS – RMS Tap Current**: analyzes and reports RMS tap currents
  - **TCAVG – Average Tap Current**: analyzes and reports average tap currents
  - **TC – Peak Tap Current**: analyzes and reports peak tap currents
  - **POWER – Resistor Power*:*** analyzes and reports resistor power, which is the value of current through a resistor multiplied by the voltage across the resistor
  - **PI – Powergate current**: analyzes and displays the currents through the resistors when power gates are connected
  - **PV – Powergate voltage**: analyzes and displays the IR drop across power-switch instances. This plot can be used to analyze and debug regions of high IR drop inside the power-gated block of the design
  - **RCAVG – Average Resistor Current**: analyzes and reports average resistor currents
  - **RCRMS – RMS Resistor Current**: analyzes and reports RMS resistor currents
  - **REffective – Effective Resistance**: analyzes and displays the pin to node resistances

  Note: For details of the above plots, see “Types of IR Drop Analysis Plots”.

© 2020 All Rights Reserved.

- Click *Power Nets* to list all power nets in the list box.
- Click *Signal Net*s to list all signal nets in the list box.
- Click *All Nets* to list all power and signal nets in the list box.
- Click *Select All Nets* to select all the nets listed in the table. For example, if you select *Power Nets* and click *Select All Nets*, all power nets listed in the table are selected. The list box lists *Net Type* and the corresponding *Net Name*.

  You can search for specific nets for which you want to plot results by typing the net name in the text field provided in the form. The list box is updated to display information about the specified net.

  Note: The net name is case-sensitive. This is shown in the image below.

- Click *Show Plot* to view the plot on the Virtuoso layout. When you click *Show Plot*, the IR/EM Results form expands to show the available plot display options. You can use these options to customize the displayed plots. These options are explained in detail in the Specifying the Display Options for IR/EM Plots section.

© 2020 All Rights Reserved.

- Click *Clear Plot* to clear the plot for the selected net from the Virtuoso layout. For this, select the net for which you want to clear the plot in the list box and then click *Clear Plot*. In the figure below, the plot for the net, VDD is cleared.

- Specify *IR threshold* for viewing IR drop violations above the specified threshold value. Select *V* or *mV* to specify the threshold voltage in volt and millivolt, respectively. When this value is specified, the *Min - Max* slider range is updated to reflect the specified minimum violation value and the plot is updated in the layout. For example, in the image below, the IR threshold value is specified as 0.25. So, the *Min* value of the slider range is updated to 0.25.

© 2020 All Rights Reserved.

Figure 8-12 Specifying the Threshold Value for Viewing IR Drop Violations

- The *Plot Powergate Nets* option is used for plotting the IR drop analysis results of power-gated designs. In this case, when you click the *Select Nets* option and select a net from the list box, all the switched nets for the selected always-ON net also get selected. The software plots results for all the switched nets together. This option is turned off by default.
- Click *Save* to save the specified settings in a configuration file.
- Click *Load* to automatically load settings saved in a previous run in the IR/EM Results form. For this, you can save the settings in a `.vpsconfig` file. When you click *Load*, the software searches for the `.vpsconfig` file. If the file is available, the settings specified in this file are restored in the form. A sample `.vpsconfig` file is shown in Example 8-1 on page 183.

© 2020 All Rights Reserved.

You can also load the EMIR control file or `emir.config` file using this button. When this file is specified, the technology file and layer map file information provided in this file is used for EMIR analysis. For more information about the EMIR control file, see Simulation Requirements and Setup in the “Data Preparation” chapter.

Figure 8-13 Viewing the IR Plot for All Power Nets

© 2020
All Rights Reserved.

Figure 8-14 Viewing IR Plot for Selected Nets

Video

To view the demonstration on plotting the IR drop analysis results in Voltus-Fi, see Viewing and Querying the IR Drop Analysis Results in Voltus-Fi-XL video.

Access to this video will depend on the availability of a web browser and a Cadence Online Support account.

Tcl Commands:

An example of the set of Tcl commands for loading and printing the IR drop analysis reports is as follows:

© 2020 All Rights Reserved.

```
load_ir_results\
    ./abc/raw.out/xps.emirtap.emir0_bin\
-print_ir_report \
    -net all_power \
    -type ir \
    -filename all_ir.txt\
```

Example 8-1 Sample Configuration File

A sample `.vpsconfig` file is shown below.

```
setvar vps_results_IR
{/icd/epsrd_t1nb_002/babita/xps/demo/design/raw.078/xps.emirtap.emir0_bin}
setvar vps_results_EM
{/icd/epsrd_t1nb_002/babita/xps/demo/design/raw.078/xps.emirtap.emir0_bin}
setvar vps_results_IR_Flow {mmsim}
setvar vps_results_IR_shrink_factor {}
setvar vps_results_EM_shrink_factor {}
setvar vps_results_EM_tech_file
{/icd/epsrd_t1nb_002/babita/xps/demo/qrcTechFile}
setvar vps_results_EM_layermap_file
{/icd/epsrd_t1nb_002/babita/xps/demo/contactmapfile}
setvar vps_results_EM_peak_dc true
setvar vps_results_EM_avg_dc true
setvar vps_results_EM_abs_avg_dc false
setvar vps_results_EM_peak_ac false
setvar vps_results_EM_rms_ac false
```


© 2020 All Rights Reserved.

Displaying and Querying EMIR Results

Voltus-Fi-XL lets you specify the plot display settings for the IR drop and EM plots that are displayed on the Virtuoso layout.

The plot display options include a *Min - Max* slider that lets you customize the range of violations you want to view. A continuous RGB gradient is used to highlight the worst violation regions in the design. The vast range of color options provided by the RGB gradient makes it easier to view a range of violations in the layout. The available plot display options of the IR/EM Results form are shown in Figure 8-17 on page 191.

The details of specifying the display options in this form are provided in Specifying the Display Options for IR/EM Plots section.

In addition to specifying settings for plots displayed in the layout, there are options to query the layout for viewing specific violations. The results of the query are highlighted in the EM/IR tab of Annotation Browser.

The following topics are covered in this section:

- Viewing EMIR Violations in Annotation Browser
- Querying EMIR Results
- Specifying the Display Options for IR/EM Plots

© 2020 All Rights Reserved.

Viewing EMIR Violations in Annotation Browser

Annotation Browser lets you view and manage violation markers generated for the design. When you click *Violation Browser* in the IR/EM Results form, Annotation Browser window appears in the upper-right corner of the Virtuoso layout window, and the *EM/IR* tab opens by default.

The following topics are covered in this section:

- Information Displayed in the EM/IR Tab of Annotation Browser
- Information Displayed in the Description Section of Annotation Browser

Figure 8-15 EM/IR Tab of Annotation Browser

The *EM/IR* tab is used for viewing the violations of IR drop and EM analyses, and it is enabled only when Voltus-Fi-XL is launched. The number of violations displayed in the *EM/IR* tab is based on the number specified in the *Elements count* field in the IR/EM Results form.

© 2020 All Rights Reserved.

Note: The check state of a violation marker is saved in the *EM/IR* tab of Annotation Browser and is reloaded when you start a new Virtuoso session.

Information Displayed in the EM/IR Tab of Annotation Browser

The following information is available in columns in the EM/IR tab of Annotation Browser. The columns that are displayed vary for different analyses types. The complete list of available columns is provided below.

- Value: Provides the violation value.
- Layers: Provides the name of the layer on which the violation is located.
- Short Message: Provides the name of the node or the resistor.
- BBox: Shows the coordinates of the marker bounding box.
- Density: Provides the Current Density value across the resistor.
- Current: Provides the current value across the resistor.
- Current Limit: Provides the current rules as per the specified EM rules.
- Length: Provides the resistor length value.
- Width: Provides the resistor width value.
- Resistance: Provides the resistance value of the resistor.
- Number of Vias: Provides the number of vias in the via array. This is valid only for via layers.
- Needed No. of Vias: Provides the number of vias needed to pass the EM violation. This is valid only for via layers.
- Via Area: Provides the area of the via array.
- Needed Width: Provides the minimum width required for metal layer resistors to pass the EM violation.
- Net: Provides the name of the net associated with the marker.
- Time: Time for peak IR drop in seconds.
- Power Gate Transistor: Provides the name of the power-gate transistor instance. This is valid only for power-gated resistors.

By default, the columns displaying the resistor/node name, layers, and violation values are visible for all types of plots.

© 2020 All Rights Reserved.

Note: The information about columns available for different analyses types for IR plots is provided in the Types of IR Drop Analysis Plots section in this chapter and for EM plots in the Types of EM Analysis Plots section of the “EM Analysis Results” chapter.

You can add or remove columns to customize the information you want to view in Annotation Browser. For this, right-click the column heading, select *Columns* from the list, and then select the columns you want to view in the tab. This is shown below.

Figure 8-16 Selecting Columns in the EM/IR Tab

Information Displayed in the Description Section of Annotation Browser

Detailed information about the selected node is provided in the *Description* section of Annotation Browser. For example, for an RJ JMAX analysis type, the following information is provided:

© 2020 All Rights Reserved.

```
Net = VSS Resistor=ri875 RJ value=3.155 layer=mt1 co-ordinates=(12.017000-
23.025000,11.225000-23.020000) resistance=0.247689 length=0.7875um width=0.24um
needed Width=0.757201um
```

For an IR drop analysis type, the following information is provided:

```
Node=459 VSS#453 IR value=0.00495278 co-ordinates=(58.334999 27.959999) layer=mt1
Net=VSS Time=7.64774e-08
```

You can view the units for corresponding values in the description section.

Note: The entries in Annotation Browser honor the user-specified plot display settings in the display options specified in the IR/EM Results form. Therefore, the results of the query comply with the user-specified plot settings.

For more information about how to use Annotation Browser and manage the different views, see the “Layout XL Assistants” chapter in *Virtuoso Layout Suite XL User Guide*.

© 2020 All Rights Reserved.

Querying EMIR Results

The following layout query features are supported in Voltus-Fi-XL.

- **Viewing worst violations**: You can select an area in the layout and zoom into the worst violation in the selected area by using the *Zoom Select Max Region* button provided in the Display form or the expanded section of the IR/EM Results form. The worst violation in the selected area is highlighted in Annotation Browser. For details, see Viewing Worst Violations.
- **Viewing all violations in selected area**: You can select an area in the layout and retrieve the values of all violations in the selected area by using the *Get Value on Layout* button provided in the Display form. All violations in the selected area are highlighted in Annotation Browser. For details, see Viewing all Violations in Selected Area.
- **Querying Specific Points and Areas in the layout for Resistor Information**: click any point or select an area in the layout to display the resistor information for that point or for all points that lie within that area in the console. This query feature is independent of Annotation Browser. For this, a *Query* check box is provided in the Display form. For details, see Querying Specific Points and Areas in the layout for Resistor Information.

© 2020 All Rights Reserved.

Specifying the Display Options for IR/EM Plots

After you load the IR drop or the EM analysis results on the IR and EM tabs of the IR/EM Results form, select the nets for which you want to view the plots, and click *Show Plot*. The IR/EM Results form expands to provide options for customizing the plot display in the layout.

Use the scroll bar provided at the bottom of the form to view the display options. Click the arrow provided on the top-left corner of the form to hide or unhide the Display form. This is shown in Figure 8-17 on page 191.

The following topics are covered in this section:

- Selecting Layers for Viewing Violations
- Customizing the Range for Viewing Violations
- Customizing the Display Options for Plots
- Viewing and Managing Violations in the layout
  - Viewing Worst Violations
  - Viewing all Violations in Selected Area
- Querying Specific Points and Areas in the layout for Resistor Information
  - Query Box Information for IR Drop Analysis
  - Query Box Information for EM Analysis
  - Query Box Information for SHE Analysis
- Saving the IR Drop Analysis Result Plots and Reports

© 2020 All Rights Reserved.

Figure 8-17 Specifying the Display Options for Plots

Selecting Layers for Viewing Violations

The following options are provided in the *Layers* group box of the Display form for selecting layers:

- Click *AV* or *NV* to select or deselect all layers. Use the check boxes provided against layer names to select or deselect specific layers. By default, all layers are selected.

  When the layer selection is updated, the plot in the layout is updated automatically. The image below shows the layout display of the total violation range for the selected layers for an IR drop analysis type plot.

© 2020 All Rights Reserved.

Note: For all layers defined in the DFII layer map file, the DFII layer names and the corresponding extraction layer names are listed in a tree view. For example, when you click `Metal1`, you will see the extraction layer name, `mt1`. This is shown in below image.

Figure 8-18 Displaying the Total Violation Range for Specified Layers in the layout – IR Analysis Type

© 2020
All Rights Reserved.

Customizing the Range for Viewing Violations

The following options are provided in the Display form for customizing the range for viewing violations.

- *Min* and *Max* sliders are provided for viewing violations that fall within the specified slider range. Move the *Min - Max* sliders up and down. The display highlights are updated according to the new slider positions. The *Min - Max* slider range functionality works differently in the following scenarios:
  - When the *Re-distribute Range* option in the Plot Options group box is unchecked, moving the *Min - Max* sliders only updates the plot display, which means, the violation range outside the color spectrum of this range will be toggled. For example, in Figure 8-17 on page 194, the *Min* and *Max* slider range is changed from showing violations between “green” and “light blue” to “red” and “light blue”.
  - When the *Re-distribute Range* option is checked, moving the *Min - Max* sliders re-distributes the range on the ruler and the plot is updated to reflect this new range. This will also update the *Min* and *Max* values in the Filter Range group box.
- The RGB color spectrum is provided next to the slider for viewing violations in different colors.
- The ruler range of violation values, for all or selected layers, is provided next to the color spectrum.

© 2020 All Rights Reserved.

Figure 8-19 Customizing the Range for Viewing IR Drop Violations for Specified Layers

- In the Plot Options group box, select *Plot out of range* to view the violation values outside the specified *Min - Max* slider range. For example, in the image below, the selected range is between “green” and “light blue” but since the *Plot out of range values* is selected, the violations above and below the slider marks are also displayed in “deep red” and “deep blue”, respectively.

  Note: The *Plot out* of *range values* option is only enabled when the *Re-distribute Range* option is selected.

© 2020 All Rights Reserved.

Figure 8-20 Viewing IR Drop Violations for the Out of Range Values

- Select *Re-distribute Range* to view the complete spectrum, from “deep red” to “deep blue”, for the specified layer selection. For example, in the image below, when this option is not selected, the slider range only shows the color range for the violations in the selected range, from “orange” to “light blue”. However, when this option is selected, the slider re-distributes the range to show the complete color spectrum, from “deep red” to “deep blue”, for the selected range.

© 2020
All Rights Reserved.

Figure 8-21 Re-Distributing the Color Spectrum for the Selected Range

- In the Filter Range group box, specify the *Min* and *Max* values, which are the minimum and maximum values for the filter range. When a plot type is chosen on the IR or EM tab, the filter values are filled in automatically in these fields. To change the default values, you can either type the new Min and Max values and click Apply or enter. The plot is updated automatically.
- Click *Reset Range* to restore the default values of filters in the *Min* and *Max* fields.

Customizing the Display Options for Plots

The following options are used to customize the plot displays:

- Click *Power Grid* to view the power grids. This option is disabled for EM plots.
- Click *Plot* to view the plots.

© 2020 All Rights Reserved.

- Click *Voltage Source* to view the voltage source locations. This option is disabled for EM plots.
- Click *Power gates* to view the power gates in the layout. This option is relevant only for power-gated designs.
- Click the *Sync DFII Layers* option to toggle the display of the DFII layers for the layers selected in the Layers group box. By default, this option is checked, which means the layers are in synch.

Viewing and Managing Violations in the layout

The following options are provided in the *Action* group box of the Display form for viewing and managing violations on the Virtuoso layout:

- Click *Violation Browser* to view and manage the violation markers generated for the current design. The Annotation Browser window opens on the right side of the Virtuoso Layout and displays the top violations. This is shown in Figure 8-17 on page 198.

  Note: For more information, see Viewing EMIR Violations in Annotation Browser.

- Specify the number of violations you want to view in the *Elements count* field. For example, you can choose to view only the top ten violations. The image below shows the top ten violations listed in the EM/IR tab of Annotation Browser and the violation markers in the plot on the Virtuoso layout.

© 2020 All Rights Reserved.

Figure 8-22 Viewing the Top Ten Violations in the layout

If the number of elements specified is 1000 and above, then the list of violations in Annotation Browser is suppressed by default. To view the violations, select *Unsuppress Expansion* from the context menu of Annotation Browser.

Note: For more information about how to use Annotation Browser and manage the different views, see the “Layout XL Assistants” chapter in *Virtuoso Layout Suite XL User Guide*.

© 2020 All Rights Reserved.

Viewing Worst Violations

Click *Zoom Select Max Violation* to zoom into and highlight the maximum violation in the selected area in the layout. For this, first click *Zoom Select Max Violation,* and then select an area in the layout to locate and zoom into the maximum violation in that area. Annotation Browser opens and the node or presistor with the worst violation in the selected area is highlighted in the browser.

In addition, the maximum violation in the selected area is marked on the ruler in the IR/EM Results form with a red arrow. This is shown in the image below.

Figure 8-23 Zooming into the Maximum Violation in the Selected Area

© 2020
All Rights Reserved.

Viewing all Violations in Selected Area

Click *Get Value on Layout* to retrieve the values of violations in the selected area from the layout view. For this, first click *Get Value on Layout* and then select an area in the layout. The violations in the selected area are marked on the ruler with red arrows and are highlighted in Annotation Browser. If there are multiple violations in the selected area, all violations will be marked in Annotation Browser. This is shown in the image below.

Figure 8-24 Retrieving Violation Values from the Layout

© 2020
All Rights Reserved.

Querying Specific Points and Areas in the layout for Resistor Information

Select *Query* to display the resistor information for a point or points within a specified area in the layout. For this, first select *Query* and then use the left mouse button to click any point or to select an area in the layout.

The Layer Selection for Query form opens. In this form, select the layers for which you want to perform the query and click OK. By default, all layers that are selected in the *Layers* group box of the Display form are selected in this form. This form is shown below.

Figure 8-25 Layer Selection for Query Form

Note: The above form opens when multiple layers are selected in the *Layers* group box or the layer selection window.

The result of the query or the resistor information for the selected point or for points in the selected area is displayed in the Query Box.

You can add or remove columns to customize the information you want to view in the Query Box. To do this:

- Right-click a row, select *Columns* from the list, and then select the columns you want to view in the tab.

To rearrange the columns:

- Drag the columns to the required position.

To save the configuration:

- Right-click anywhere in the Query Box and select *Save Config*.

© 2020 All Rights Reserved.

Figure 8-26 Selecting Columns in the Query Box

Different information is displayed in the query box depending upon whether the query is for IR drop, EM analysis, SHE analysis, or RMS hot-spot waiving analysis. This is covered in the following sections:

- Query Box Information for IR Drop Analysis
- Query Box Information for EM Analysis
- Query Box Information for SHE Analysis
- Query Box Information for RMS EM Hot Spot Waiving Analysis

Query Box Information for IR Drop Analysis

The following information is reported in the Query Box columns for IR drop analysis query:

© 2020 All Rights Reserved.

1. *name*: The name of the resistor for which information is being queried
2. *layer*: The name of the layer on which the resistor is located
3. *netName*: The net name of the resistor
4. *node1*: The name of the first node of the resistor
5. *node2*: The name of the second node of the resistor
6. *IRmaxNode1(mA)*: The maximum IR drop on node 1 in milliampere
7. *IRmaxNode2(mA)*: The maximum IR drop on node 2 in milliampere
8. *IRavgNode1(mA)*: The average IR drop on node 1 in milliampere
9. *IRavgNode2(mA)*: The average IR drop on node 2 in milliampere
10. *Ipeak(A)*: The peak current of the resistor in ampere
11. *Iavg(A)*: The average current of the resistor in ampere
12. *Irms(A)*: The RMS current of the resistor in ampere
13. *layerType*: The layer type, metal or via
14. *x(um)*: The x-coordinate of the resistor in micrometer
15. *y(um)*: The y-coordinate of the resistor in micrometer
16. *Reff*: The effective resistance value of the resistor. The effective resistance values are displayed only when effective resistance plot type is selected. For other IR drop analysis plots, *NA* is displayed in the query box column.
17. *Number of Shapes*: The number of shapes in power net layout

A part of a sample query box for querying a selected area for IR drop analysis is shown below.

© 2020 All Rights Reserved.

Figure 8-27 Displaying IR Drop Analysis Query Results for a Selected Area in the Query Box

name	layer	netName	node1	node2	IRmaxNode1 (V)	IRmaxNode2 (V)	IRavgNode1 (V)	IRavgNode2 (V)	Ipeak (A)	Iavg (A)	Irms (A)	layerType	x (um)	y (um)	Reff
rs395	via1	VDD	VDD#33	VDD#70	0.00497933	0.00497793	0.00448571	0.00448572	3.53172e-05	7.19112e-08	3.26424e-06	Metal	15.945	78.480	NA
rg7	mt3	VDD	VDD#82	VDD#83	0.00416405	0.00449885	0.00384609	0.0041636	0.00132807	0.00125874	0.00125881	Metal	30.100	84.856	NA
rs396	via1	VDD	VDD#71	VDD#72	0.00582559	0.00559969	0.00544837	0.00522806	0.00130496	0.00125894	0.00125897	Metal	30.095	71.540	NA
rh26	mt2	VDD	VDD#67	VDD#68	0.00485625	0.0049069	0.00448626	0.00448602	9.59088e-05	1.95354e-07	8.86367e-06	Metal	27.411	78.480	NA
ri86	mt1	VDD	VDD#36	VDD#37	0.00500817	0.0050006	0.00448559	0.00448562	1.72314e-05	3.50863e-08	1.59268e-06	Metal	11.535	83.703	NA
ri207	mt1	VDD	VDD#185	VDD#229	0.0064886	0.00664659	0.00609394	0.00624976	0.000123725	0.000119357	0.000119359	Metal	39.280	70.340	NA
ri85	mt1	VDD	VDD#35	VDD#36	0.00500055	0.00500817	0.00448562	0.00448559	3.06103e-05	6.23282e-08	2.82927e-06	Metal	11.535	81.526	NA
ri81	mt1	VDD	VDD#31	VDD#32	0.00497816	0.00498067	0.00448572	0.00448571	3.19126e-05	6.49791e-08	2.94956e-06	Metal	19.081	78.480	NA
rr646	via2_out_capind	VDD	VDD#72	VDD#87	0.00559969	0.00537379	0.00522806	0.00500774	0.00130496	0.00125894	0.00125897	Metal	30.095	71.540	NA
ri92	mt1	VDD	VDD#36	VDD#44	0.00500817	0.00501002	0.00448559	0.00448558	4.78417e-05	9.74145e-08	4.42195e-06	Metal	12.095	82.130	NA
ri84	mt1	VDD	VDD#34	VDD#35	0.00498843	0.00500055	0.00448567	0.00448562	1.63642e-05	3.33203e-08	1.5125e-06	Metal	12.791	79.606	NA
ri170	mt1	VDD	VDD#188	VDD#189	0.00647101	0.00638381	0.00607862	0.00599364	0.000154188	0.000148767	0.000148771	Metal	34.980	71.360	NA
ri94	mt1	VDD	VDD#44	VDD#46	0.00501002	0.00501072	0.00448558	0.00448558	1.23683e-05	2.51841e-08	1.14319e-06	Metal	12.655	83.633	NA
ri178	mt1	VDD	VDD#195	VDD#196	0.00596731	0.00611754	0.00558612	0.0057324	0.000241788	0.000233302	0.000233309	Metal	28.400	70.925	NA
ri79	mt1	VDD	VDD#28	VDD#30	0.00500563	0.0049971	0.0044856	0.00448564	3.42883e-05	6.98149e-08	3.16908e-06	Metal	21.335	81.526	NA
ri78	mt1	VDD	VDD#26	VDD#28	0.00500749	0.00500563	0.00448559	0.0044856	4.78388e-05	9.74047e-08	4.42146e-06	Metal	20.775	82.588	NA
ri77	mt1	VDD	VDD#26	VDD#27	0.00500749	0.00501622	0.00448559	0.00448556	2.94809e-05	6.00261e-08	2.72476e-06	Metal	20.215	82.398	NA
ri91	mt1	VDD	VDD#41	VDD#42	0.00499902	0.00500196	0.00448563	0.00448562	1.35551e-05	2.75994e-08	1.25281e-06	Metal	21.337	84.720	NA
ri83	mt1	VDD	VDD#33	VDD#34	0.00497933	0.00498843	0.00448571	0.00448567	3.21691e-05	6.55013e-08	2.97328e-06	Metal	14.996	78.480	NA
ri87	mt1	VDD	VDD#37	VDD#38	0.0050006	0.00499671	0.00448562	0.00448564	5.24138e-06	1.06725e-08	4.84458e-07	Metal	12.666	86.308	NA
ri223	mt1	VDD	VDD#246	VDD#247	0.00687127	0.00687168	0.00647337	0.00647377	9.71944e-06	9.37623e-06	9.37641e-06	Metal	35.840	58.140	NA
ri80	mt1	VDD	VDD#30	VDD#31	0.0049971	0.00497816	0.00448564	0.00448572	2.85923e-05	5.82179e-08	2.64266e-06	Metal	20.340	79.606	NA
ri299	mt1	VDD	VDD#333	VDD#334	0.00707481	0.00708072	0.0066729	0.00667873	1.83465e-06	1.76998e-06	1.77001e-06	Metal	39.280	42.850	NA
ri289	mt1	VDD	VDD#322	VDD#323	0.00705489	0.00712512	0.00665334	0.00672261	2.18023e-05	2.10341e-05	2.10345e-05	Metal	37.560	32.970	NA

Query Box Information for EM Analysis

The following information is reported in the Query Box columns for EM analysis query:

1. *name*: The name of the resistor for which information is being queried.
2. *layer*: The name of the layer on which the resistor is located.
3. *netName*: The net name of the resistor.
4. *node1*: The name of the first node of the resistor.
5. *node2*: The name of the second node of the resistor.
6. *type*: The layer type, metal or via.
7. *current_direction*: The current direction of the resistor.
8. *Iavg_rule*: The EM rule used for average analysis.
9. *Ipeak_rule*: The EM rule used for max/peak analysis.
10. *Irms_rule*: The EM rule used for RMS analysis.
11. *Iacpeak_rule*: The EM rule used for AC peak analysis.
12. *x(um)*: The x-coordinate of the resistor in micrometer.

© 2020 All Rights Reserved.

13. *y(um)*: The y-coordinate of the resistor in micrometer.
14. *w(um)*: The width of the resistor in micrometer.
15. *l(um)*: The path length in micrometer.
16. *Ipeak(A)*: The peak current of the resistor in ampere.
17. *Iavg(A)*: The average current of the resistor in ampere.
18. *Irms(A)*: The RMS current of the resistor in ampere.
19. *Iavg_max(A)*: The maximum average Current Density in ampere.
20. *Ipeak_max(A)*: The maximum peak Current Density in ampere.
21. *Irms_max(A)*: The maximum rms Current Density in ampere.
22. *acpeak_max(A)*: The maximum AC peak Current Density in ampere.
23. *delta_T(C)*: The rise in temperature of the resistor caused due to Joule heating in degree Celsius.
24. *Td(us)*: The total on-time period in microsecond.
25. *r*: The duty ratio.
26. *a(um^2)*: The area of the via in micrometer square. In case of metal layer, the value will be `"-"`.
27. *N*: The number of vias in the via array. In case of metal layer, the value will be `"-"`.

A part of a sample query box for querying a selected area for EM analysis is shown below.

Note: All columns in the EM analysis query are not visible in the query box when it opens. Use the scroll bar to see more information in the query box.

© 2020 All Rights Reserved.

Figure 8-28 Displaying EM Analysis Query Results for a Selected Area in the Query Box

Query Box

	name	layer	netName	node1	node2	type	current_direction	Imax_rule	Iavg_rule	Irms_rule	Iacpeak_rule	x (um)	y (um)	w (um)	l (um)	Iavg (A)	Ipeak (A)	Irms (A)	Iavg_max (A)	Ipeak_max (A)	Irms_r
1	rr650	via...	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	45.025	71.540	0	-1	0.00...	0.0011...	0.0011...	0	0	0
2	rs392	via1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	27.350	87.845	0	-1	1.50...	6.4041...	5.265...	0	0	0
3	rg17	mt3	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	45.020	89.059	1	31.5...	0.00...	0.001...	0.0011...	0	0	0
4	rr645	via...	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	30.100	78.480	0	-1	1.95...	9.590...	8.863...	0	0	0
5	rh43	mt2	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	47.709	78.480	0.26	16.5...	9.414...	0.000...	1.0445...	0	0	0
6	rs400	via1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	45.025	71.540	0	-1	0.00...	0.0011...	0.0011...	0	0	0
7	rs396	via1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	30.095	71.540	0	-1	0.00...	0.001...	0.001...	0	0	0
8	ri123	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	39.280	87.350	0.26	154....	1.62...	1.6591...	1.628...	0	0	0
9	rr646	via...	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	30.095	71.540	0	-1	0.00...	0.001...	0.001...	0	0	0
10	ri98	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	27.631	95.436	0.5	14.5...	1.507...	6.4172...	5.2717...	0	0	0
11	rh25	mt2	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	27.510	88.145	1	0.92	1.50...	6.4076...	5.266...	0	0	0
12	ri104	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	41.000	93.460	0.26	154....	1.50...	1.5378...	1.509...	0	0	0
13	ri348	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	47.570	87.739	0.5	19.6...	1.33...	1.0471...	3.7675...	0	0	0
14	ri170	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	34.980	71.360	0.24	583...	0.00...	0.000...	0.000...	0	0	0
15	ri122	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	39.280	89.820	0.26	154....	1.62...	1.6591...	1.628...	0	0	0
16	ri347	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	47.640	85.625	0.5	19.6...	1.143...	9.658...	3.1619...	0	0	0
17	ri101	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	41.000	83.580	0.26	154....	2.07...	2.1182...	2.078...	0	0	0
18	ri136	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	35.840	83.580	0.26	154....	4.151...	4.228...	4.1518...	0	0	0
19	ri162	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	44.675	71.360	0.6	583...	0.00...	0.000...	0.000...	0	0	0
20	ri157	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	46.160	68.020	0.12	583...	5.19...	5.459...	5.199...	0	0	0
21	ri188	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	44.410	71.299	0.12	583...	0.00...	0.000...	0.000...	0	0	0
22	ri192	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	44.440	60.610	0.12	583...	2.86...	2.9731...	2.867...	0	0	0
23	ri164	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	42.075	71.360	0.24	583...	0.00...	0.000...	0.000...	0	0	0
24	ri309	mt1	VDD	VDD#...	VDD...	Metal	UnKnown	NA	NA	NA	NA	41.860	26.120	0.24	583...	3.61...	3.7438...	3.613...	0	0	0

Query Box Information for SHE Analysis

The following information is reported in the Query Box columns for SHE analysis query in addition to that displayed for EM analysis:

1. *feolT(C)*: The self-heating coupling temperature in degree Celsius.
2. *beolT(C)*: The temperature in degree Celsius caused by Joule heating and self-heating coupling.
3. *Iavg_sh(A)*: The average current of the resistor in ampere while performing SHE analysis.
4. *Iavg_sh_max(A)*: The maximum average Current Density in ampere while performing SHE analysis.

Query Box Information for RMS EM Hot Spot Waiving Analysis

The columns below are displayed when the RMS EM hot spot waiving analysis is enabled, which is a limited-access feature:

Note: The RMS EM hot spot waiving analysis, is a limited-access feature. To use this feature, contact your Cadence representative and explain your usage requirements.

© 2020 All Rights Reserved.

1. *rms_%fail*: The pass or fail percentage for RMS EM analysis
2. *avg_%fail*: The pass or fail percentage for average EM analysis
3. *rms_minW(um)*: The minimum width in micrometer required for the metal layer resistors to pass the RMS EM violation
4. *rmsT(C)*: The temperature of the resistor in degree Celsius caused by its own Joule heating
5. *jouleHeatT(C)*: The temperature of the resistor in degree Celsius caused by its own Joule heating and that of neighboring resistors
6. *Tmetal(C)*: The effective temperature of the resistor in degree Celsius, which also includes the temperature caused by Joule heating.
7. *PG_wire*: Indicates whether or not supply net exists in the allowed range of the resistor
8. *rms_waived*: Indicates whether or not RMS violation is waived

Note: The Query Box displays results for the first thousand entries at one time. If your query results in more than 1000 entries, the following pop-up box is displayed:

Important

When the *Query* check box is selected, the left mouse button is used to click a point or select an area in the layout. At this time, you cannot perform any other operation with the left mouse button. To use this button for any other layout operation, deselect the *Query* option.

Highlighting Query Results in the layout

For the query results listed in the Query Box window, you can highlight the resistor in the layout. You can select multiple rows to highlight multiple resistors at the same time.

In the figure below, you can see the two resistors selected in the query box are highlighted in the layout in white. This is shown using a red text box for better visibility.

© 2020 All Rights Reserved.

Figure 8-29 Highlighting Query Results in the layout

Saving the IR Drop Analysis Result Plots and Reports

The following options are provided for saving the plots and reports of IR drop analysis results:

- Click *Save Plot* to save a screen capture of the plot. When you click this button, the Export Image form opens. This form lets you take a screen capture of the plot displayed in the Virtuoso layout and save it to a standard image format. This form is shown below.

© 2020 All Rights Reserved.

Figure 8-30 Export Image Form

You can save images in any of these: BMP, JPG, PNG, PPM, XBM, and XPM. To specify the format, use the appropriate suffix with the filename. Note, however, that PNG is the recommended format; not all features are supported by the other formats.

Note: For more information about the settings that can be specified in this form, see "Screen Capture Functions" in the *Cadence® User Interface SKILL Reference* guide.

- Click *Save Report* to save the report for the analysis type as a text file. When the *Save Report* button is clicked, the Select report file dialog box opens. You can choose an

© 2020 All Rights Reserved.

existing file to save the report, or you can save the report in a new file. The dialog box is shown below.

Note: To save and print the report for all elements, leave the *Elements count* field empty and then select *Save Report*.

© 2020
All Rights Reserved.

- Click *Open Report after Saving* to view the report in the text editor. A sample report for IR drop analysis that opens when you select this option and click *Save Report* is shown below.

/home/anuk/MMSIM_EMIR_WORKSHOP/ir_report (on sjfhw636)

```
IR RESULTS

BINARY FILE          = /home/anuk/MMSIM_EMIR_WORKSHOP/emir_simulation/aps.direct.raw/input.emir0_bin
RESULTS FILE CREATED = 2019-Feb-18 06:04:45 (2019-Feb-18 14:04:45 GMT)
USER SUPPLIED VALUES:
   RESULTS TYPE      =
   TRANSIENT START   = 0
   TRANSIENT STOP    = 1.2e-07
   SIM TEMPERATURE   = 27 C

---------------------- "VDD" NET: Vref = 2.5V -----------------------------
max

Resistor        IR value        Layer           X1       Y1       X2       Y2       Res      Length   Width    Ipeak
                                                                                                (u)      (u)      (mA)
rmw661          0.00713329      mwires          37.56000 38.04000 37.56000 38.04000 0.001    0        1        0
ru608           0.00713329      cont_pdiff      37.56000 38.04000 37.56000 47.92000 15       0        0        8.2601e-05
ru607           0.00713329      cont_pdiff      37.56000 38.04000 37.56000 45.32000 0.789474 0        0        0.00175077
ru606           0.00713329      cont_pdiff      37.56000 38.04000 37.56000 40.38000 0.789474 0        0        0.00939879
ru605           0.00713329      cont_pdiff      37.56000 38.04000 37.56000 35.44000 0.789474 0        0        0.0105702
ru604           0.00713329      cont_pdiff      37.56000 38.04000 37.56000 30.50000 0.789474 0        0        0.10152
ri293           0.00713207      mt1             37.56000 45.32000 37.56000 47.92000 1.73333  2.6      0.12     8.2601e-05
ri292           0.00713193      mt1             37.56000 40.38000 37.56000 45.32000 3.29333  4.94     0.12     0.00183337
ri291           0.00712603      mt1             37.56000 38.04000 37.56000 40.38000 0.039    2.34     0.12     0.0112322
ri290           0.0071256       mt1             37.56000 35.44000 37.56000 38.04000 0.043333 2.6      0.12     0.0112322
ri289           0.00712512      mt1             37.56000 30.50000 37.56000 35.44000 3.29333  4.94     0.12     0.0218023
rmw668          0.00711602      mwires          35.84000 38.04000 35.84000 38.04000 0.001    0        1        0
ru603           0.00711602      cont_pdiff      35.84000 38.04000 35.84000 47.92000 15       0        0        8.26195e-05
ru602           0.00711602      cont_pdiff      35.84000 38.04000 35.84000 45.32000 0.789474 0        0        0.00175117
ru601           0.00711602      cont_pdiff      35.84000 38.04000 35.84000 40.38000 0.789474 0        0        0.0094009
ru600           0.00711602      cont_pdiff      35.84000 38.04000 35.84000 35.44000 0.789474 0        0        0.0105726
ru599           0.00711602      cont_pdiff      35.84000 38.04000 35.84000 30.50000 0.789474 0        0        0.101543
```

Comparing Reports

The Display form provides an option to compare reports for different analysis types that are generated in the GUI and batch mode. For this, click *Compare Report* in the Display form.

© 2020 All Rights Reserved.

Figure 8-31 Comparing Reports

The pop-up window, *Compare Reports*, opens. In this window, you can perform the following tasks:

© 2020
All Rights Reserved.

- Specify the path to the *GUI Report File* generated earlier. When you click the *Browse* button, the Select Existing GUI File pop-up window opens. Select an existing report file and click *Open*.

© 2020
All Rights Reserved.

- Generate a new report for any plot. For this, first select *Generate New Report* and then use the Browse button. The Select GUI report file pop-up window opens. Provide the location and the name of the new report and click *Save.*

- Specify the path to the existing *Batch Report.* Click *OK.*

The vimdiff window opens. In this window, you can see the contents of the two reports next to each other for easy comparison. The figure below displays a vimdiff window comparing two signal violation reports.

© 2020
All Rights Reserved.

Figure 8-32 Viewing Reports in vimdiff

Types of IR Drop Analysis Plots

All rail analysis plots are available in the list in the *Rail Analysis* field in the IR/EM Results form.

Following types of IR drop analysis results plots are available:

- IR, IRAVG – IR Drop Plots

© 2020 All Rights Reserved.

- RC, RCAVG, and RCRMS – Resistor Current Plots
- IV – Transistor Voltage Plot
- TC, TCAVG, TCRMS – Tap Current Plots
- POWER – Resistor Power Plots
- PI – Powergate Current Plot
- PV – Powergate Voltage Plot
- REffective – Effective Resistance Plot

IR, IRAVG – IR Drop Plots

Analyzes and reports voltage drop. The IR drop plot is displayed on the Virtuoso layout. The IR drop data is saved inside the state directory or the simulation result file of the net. The worst case voltage drop can be displayed using the *Violation Browser* check box. The segments of the design with high IR drop can then be debugged.

There are two types of IR plots available in the list – **IR - IR Drop** and **IRAVG - IR Avg Drop**. The IR Drop plot displays the peak voltage drop while the IRAVG plot displays the average voltage drop.

Note: If in the `emir.conf` file in the bin directory, the `report_voltage` variable is set to `true`, the IR plots will display absolute voltage values instead of the IR drop values.

Information Displayed in Annotation Browser for IR Drop Analysis

The EM/IR tab of Annotation Browser displays the following columns for this analysis type:

- Value
- Layers
- Short Message
- Net
- Time

The IR plot is shown in the Displaying and Querying EMIR Results section.

© 2020 All Rights Reserved.

RC, RCAVG, and RCRMS – Resistor Current Plots

Analyze and report resistor currents. After observing IR drop plots, the next step is to understand how the current flows in the design to create the generated IR drop plot. It is important to examine resistor currents (RC) plots because they exhibit behaviors that are not obvious from examining only IR drop plots. The RC plots show current-flow trends that you either might not have expected or current from several power pins reconverging in the middle of the chip to create high currents in some wires.

There are three types of RC plots available in the list – **RC - Resistor Current, RCAVG - Average Resistor Current**, and **RCRMS - RMS Resistor Current**.

Note: The **RC - Resistor Current** plot is available for all analysis types for both IR drop and EM analysis results (both for the `qrcTechFile` and `emDataFile` flow).

Note: The **RCAVG** and **RCRMS** plots are available for all analysis types for viewing the IR drop results but only available in the `qrcTechFile` flow and the EM Only ICT File flow for viewing the EM analysis results. These plots are not available in the `emDataFile` flow.

Information Displayed in Annotation Browser for RC Plot Types

The EM/IR tab of Annotation Browser displays the following columns for RC - Peak, RCAVG, and RCRMS plot types:

- Value
- Layers
- Short Message
- Width
- Length
- Resistance
- Via Area
- Number of Vias

The plots shown below display peak and average resistor currents. The violation range is customized by using the *Min Max* sliders in the IR/EM Results form.

© 2020 All Rights Reserved.

Figure 8-33 RC – Peak Resistor Current Plot

pmos2v
pmos2v
pmos2v

Figure 8-34 Average Resistor Current Plot

IV – Transistor Voltage Plot

Analyzes and reports the voltage drop at device nodes, also known as tap nodes.

Information Displayed in Annotation Browser for Transistor Voltage Plot Type

The EM/IR tab of Annotation Browser displays the following columns for transistor voltage plot type:

- Value
- Layers
- Short Message

© 2020 All Rights Reserved.

- Net
- Time

This plot is shown below.

Figure 8-35 Transistor Voltage Plot

TC, TCAVG, TCRMS – Tap Current Plots

Analyze and report tap currents. These plots show the current distribution inside the transistor. The power-grid library generation program characterizes the distribution of the current inside the transistor and replaces the devices connected to the power grid with current taps (sink). The relative distribution of the current inside the transistor power-grid view library is based on the device width and length.

© 2020 All Rights Reserved.

If an accurate power-grid view is used for the memories during rail analysis, the tap current plots clearly highlight the low current distribution in the transistor region and high power distribution in the sense amplifier and the decoder circuit of the memory.

There are three types of TC plots available in the list – **TC - Peak Tap Current, TCAVG - Average Tap Current**, and **TCRMS - RMS Tap Current**.

Note: The **TC - Peak Tap** Current plot is available for all analysis types for both IR drop and EM analysis results.

Note: The **TCAVG** and **TCRMS** plots are available for all analysis types for viewing the IR drop results but only available in the `qrcTechFile` flow and the EM Only ICT File flow for viewing the EM analysis results. These plots are not available in the `emDataFile` flow.

Information Displayed in Annotation Browser for Tap Current Plot Types

The EM/IR tab of Annotation Browser displays the following columns for TC - Peak, TCAVG, and TCRMS plot types:

- Value
- Layers
- Short Message
- Resistance
- Via Area
- Number of Vias
- Needed Number of Vias
- Net

Note: For details about the information displayed in the columns of Annotation Browser, see Information Displayed in the EM/IR Tab of Annotation Browser.

The plot shown below displays peak tap currents. The violation range is customized by using the *Min Max* sliders in the IR/EM Results form.This plot is shown below.

© 2020 All Rights Reserved.

Figure 8-36 Peak Tap Current Plot for All Power Nets – Displaying Total Violation Range

pmoscap2v
pmos2v
pmos2v
pmos

- Short Message
- Resistance
- Length
- Width
- Via Area
- No. of Vias

Note: For details about the information displayed in the columns of Annotation Browser, see Information Displayed in the EM/IR Tab of Annotation Browser.

The plot shown below displays resistor power plot for the top ten violations.

Figure 8-37 Resistor Power Plot – Displaying Top Ten Violations

© 2020 All Rights Reserved.

Click *Save Report* in the Display form to save the report for this plot. You can open the report for viewing. A sample report is shown below. The reporting for this plot is only supported in the GUI mode.

Figure 8-38 Sample Report for Power - Resistor Power Plot

/home/anuk/MMSIM_EMIR_WORKSHOP/emir_simulation/power_ir_plot (on sjfhua135)

```
POWER RESULTS

BINARY FILE          = /home/anuk/MMSIM_EMIR_WORKSHOP/emir_simulation/aps.direct.raw/input.emir0_bin
RESULTS FILE CREATED = 2019-Feb-18 23:58:39 (2019-Feb-19 07:58:39 GMT)
USER SUPPLIED VALUES:
   RESULTS TYPE      =
   TRANSIENT START   = 0
   TRANSIENT STOP    = 1.2e-07
   SIM TEMPERATURE   = 27 C

----------------------- "VSS" NET: Vref = 0V ------------------------------
POWER
```

Resistor	POWER value	Layer	X1	Y1	X2	Y2	Res	Length (u)	Width (u)
ri1127	3.13952e-06	mt1	63.25500	18.77050	63.25500	23.02000	0.3304	4.2495	1
ri1154	2.86222e-06	mt1	63.25500	18.77050	63.25500	14.64150	0.33032	4.129	1
ri1155	2.58498e-06	mt1	63.25500	14.64150	63.25500	10.51250	0.33032	4.129	1
ri801	2.48841e-06	mt1	11.24000	18.76550	11.22500	23.02000	0.3308	4.2545	1
ri1156	2.30775e-06	mt1	63.25500	10.51250	63.25500	6.38350	0.33032	4.129	1
ri1128	2.16527e-06	mt1	63.25500	23.02000	62.17500	23.03000	0.337497	1.08	0.24
ri803	2.07897e-06	mt1	11.24000	18.76550	11.24000	14.63150	0.33072	4.134	1
ri1157	2.03052e-06	mt1	63.25500	6.38350	63.25500	1.50500	0.39028	4.8785	1
ri865	2.0073e-06	mt1	12.42000	23.02000	12.01750	23.02500	0.135833	0.4075	0.24
rh738	1.94307e-06	mt2	37.55500	1.48000	18.64500	1.48000	0.5673	18.91	2
ri875	1.92246e-06	mt1	12.01750	23.02500	11.22500	23.02000	0.247689	0.7875	0.24
rs1223	1.88988e-06	via1	15.63000	73.21000	15.63000	73.21000	0.093333	0	0
ri1129	1.80084e-06	mt1	62.17500	23.03000	61.28750	23.02000	0.288141	0.8875	0.24
rs1251	1.70296e-06	via1	63.25500	1.50500	63.25500	1.50500	0.093333	0	0
ri804	1.66962e-06	mt1	11.24000	14.63150	11.24000	10.49750	0.33072	4.134	1
rh731	1.62463e-06	mt2	63.25500	1.50500	59.45750	1.50000	0.113925	3.7975	2
rh732	1.52901e-06	mt2	59.45750	1.50000	55.66000	1.48000	0.113925	3.7975	2
rh733	1.4334e-06	mt2	55.66000	1.48000	55.24900	1.50000	0.01233	0.411	2
rh734	1.42305e-06	mt2	55.24900	1.50000	50.92800	1.50000	0.12963	4.321	2
ri1130	1.35973e-06	mt1	61.28750	23.02000	60.89500	23.03000	0.123141	0.3925	0.24
rh735	1.31425e-06	mt2	50.92800	1.50000	46.60700	1.50000	0.12963	4.321	2
ri805	1.26027e-06	mt1	11.24000	10.49750	11.24000	6.36350	0.33072	4.134	1

PI – Powergate Current Plot

This plot is applicable only to power-gated design and analysis.

This plot displays the currents across resistors when power gates are connected.This plot is used to analyze whether the current through power gates exceeds the saturation current or not, and if it does, to debug such instances in the design. The saturation current and the on-resistance of the power switches are characterized and stored inside the power-grid library of the power-gate transistor.

© 2020 All Rights Reserved.

Information Displayed in Annotation Browser for Powergate Plot Types

The EM/IR tab of Annotation Browser displays the following columns for both, powergate current and powergate voltage plot types:

- Value
- Layers
- Short Message
- Resistance
- Power Gate Transistor

This PI plot is shown below. In this plot, the resistor with the maximum current violation is zoomed into using the *Zoom Select Max Violation* option in the IR/EM Results form.

© 2020 All Rights Reserved.

Figure 8-39 Powergate Current Plot

A sample PI report is shown below:

© 2020 All Rights Reserved.

```
pi RESULTS

BINARY FILE = /vols/ssvrd_t1_010/nanduc/VPS/CCRs/ruchin_pg_case/vps_pg_design/header_vpsml/output.pg/xps.emir0_bin
RESULTS FILE CREATED = 2013-Dec-03 23:17:20 (2013-Dec-04 07:17:20 GMT)
USER SUPPLIED VALUES:
   RESULTS TYPE       = TRANSIENT
   TRANSIENT START    = 0
   TRANSIENT STOP     = 3e-08
   SIM TEMPERATURE    = 25 C
|
----------------------- "VDDG" NET: Vref = 1.200000V -------------------------------
max

Resistor   Idsat   pi Value        Power Gate Transistor   X1          Y1          X2          Y2          Resistance
rk167      NA      4.906e-06       XM16                    3.179000    4.335000    3.210000    4.335000    3.75
rk166      NA      4.90562e-06     XM15                    3.179000    2.125000    3.210000    2.125000    3.75
rk162      NA      1.01019e-06     XM10                    2.114000    1.260000    2.145000    1.260000    3.75
rj159      NA      1.00065e-06     XM8                     1.759000    3.855000    1.790000    3.855000    3.75
rk164      NA      9.62062e-07     XM13                    2.544000    0.395000    2.575000    0.395000    3.75
rj157      NA      9.5882e-07      XM6                     1.049000    3.855000    1.080000    3.855000    3.75
rj158      NA      9.58674e-07     XM7                     1.759000    2.555000    1.790000    2.555000    3.75
rk165      NA      9.58373e-07     XM14                    3.179000    1.260000    3.210000    1.260000    3.75
rk163      NA      4.82571e-07     XM11                    2.469000    2.125000    2.500000    2.125000    3.75
rj156      NA      4.7966e-07      XM5                     1.049000    1.260000    1.080000    1.260000    3.75
```

PV – Powergate Voltage Plot

This plot displays the IR drop across power-switch instances. This plot can be used to analyze and debug the regions of IR drop inside the power-gated block. For example, if the IR drop, across power gate is already high, the IR drop inside the power-gated block will be much higher. In this case, the power-gate placement will have to be refined or more power gates will have to be added to resolve the IR drop problem.

This plot is shown below.

© 2020 All Rights Reserved.

Figure 8-40 Powergate Voltage Plot

A sample PV report is shown below:

© 2020 All Rights Reserved.

```
pv RESULTS

BINARY FILE | = /vols/ssvrd_t1_010/nanduc/VPS/CCRs/ruchin_pg_case/vps_pg_design/output.pg/xps.emir0_bin
RESULTS FILE CREATED = 2013-Dec-03 23:18:03 (2013-Dec-04 07:18:03 GMT)
USER SUPPLIED VALUES:
   RESULTS TYPE       = TRANSIENT
   TRANSIENT START    = 0
   TRANSIENT STOP     = 3e-08
   SIM TEMPERATURE    = 25 C

----------------------- "VDDG" NET: Vref = 1.200000V -------------------------------
max

Resistor   Idsat  pv Value   Power Gate Transistor X1         Y1         X2         Y2         Resistance
rj159      NA     0.51355    XM8                   1.759000   3.855000   1.790000   3.855000   3.75
rj157      NA     0.513548   XM6                   1.049000   3.855000   1.080000   3.855000   3.75
rk167      NA     0.513542   XM16                  3.179000   4.335000   3.210000   4.335000   3.75
rj158      NA     0.513535   XM7                   1.759000   2.555000   1.790000   2.555000   3.75
rk164      NA     0.51353    XM13                  2.544000   0.395000   2.575000   0.395000   3.75
rj156      NA     0.513523   XM5                   1.049000   1.260000   1.080000   1.260000   3.75
rk162      NA     0.513521   XM10                  2.114000   1.260000   2.145000   1.260000   3.75
rk163      NA     0.51352    XM11                  2.469000   2.125000   2.500000   2.125000   3.75
rk165      NA     0.513462   XM14                  3.179000   1.260000   3.210000   1.260000   3.75
rk166      NA     0.513273   XM15                  3.179000   2.125000   3.210000   2.125000   3.75
```

REffective – Effective Resistance Plot

Uses the data generated by the static power grid solver (SPGS) to display the pin-to-node resistances on the Virtuoso layout. For more information about how resistances are calculated and reported by the SPGS, see the Static Power Grid Solver section.

In this plot type, you can also plot the least-resistive path for a net. For details, see Plotting the Least-Resistive Path.

The effective resistance plot for all nets is shown below.

© 2020 All Rights Reserved.

Figure 8-41 REffective Plot for All Nets

Plotting the Least-Resistive Path

The least-resistive path (LRP) plot lets you identify weakly connected instances in the design during early stages of power planning. The resistance for an instance pin is calculated as the total resistance along the least resistance path. If an instance has multiple power pins connected to the power grid, the LRP plot uses the pin with the worst (highest) resistance value to plot the instance-based data. This plot highlights the current path for the selected instance to the voltage source. A long LRP usually results in high resistance and potentially high voltage drop.

In addition to identifying and displaying the worst IR drop violations, the LRP feature in Voltus-Fi-XL lets you plot LRP on demand for any node. You can view the LRP for any high IR drop node by selecting it in the layout. There are two options provided in the GUI, *Get Layout*

© 2020 All Rights Reserved.

Node and *Get Marker Node*, to let you select a node on the GUI and view its LRP and to select an object in Annotation Browser and view the LRP for the node on the selected marker. For details, see The LRP Browser and Displaying LRP for the Node on the Selected Marker, respectively.

Batch Mode Support for LRP Analysis

The following batch commands is used for loading and printing RLRP analysis reports in Voltus-Fi-XL:

- load_ir_results

  Loads the IR drop analysis results

- print_rlrp_report

  Prints the RLRP analysis report

Batch Mode Support for LRP Analysis

The following batch commands is used for loading and printing RLRP analysis reports in Voltus-Fi-XL:

- load_ir_results

  Loads the IR drop analysis results

- print_rlrp_report

  Prints the RLRP analysis report

Tcl Commands:

An example of the set of Tcl commands for loading and printing the LRP analysis reports is as follows:

```
load_ir_results\
    ./abc/raw.out/xps.emirtap.emir0_bin\
print_rlrp_report \
    -net TVDD \
    -filename RLRP.rpt \
    -tap MavD7_1_unmatched#d \
```

The detailed description, syntax and examples of how to use these commands are provided in the load_ir_results section of the “Batch Mode Execution” chapter.

© 2020 All Rights Reserved.

GUI Support for LRP Analysis

The use model for LRP analysis is detailed below. To view the LRP plot for a net, perform the following steps after loading the IR drop analysis results:

- Click *Select Nets* to select the net for which you want to view the LRP plot.

  Note: The LRP plot feature works for one net at a time. It can only be used in the "*Select Nets*" mode and not in the "*All Nets*" mode.

- Click *Show Plot*. The LRP Browser tab in the IR/EM Results form gets enabled. This is shown below.

© 2020 All Rights Reserved.

Figure 8-42 IR/EM Results Form – Plotting the Least-Resistive Path

The LRP Browser

The LRP Browser tab lets you specify nets, layers, and nodes for which you want to view the LRP plots. The various options provided for this are detailed below.

Voltus-Fi-XL provides support in the form of tips about the information to be filled out in the various fields in the LRP Browser form. This tooltip appears when you hover over a field in the form. This is shown below.

© 2020 All Rights Reserved.

Figure 8-43 Tooltips for LRP Browser Tab

To start, click the LRP Browser tab of the IR/EM Results form. The form shown below opens.

© 2020
All Rights Reserved.

Figure 8-44 LRP Browser Tab of the IR/EM Results Form

There are two group boxes in this tab, the LRP Paths and Resistance Path.

- In the LRP Paths group box, specify the nets for which you want to perform the LRP analysis in the *Select Nets* field. The net selected in the IR tab is selected by default.

© 2020 All Rights Reserved.

- Click *Compute LRP* to compute LRP values for all nodes of the selected net. The information is populated in the Node ID - Node Name and LRP Value columns of the table in the LRP Paths group box.
- Click *Shorted layers* to short specific layers for the LRP analysis. For details, see Shorting Layers for LRP Analysis.
- Click *Get layout Node* to select a node in the layout. Click this button and then click anywhere in the Virtuoso layout window. The node-name text box will print information about the *X*- and *Y*- co-ordinates, and the *Layer* name for the node present nearest to the location of the click. The LRP path of the selected node will be shown in the table below.
- Click *Get Marker Node* to view the LRP values for a node, if present, on a selected marker. For this, enable Annotation Browser, select a marker and click this button. If the marker is associated with a node name, the LRP for the node will be shown in the table below. For details, see Displaying LRP for the Node on the Selected Marker.
- Type the name of the node in the *Node Id or Name* to retrieve the LRP value for the specific node. Node names can be regular expressions, such as `*MPM` shown in the figure above. The LRP table shows all nodes matching the regular expression. This saves time taken to scroll through the list of nodes to find information for a specific node.

  Note: You can type the name of any node in this field, including a node that is not included in the list of violations. In this case, the table in the *Resistance Path* group box shows the information for the node of the selected net and the plot displays the LRP path.
- Specify the *Number of Paths* or the number of violations that you want to view. By default, top 100 violations are listed.
- The table shows the LRP for the selected net to all instances or nodes defined for the net in the *Node ID - Node Name* column and their corresponding LRP values in the *LRP Value* column. For each instance, the table shows the cumulative resistance on the path.
- Select an instance in the list to view its LRP in the layout. When an instance is clicked, the LRP path is highlighted in red in the Virtuoso layout. This is shown in below image.
- In the *Resistance Path* group box of the LRP Browser tab, you can view the list of resistors for a path. For details of the columns in the table, see Resistance Path Group Box Table Information. Select *Auto Zoom* to automatically zoom to the resistor being selected in the path in the layout. The LRP path is highlighted with a white border and the resistor from the path is highlighted in red. This is shown below.

© 2020 All Rights Reserved.

Figure 8-45 Least-Resistive Path for a Resistor Highlighted on the Virtuoso Layout

Note: If the tables in the LRP browser tab do not show any entries, ensure the following:

- The selected net has at least one pin location or voltage-source location
- The selected net has nodes connected to it
- Click *Print Report* to print the LRP values of the specific resistors in a text file. The Select report file pop-up window opens. Specify the name of the report file and click *Save*. The report opens in the console. A sample report is shown in Default Reports Generated for EMIR Analysis.
- Click *Clear Highlight* to clear the LRP highlighted in the layout.

© 2020 All Rights Reserved.

- Check *segmentwise* to view the combined resistance values for a layer in the Resistance Path table. When this option is enabled, all the resistors on a layer, metal or via, will be merged into one segment for the purpose of LRP reporting. In the image below, the resistors, `rg12`, `rg13`, and `rg14` on layer, `mt3` are shown in the Resistor Path table. This is the display when the *segmentwise* option is not selected.

Figure 8-46 Display of Resistor Paths

	Res-Name	LRP Val (Ohm)	V-drop (V)	Layer	Length (u)	Width (u)	Acc. LRP (Ohm	Acc. V-drop (V)	Current (A)	Acc. current (A)	Time (s)
13	rg12	0.032808	0.1093	mt3	0.6395	1.17	0.120308	0.976067	0.0433183	-0.587004	0
14	rg13	0.165539	0.110721	mt3	2.775	1	0.285847	1.08679	0.0433183	-0.543686	0
15	rg14	0.006654	0.117892	mt3	0.1295	1.17	0.292501	1.20468	0.0389955	-0.504691	0

Figure 8-47 Segment-wise Display of Resistor Paths

	Res-Name	LRP Val (Ohm)	V-drop (V)	Layer	Length (u)	Width (u)	Acc. LRP (Ohm	Acc. V-drop (V)	Current (A)	Acc. current (A)	Time (s
2	segment	0.0875	0.10551	via2_o...	0	0	0.0875	0.866766	0.0433183	-0.630323	0
3	segment	1.88097	1.55195	mt3	31.5135	1.17	1.96847	2.41871	0.437306	-0.193016	0
4	segment	0.175	0.190274	via2_o...	0	0	2.14347	2.60899	-0.0388989	-0.231915	0

In the above image, the resistors for the layer, `mt3` are combined to display the resistor path for the segment. This is highlighted in red in the above image.

Resistance Path Group Box Table Information

The Resistance Path table lists all resistors in the LRP path for the node selected in the Node ID - Node Name table. You can click any row to highlight the resistor in the layout window.

A part of the Resistance Path table is shown in the image below. The details of the column information is provided subsequently.

Figure 8-48 Information in the Resistance Path Table

Resistance Path

	Res-Name	LRP Val (Ohm)	V-drop (V)	Layer	Length (u)	Width (u)	Acc. LRP (Ohm	Acc. V-drop (V)	Current (A)	Acc. current (A)	Time (s)	current
15	rg14	0.006654	0.117892	mt3	0.1295	1.17	0.292501	1.20468	0.0389955	-0.504691	0	0.03332
16	rg15	0.248683	0.118152	mt3	4.1445	1	0.541184	1.32283	0.0389955	-0.465695	0	0.03895
17	rg16	0.25224	0.127849	mt3	4.204	1	0.793424	1.45068	0.0389955	-0.4267	0	0.03895

Auto Zoom | Compute LRP | Print Report | Clear Highlight | se

© 2020 All Rights Reserved.

It provides the following information:

- *Res Name*: Name of the resistor on the layer
- *R Val (Ohm)*: The resistance value of the resistor
- *V-Drop (V)*: The voltage drop across the resistor. This value increases as the path moves away from the pin.
- *Layer*: Name of the layer on which the resistor is located
- *Length (u)*: The length of the resistor in micron
- *Width (u)*: The width of the resistor in micron
- *Acc.R Val (ohm)*: The accumulative resistance value along the LRP path. This column displays the total resistance of all the resistors till this point in the path.
- *Acc.V Drop (V)*: The accumulative voltage drop along the LRP. This column displays the total voltage drop of all the resistors till this point in the path.
- *Current (A)*: The current across the resistor
- *Acc.Current (A)*: The accumulative current along the LRP till this point
- *Time*: The simulation time captured from the simulation result file
- *current-density (A/u)*: The Current Density across the resistor
- *Location (u)*: The location of the resistor
- *Voltage diff (V)*: The voltage difference at this resistor. It is the difference between the voltage drop between this resistor and the previous resistor.

Shorting Layers for LRP Analysis

While computing LRP for a node or instance, you can short specific layers. When you do this, all resistance on the specified layer is shorted, that means, the LRP display shows a value of 0 for all resistors on the shorted layer. This is shown in below images.

When you click *Shorted layers*, the *Filters* pop-up window opens. In this window, select the layer you want to short. You can short multiple layers.

For example, in the image below, layer `Metal2` is shorted.

© 2020 All Rights Reserved.

Figure 8-49 Specifying the layers to be shorted for the LRP Analysis

When a layer is shorted, the *Layer* information in the Resistance Path group box shows a value of 0 in the *R val (Ohm)* column for all resistors on that layer. For example, in the image below, the LRP value for resistor, `rh740` on layer, `m2` is 0.

© 2020 All Rights Reserved.

Figure 8-50 LRP Value for Resistors on the Shorted Layer

Displaying LRP for the Node on the Selected Marker

You can display the LRP for a node on the selected marker in Annotation Browser. When you select a violation marker in the Browser, it is highlighted in the layout. If the marker is associated with a node name, you can view the LRP for the node. This is particularly useful when either REffective or the transistor voltage plot is being shown and you want to see the LRP for the selected marker node.

© 2020 All Rights Reserved.

The steps are detailed below.

- In the IR tab, select *IV – Transistor Voltage* or *REffective – Effective Resistance* as the plot type and click *Show Plot*.
- In the Display form, select *Violation Browser*. The Annotation Browser window opens in the layout. Select a violation and the violation marker is displayed in the layout.
- In the LRP Browser tab, click *Get Marker Node*. The information for the node – the x- and y- co-ordinates and the layer name – on the marker is populated in the LRP browser tab.
- Click the node name in the LRP table and then click *Compute LRP* to compute the LRP value for the node and to view the least-resistive path for the node in the layout.

  An example is shown below.

© 2020 All Rights Reserved.

Figure 8-51 Displaying LRP for the Node on the Selected Marker

Supporting LRP Analysis in Power-Gated Designs

In Voltus-Fi-XL, you can view the LRP plot for power-gated designs. An always-ON net LRP plot shows LRP path for all nodes in the always-ON net as well as for all nodes connected to the switched net.

The steps performed to select the LRP analysis type in the IR/EM Results form are the same as those described in the previous section. In addition, select the always-ON net. For example, the net TVDD, shown below is an always-ON net.

Note: The net, `VDD!`, shown in below image, is a switched net.

© 2020
All Rights Reserved.

Figure 8-52 Selecting the Always-On Net for Viewing the LRP

Select the LRP Browser Tab. This is shown below.

© 2020
All Rights Reserved.

Figure 8-53 LRP Browser Tab for Power-Gated Designs

In the above form, the *LRP Paths* group box displays the LRP values for the nodes connected to the always-ON net. It also shows the LRP values for the nodes connected to the switched net.

The table in the *Resistance Path* group box shows the resistance for the node selected in the LRP Paths table above.

Note: The resistance value of the resistor, "`rON`", is assumed to be zero ohms.

© 2020 All Rights Reserved.

Video

To view the demonstration on performing the LRP analysis in Voltus-Fi, see Plotting the Least Resistive Path (LRP) video.

Note: Access to this video will depend on the availability of a web browser and a Cadence Online Support account.

LRP Reports

You can create RLRP reports in batch mode by using the print_rlrp_report. You can also generate and view the report by using the *print report* option in the LRP Browser tab in the GUI. You can specify the name of the report file and click *Save*. The report opens in the console. A sample report is shown in Default Reports Generated for EMIR Analysis.

© 2020 All Rights Reserved.

Calculating Effective Resistance between any Two Nodes on a Net

In Voltus-Fi-XL, you can calculate the effective resistance between any two nodes (pins, tap nodes, or subnodes), either on the same net or on different layers of the same net. This feature is supported both in the GUI and in batch mode.

This feature uses the DSPF file, used for Spectre simulation, and the SPGS feature of Spectre. It works only for IR drop analysis. For more information about the SPGS flow, see Static Power Grid Solver.

Note: To run this feature, it is recommended that you use MMSIM/Spectre 15.1.0 ISRx and later versions.

Batch Mode Support for Calculating Effective Resistance

The following batch commands are used for calculating the effective resistance between nodes in Voltus-Fi-XL:

- load_ir_results

  Loads the IR drop analysis results

- pin_2_pin_res

  Calculates the effective resistance between any two nodes

Tcl Commands:

An example of the set of Tcl commands for reporting the effective resistance between two nodes is as follows:

```
load_ir_results \
    ./abc/raw.out/xps.emirtap.emir0_bin

pin_2_pin_res \
    -xdspf abc/raw.out/abc_sample.dspf \
    -x1 83.40 \
    -y1 34.205 \
    -x2 83.40 \
    -y2 35.205
    -layer1 mt1 \
    -layer2 mt2 \
```


© 2020 All Rights Reserved.

```
-net VDD
```

For details, see the load_ir_results in the “Batch Mode Execution” chapter.

GUI Support for Calculating Effective Resistance

In the GUI mode, this feature is enabled through the *Pin2PinR* tab provided in the IR/EM Results form. By default, this tab is disabled. To enable this tab, perform the following steps:

- Select the IR tab.
- Load the IR drop analysis results.
- Select a net and click Show Plot.

The *Pin2PinR* tab is enabled. This is shown below.

© 2020 All Rights Reserved.

Figure 8-54 IR/EM Results form – Pin2PinR Tab

To calculate effective resistance between two nodes on a net, provide the following information on this tab:

- On the *Pin2PinR* tab, the *Net* field is populated with the name of the net selected in the *IR* tab. You can change your selection of nets in the *Layers* group box.
- Specify the *xDSPF File*, which was used for Spectre simulation.
- In the *Node 1* group box, provide the following information for the first node for which resistance is to be calculated:
  - Select the *Layer* on which you want to calculate the node's resistance from the list. By default, this field is populated with the name of the layer selected in the Display

© 2020 All Rights Reserved.

form. If multiple layers are selected in the Display form, all the layer names will appear in the list.

- Specify the *X* and *Y* co-ordinates for the node. You can type the co-ordinates in the X and Y fields. Alternatively, you can get the co-ordinates from the layout. For this, first click *Get X Y* and then select a point on the specified layer in the layout. The co-ordinates of this point are populated in the *X* and *Y* fields. If there is no node on the specified point, the tool will locate the node nearest to the specified point and populate its location in the *X* and *Y* fields.

- In the *Node 2* group box, provide the same information as above for the second node.

  Note: You can choose a node either on the same layer or on a different layer.

- Click *Add Data* to add the location of the two nodes in the table provided in the form. You can add multiple pairs of co-ordinates in the table.

- Click *Run* to calculate the effective resistance between the specified nodes. The table is updated with the effective resistance between the node pairs. This is shown in the figure below.

  Note: After the run, the effective resistance values are also written in the log file, `vfi.log`.

Figure 8-55 Effective Resistance Information Displayed in the Table

	X1:Y1,Layer	X2:Y2,Layer	Reffective	Node1	Node2	Node1 X:Y	Node2 X:Y
1	34.470000:76.045000,mt1-Metal1	24.235000:82.075000,mt2-Metal2	1.44033	VSS#362	VSS#288	36740:76190	26650:82180
2	43.730000:77.305000,mt1-Metal1	25.920000:81.935000,mt2-Metal2	1.4703	VSS#398	VSS#287	42388:76190	27350:82270

- To highlight the effective resistance path in the layout after the run, select the row for which you want to calculate effective resistance in the table.

  This is shown in below image. In the image, the effective resistance path between Node 1 specified on layer, m1 and Node 2 specified on layer, m2 is highlighted in red.

© 2020 All Rights Reserved.

Figure 8-56 Effective Resistance Path Highlighted in the layout

- Select *Auto Zoom* to automatically zoom into the highlighted path in the layout when a row is selected in the table.
- Click *Delete data* to delete the data added in any row in the table.
- Click *Clear Hilight* to clear the highlighted path in the layout.

© 2020
All Rights Reserved.

Video

To view the demonstration performing pin-to-pin resistance check in Voltus-Fi, see Calculating Effective Resistance between any Two Nodes on a Net video.

Note: Access to this video will depend on the availability of a web browser and a Cadence Online Support account.

Generating EMIR Analysis Reports in Voltus-Fi-XL

The following topics are covered in this section:

- Reports Generated Using the EMIR Control File
- Format of IR Reports
- Default Reports Generated for EMIR Analysis
- RLRP Report
- Reports Generated for Multiple Bin Files from Different Testbenches in a Design

Reports Generated Using the EMIR Control File

You can generate EMIR reports in text, HTML, or CSV format by specifying the EMIR control file (`emir.conf`), the EMIR database file, and the output directory. In the EMIR control file, the `report` option of the `emirutil` command is set to `text`, `html`, or `csv` depending upon the required output.

For example, `emirutil report=text` generates a text report. Similarly, `emirutil report=csv` generates a report in a compressed tar file. A CSV report is available on unzipping or untarring this file.

The syntax of the command to generate EMIR reports in text, HTML, or CSV format is as follows:

```
vfibatch -control <confFileName> -db <binFileName> -outdir <outDirName>
```

For example,

```
vfibatch -control emir.conf -db xps.emir0_bin -outdir out1
```

or,

```
emirreport -control emir.conf -db xps.emir0_bin
```

Note: Specifying the output directory is optional.

© 2020 All Rights Reserved.

The syntax of the command to generate EMIR reports using multiple bin files is as follows:

```
vfibatch -control emir.conf.report -db "binFileName1 binFileName2"
```

Important

For details of the supported EMIR control file options that can be specified, see EMIR Control File Options Supported in Voltus-Fi-XL. For details on how to specify variables using the `emirutil` command in the EMIR control file, see Using the emirutil Command in the EMIR Control File in the "Variables" chapter.

Format of IR Reports

An IR report is generated in the following default format depending on which command is run:

- **Node-Based Format**: A node-based report is generated when the `emirreport` command is run. The report is generated independent of from where the `emirreport` binary is picked, Spectre or Voltus-Fi installation.

VOLTAGE-DROP (V)	NETNAME	TIME (s)	LAYER	X (um)	Y (um)
19.729m	MavI2/avI29/avD30_12#s	1.256n	mwires	83.400	34.205

- **Resistor-Based Format**: A resistor-based report is generated when the `vpsbatch` or `vfibatch` command is run. In this case, the report displays the high drop on the nodes connected to the resistor.

Resistor	ir value	Layer	X1	Y1	X2	Y2	Res	Length (u)	Width (u)	Ipeak (mA)
rmw88352	0.0023634	mwires	43.130	17.160	43.130	17.160	0.001	0	1	0

However, you can change the format of the report as follows:

- If the `emirutil extendedreport=true` option is specified, the report is always generated in resistor-based format.
- If the `emirutil node_based_ir=true` option is specified, the report is always generated in node-based format.
- If both the `emirutil extendedreport=true` and `emirutil node_based_ir=true` options are specified, the report is always generated in node-based format.

© 2020 All Rights Reserved.

Default Reports Generated for EMIR Analysis

By default, Voltus-Fi-XL generates the following reports for EMIR analysis:

- IR drop analysis reports with the extension, `#.rpt_ir`
- EM analysis reports with the extension, `#.rpt_em`
- Pin current reports with the extension, `#.rpt_pin`
- Summary report (`summary.rpt`), which includes reports for all analyses (IR drop and EM) performed in the current run

Note: All default reports print the software version number in the header. You can view this in the sample reports shown in the subsequent sections.

The following reports are detailed:

- IR Report Format
- EM Report Format
- Pin Current Report Format
- Summary Report Format

IR Report Format

In the IR drop analysis report file, information is provided in the following format for each net. The type of analysis is mentioned for each net.

```
----------------------- "VDD" NET: Vref = 2.500000V ----------------------------
--
max
Resistor   ir value   Layer   X1   Y1   X2   Y2   Res   Length   Width   Ipeak
```

Sample IR Drop Analysis Report

A part of the sample IR drop analysis report is shown below.

```
VOLTAGE DROP RESULTS
VERSION = v06.17-e021_1

BINARY FILE          =
/home/anuk/MMSIM_EMIR_WORKSHOP/emir_simulation/aps.direct.raw/input.emir0_bin
```


© 2020 All Rights Reserved.

```
RESULTS FILE CREATED = 2015-Dec-16 23:15:18 (2015-Dec-17 07:15:18 GMT)
USER SUPPLIED VALUES:
   RESULTS TYPE      = TRANSIENT
   TRANSIENT START   = 0
   TRANSIENT STOP    = 1.2e-07
   SIM TEMPERATURE   = 27 C
   AVERAGE           = nil
   PEAK              = t
   SIGVMAX           = t
   SIGVAVG           = nil
-------------------- "VDD" NET: Vref = 2.500000V -------------------------------
max
Resistor     ir value        Layer      X1       Y1       X2       Y2

ri293        0.00713207      mt1        37.56    45.32    37.56    47.92
ri292        0.00713193      mt1        37.56    40.38    37.56    45.32
ri291        0.00712603      mt1        37.56    38.04    37.56    40.38
ri290        0.0071256       mt1        37.56    35.44    37.56    38.04
ri289        0.00712512      mt1        37.56    30.5     37.56    35.44
```

EM Report Format

In the EM analysis report file, information is provided in the following format for each net. The type of analysis is mentioned for each net.

```
----------------------- NET "VDD" -------------------------------
avg
Pass/Fail %   Resistor   layer   Current   Width   PathLength I_limit
X1   Y1   X2   Y2   J/JMAX   Res   ViaArea   No of needed width/#via   J_limit
```

Sample EM Analysis Report

The image below shows a part of the EM analysis report.

© 2020 All Rights Reserved.

```
ELECTROMIGRATION ANALYSIS RESULTS

VERSION = v15.10-dev.alankar
BINARY FILE          = xps.emirtap.emir0_bin
TECH FILE          = /vols/ssvrd_t1_012/alankar/618code/vsa/output/dftout/lnx86_64_opt/vps_emirconf_p6iqjjs/qrcTechfile
RESULTS FILE CREATED = 2020-May-20 10:10:24 (2020-May-20 17:10:24 GMT)
USER SUPPLIED VALUES:
   RESULTS TYPE      = tran
   TRANSIENT START   = 0 s
   TRANSIENT STOP    = 1e-08 s
   SIM TEMPERATURE   = 25 C
   SIM HIERARCHY     =
   DSPF SUBCKT NAME  =
   DSPF GLOBAL TEMPERATURE     =
   DSPF OPERATING TEMPERATURE  =
   Tj                = 25 C
   deltaT            = 5
   AVERAGE           = t
   AVERAGE ABS       = t
   AVERAGE POS       = nil
   ACPEAK            = t
   CUSTOM            = nil
   RMS               = t
   PEAK              = t

---------------------- NET "VDD" ------------------------------
avg

Pass/Fail %  Resistor  layer   Current      Width   PathLength I_limit X1       Y1       X2       Y2       J/JMAX     Res      ViaArea No of needed width/#via  J_limit
                                (A)          (um)    (um)       (mA)   (um)     (um)     (um)     (um)                          (nm^2) vias  (um/#)      (A/um)
pass-75.5    rg1974    metal3  1.619m       2       103.149    6.6    0.250    1.750    0.000    0.001    0.245362   0.008084 NA      NA    0.4909             0.0033
pass-79.9    rg1973    metal3  1.329m       2       103.149    6.6    0.550    1.750    0.250    1.750    0.201308   0.00972  NA      NA    0.4028             0.0033
pass-83.4    rg1972    metal3  1.096m       2       103.149    6.6    0.850    1.750    0.550    1.750    0.16608    0.00972  NA      NA    0.3323             0.0033
pass-86.2    rg1971    metal3  911.7u       2       103.149    6.6    1.150    1.750    0.850    1.750    0.138136   0.00972  NA      NA    0.2763             0.0033
pass-86.7    rh2894    metal2  83.69u       0.285   72.807     0.627  41.130   19.435   41.128   20.945   0.133482   0.269506 NA      NA    0.0382             0.0022
pass-87.8    rh2915    metal2  77.77u       0.29    72.807     0.638  41.265   17.435   41.130   16.990   0.121898   0.05647  NA      NA    0.0354             0.0022
pass-88.1    rh2019    metal2  522.5u       2       120.325    4.4    1.750    7.985    1.750    1.750    0.118749   0.202014 NA      NA    0.2376             0.0022
pass-88.4    rg1970    metal3  767.2u       2       103.149    6.6    1.450    1.750    1.150    1.750    0.116248   0.00972  NA      NA    0.2325             0.0033
pass-89.1    rh2018    metal2  478.3u       2       120.325    4.4    1.750    8.285    1.750    7.985    0.108697   0.00972  NA      NA    0.2175             0.0022
pass-89.9    rh2017    metal2  444.5u       2       120.325    4.4    1.750    10.975   1.750    8.285    0.101025   0.087156 NA      NA    0.2022             0.0022
pass-90.1    rg1969    metal3  656.4u       2       103.149    6.6    1.750    1.750    1.450    1.750    0.0994583  0.00972  NA      NA    0.199              0.0033
```

The table below lists the analysis type and the corresponding section heading in the EM report file.

EM analysis type	Section heading in the EM Report File
Average	`avg`
Peak	`max`
RMS	`rms`
Absolute Average	`Absavg`

Generating a Report for All Resistors

You can generate a report to view the resistor information for all resistors in the design. To generate this report, use the `emirutil` command to set `reportAllResistor` to `true` in the EMIR configuration file.

```
emirutil reportAllResistor=true
```

For more information, see EMIR Control File Options Supported in Voltus-Fi-XL in the “Data Preparation” chapter.

© 2020 All Rights Reserved.

A report, with the suffix, "`#.rpt_all`" is generated. The format of the report is the same as that of the "`.#rpt_em`" report.

Pin Current Report Format

In the pin current report file, the `Imax`, `Iavg` and `Irms` currents for pins for each net are reported.

```
-----------------------NET "VDD"------------------------------
Name          Imax          Iavg          Irms
```

Sample Pin Current Report

A part of a sample pin current report is shown below.

```
PIN CURRENT RESULTS

VERSION              = v15.10-dev
BINARY FILE          = /xps.raw/xps.emirtap.emir0_bin
RESULTS FILE CREATED = 2016-Jan-27 01:42:30 (2016-Jan-27 09:42:30
USER SUPPLIED VALUES:
   RESULTS TYPE      = tran
   TRANSIENT START   = 0
   TRANSIENT STOP    = 1e-08

----------------------- NET "VDD" ------------------------------

Name            Imax            Iavg            Irms
                (A)             (A)             (A)
VDD             22.0183 m       1.38584 m       2.75948 m


----------------------- NET "VSS" ------------------------------

Name            Imax            Iavg            Irms
                (A)             (A)             (A)
VSS             -23.4348 m      -1.90594 m      3.04235 m
```

Summary Report Format

In the summary report, there are two tables of information for each net.

The first table provides, in its header, the name of the net, the simulation temperature, and the measurement window or the start and stop time for the analyses. For each net, the table lists the layer name and the corresponding maximum value of each analysis type for that layer. Information for different types of analyses is displayed in different columns.

© 2020 All Rights Reserved.

The format of the first table is as follows:

```
Net name            GND
Sim temperature     125
Measurement window   0 - 5.9e-08

Layer            iavg
```

The second table lists the layer names. For each layer and analysis type, there are two columns of information. The first column lists the number of resistors on which the analysis was run (`analysistype_run_on`) and the second column lists the number of resistors that were ignored in the analysis (`analysistype_num_skip`). The two columns are repeated for different types of analyses.

The format of the second table is as follows:

```
Layer    analysistype1_run_on   analysistype1_num_skip   analysistype2_run_on   a
nalysistype2_num_skip
```

Sample Summary Report

The image below shows a part of the summary report:

```
Net name            GND
Sim temperature     125
Measurement window   0 - 5.9e-08

Layer            vmax            irms            iavg            imax

metal1          0.0137511       0.0541541       1.92388         0.339854
metal2          0.0098085       0.541685        11.6608         0.478398
metal3          0.00587761      0.227175        1.9191          0.445334
metal4          0.00451982      0.109318        0.778421        0.11097
n_odcont         0.022525        0               0.425099        0
n_poly           0.00261004      0               0               0
n_polycont       0.00261902      0               2.87543e-08     0
p_odcont         0.00614363      0               0.146129        0
p_poly           0.00261004      0               0               0

Layer               vmax_run_on vmax_num_skip irms_run_on irms_num_skip
```


© 2020 All Rights Reserved.

```
_____________________________________________________________________
metal1              15537       0              15537       0
metal2              7253        0              7253        0
metal3              2253        0              2253        0
metal4              616         0              616         0
n_odcont            6526        0              0           6526
n_poly              222         0              0           222
n_polycont          80          0              0           80
p_odcont            3291        0              0           3291
p_poly              34          0              0           34
_____________________________________________________________________
Net name            VDD
Sim temperature     125
Measurement window 0 - 5.9e-08
_____________________________________________________________________
Layer            vmax            irms            iavg            imax
_____________________________________________________________________
metal1          0.0288203       0.0742177       1.0235          0.621502
metal2          0.0141003       0.46877         10.0913         1.45709
metal3          0.00556588      0.191977        1.40789         0.184331
metal4          0.00391154      0.113201        0.562096        0.106243
n_odcont        0.00902663      0               0.0742855       0
n_poly          0.00260217      0                  0                  0
```

RLRP Report

RLRP analysis report (`RLRP.rpt`) includes the LRP values of the instances or tap-nodes of the specified net. In the RLRP analysis report file, information is provided in the following format for each net.

```
Shortest path for net <net name> to tap <tap name> : total resistance :
<resistance value>
res-name   coordinates   res   layer   length   width   volt   acc-lrp
acc-volt   node1   node2
```


© 2020 All Rights Reserved.

Sample RLRP Report

```
RLRP RESULTS

BINARY FILE          = ./ts_28hpm_sr_222_hs_f_hc8t_shlas_32x4x4_pgn.raw/ts_28hpm_sr_222_hs_f_hc8t_shlas_32x4x4_pgn.emirtap.emir0_bin
RESULTS FILE CREATED = 2019-Aug-28 23:58:13 (2019-Aug-29 06:58:13 GMT)
USER SUPPLIED VALUES:


Shortest path for net VDD to tap xts_28hpm_sr_222_hs_f_hc8t_shlas_32x4x4_pgn.xcr_0.xsram_1.xcore_xdec_2.xcore_top_1.xmc_top_0_3.xrefrow_top_1.XMC4.XPUR.mxp0:BULK (model NA) : total resistance : 142.732
    res-name        coordinates                          res            layer          length         width          volt           acc-lrp        acc-volt       node1          node2
    --------------------
    r278804         (0.722 75.975 2.13 75.975)           2.95884        metal4         1.40756        0.25           0              2.95884        0              VDD            VDD:773
    r278848         (2.13 75.975 3.785 75.975)           3.47913        metal4         1.655          0.25           0.000340726    6.43797        0.000340726    VDD:773        VDD:788
    r278875         (5.25 75.975 3.785 75.975)           3.07971        metal4         1.465          0.25           0.000720783    9.51768        0.00106151     VDD:786        VDD:788
    r278871         (8.105 75.975 5.25 75.975)           6.00176        metal4         2.855          0.25           0.000925225    15.5194        0.00198673     VDD:784        VDD:786
    r278868         (8.59 75.975 8.105 75.975)           1.01956        metal4         0.485          0.25           0.000957427    16.539         0.00294416     VDD:783        VDD:784
    r278869         (8.59 75.975 10.165 75.975)          3.31095        metal4         1.575          0.25           0.000957427    19.8499        0.00390159     VDD:783        VDD:794
    r278887         (11.795 75.975 10.165 75.975)        3.42657        metal4         1.63           0.25           0.00118552     23.2765        0.00508711     VDD:792        VDD:794
    r278884         (13.245 75.975 11.795 75.975)        3.04818        metal4         1.45           0.25           0.00126793     26.3247        0.00635504     VDD:791        VDD:792
    r278885         (13.245 75.975 16.32 75.975)         6.46424        metal4         3.075          0.25           0.00126793     32.7889        0.00762297     VDD:791        VDD:800
    r278900         (18.78 75.975 16.32 75.975)          5.17139        metal4         2.46           0.25           0.00155367     37.9603        0.00917664     VDD:798        VDD:800
    r278897         (19.76 75.975 18.78 75.975)          2.06015        metal4         0.98           0.25           0.0015942      40.0205        0.0107708      VDD:797        VDD:798
    r278898         (19.76 75.975 21.365 75.975)         3.37402        metal4         1.605          0.25           0.0015942      43.3945        0.012365       VDD:797        VDD:803
    r278793         (22.565 75.975 21.365 75.975)        2.52263        metal4         1.2            0.25           0.0016982      45.9171        0.0140632      VDD:755        VDD:803
    r278791         (22.942 75.975 22.565 75.975)        0.792994       metal4         0.377222       0.25           0.00171187     46.7101        0.0157751      VDD:754        VDD:755
    r278789         (24.35 75.975 22.942 75.975)         2.83144        metal4         1.40778        0.25           0.00176076     49.5416        0.0175359      VDD:753        VDD:754
    r278786         (24.35 75.975 24.35 75.975)          5.06486        via3           0              0              0.00180157     54.6064        0.0193374      VDD:752        VDD:753
    r278788         (24.35 75.975 24.35 75.885)          0.313649       metal3         0.09           0.16           0.00180157     54.9201        0.021139       VDD:752        VDD:12275
    r302880         (24.35 75.255 24.35 75.885)          2.19554        metal3         0.63           0.16           0.00182129     57.1156        0.0229603      VDD:12273      VDD:12275
    r302879         (24.35 75.255 24.35 75.255)          10.1297        via2           0              0              0.00182129     67.2453        0.0247816      VDD:12273      VDD:12274
    r300588         (24.505 75.255 24.35 75.255)         0.929929       metal2         0.155          0.09           0.00183571     68.1752        0.0266173      VDD:10845      VDD:12274
    r279001         (24.522 75.222 24.505 75.255)        16.5005        via1           0              0              0.00183806     84.6757        0.0284554      VDD:838        VDD:10845
    r278997         (24.522 75.222 24.522 75.197)        57.8893        p_odcont       0              0              0.00183806     142.565        0.0302934      VDD:838        VDD:1020
```

Reports Generated for Multiple Bin Files for Different Nets from the Same Testbench

This section describes the use model for analyzing IR/EM results when multiple emir bin files are available for different nets of the same testbench. This is useful in large designs, where simulation is run separately for different nets to improve the runtime.

In batch mode, the following command is used to generate separate IR/EM reports for multiple bin files:

```
vfibatch -control emir.conf.report -db "xps.vdd/xps.raw/xps.emirtap.emir0_bin
xps.vss/xps.raw/xps.emirtap.emir0_bin" -text abc
```

Note: The bin filenames are separated by a space as shown above. The `-text` option is used to specify the name of the result file that will include the results for both nets. In the above example, the following report files will be generated for IR and EM analysis, respectively:

```
abc.rpt_ir
abc.rpt_em
```

Reports Generated for Multiple Bin Files from Different Testbenches in a Design

This section describes the use model for analyzing IR/EM results when multiple emir bin files are available for the same design, but different testbenches. The following reports can be generated for such designs in Voltus-Fi-XL.

© 2020 All Rights Reserved.

- **Simulation deck-based reports**: In this scenario, separate EM and IR reports are generated for each simulation deck or the bin file.
- **Consolidated report for all simulation decks**: In this scenario, consolidated EM and IR reports are generated for all simulation decks or emir bin files.

Creating the Simulation Deck-Based Reports

This use model requires that the multiple `#.emir_bin` files should be available in the same directory. For example, `test.emir0_bin`, `test.emir1_bin`, `test.emir2_bin`, and so on.

In batch mode, the following command is used to generate separate IR/EM reports for multiple bin files:

```
vfibatch -control conf filename -db_path
path_of_directory_containing_emir_bin_files -outdir outDirName
```

For example,

```
vfibatch -control emir.conf -db_path test/test.raw -outdir out
```

This command will generate separate IR/EM reports for each emir bin file.

For example, if there are two bin files, `test.emir0_bin` and `test.emir1_bin`, then the following reports will be generated:

- `out/test.rpt_em` and `out/test.rpt_ir` will be generated corresponding to the `test.emir0_bin` file
- `out/test.rpt_em1` and `out/test.rpt_ir1` will be generated corresponding to `test.emir1_bin` file

Note: The format of these reports is the same as that of the default IR/EM reports generated in case of a single `emir_bin` file run.

Consolidated Reports for All Simulation Decks

A single consolidated EM report and IR report will be generated comprising results from all simulation decks (`#.emir_bin` files). To enable the generation of consolidated reports, set the following:

- In the GUI, open the Variables form available in the EM tab of the IR/EM Results form, and set the EM variable, `consolidatedReport,` to `true`. By default, this variable is set to `false`.

© 2020 All Rights Reserved.

OR

- In batch mode, set the following in the `emir.conf` file:

  ```
  emirutil consolidatedReport=true
  ```

Note: The `vfibatch` command having the option, "`-db_path/-db_multi_path`" with emir variable "`consolidatedReport`" set to "`true`" in the emir configuration file will generate the consolidated reports for both, EM and IR analyses.

By default, when the `consolidatedReport` option is set to `true`, the `worstResult` option is also set to true, which means only the worst violation will be reported.

```
emirutil consolidatedReport=true
emirutil worstResult=true
```

To view all the results in the consolidated report, set the `worstResult` option to `false`.

For details of the above options, see EMIR Control File Options Supported in Voltus-Fi-XL in the “Data Preparation” chapter.

There are two possible scenarios in the creation of consolidated reports:

- Scenario 1, when all emir bin files are in the same directory
- Scenario 2, when the emir bin files are in different directories

These scenarios are detailed below.

Scenario 1: The emir bin Files are in the Same Directory

In this scenario, there are multiple emir bin files in the same directory. For example, `test.emir0_bin`, `test.emir1_bin`, `test.emir2_bin`, and so on.

To generate a consolidated report in this scenario, use the following command in batch mode:

```
vfibatch -control conf_file_name -db_path
path_of_directory_containing_emir_bin_files -outdir outDirName
```

Note: The above command uses the `-db_path` option to specify the path of the directory containing all the bin files.

For example,

```
vfibatch -control emir.conf -db_path test/test.raw -outdir out
```


© 2020 All Rights Reserved.

Scenario 2: The emir bin Files are in Different Directories

To generate a consolidated report in this scenario, use the following command in batch mode:

```
vfibatch -control conf filename -db_multi_path
name_of_file_containing_paths_to_emir_bin_files -outdir
outDirName
```

Note: The above command uses the `-db_multi_path` option to specify the name of the file that contains the paths to the emir bin files containing the simulation results of different testbenches of the same design but in different directories.

For example,

```
vfibatch -control emir.conf -db_multi_path emir_bin_files -outdir
out
```

Where, the file, "`emir_bin_files`" contains the following information:

```
test/test.raw/test.emir0_bin
test1/test1.raw/test1.emir0_bin
test2/test2/raw/test2/emir0_bin
```

A part of a sample consolidated report is shown below.

© 2020 All Rights Reserved.

Figure 8-57 Sample Consolidated EM Report

```
pass-30.0  ra2    via2   8.41499e-07  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir0_bin

pass-100.0  ra2    via2   5.41499e-07  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir1_bin

pass-100.0  ra2    via2   3.41499e-07  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir2_bin


---------------------- NET "VSS" ----------------------------

avg

Pass/Fail  %  Resistor  layer   Current    Width   PathLength I limit X1    Y1    X2    Y2    J/JMAX    Res    ViaArea No of
needed width/#via  J limit emir_bin

                    (mA)      (um)   (um)     (mA)                                (nm^2) vias  (um/#)          (A/um)

fail+40.0  ra4    via2   3.41  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir1_bin

fail+30.0  ra4    via2   3.00  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir0_bin

fail+20.0  ra4    via2   2.70  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir2_bin

pass-30.0  ra5    via2   8.41499e-07  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir0_bin

pass-100.0  ra5    via2   5.41499e-07  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir1_bin

pass-100.0  ra5    via2   3.41499e-07  NA    0.55    2.2   4.78  9.897  4.78  9.834  1.55227e-07 1500   33000  1   1
0.0666667  test.emir2_bin
```


© 2020 All Rights Reserved.

Figure 8-58 Sample Consolidated IR Report

```
VOLTAGE DROP RESULTS

VERSION = v15.10-dev.
RESULTS FILE CREATED = 2015-Oct-28 01:45:33 (2015-Oct-28 08:45:33 GMT)

---------- "VG" NET: Vref = 0.000000V ----------
max

Resistor    ir value    Layer   X1    Y1    X2    Y2    Res   Length  Width  Ipeak   emir_bin
                                                  (u)   (u)   (mA)
ra3         0.00402345  m1      4.78  9.834 4.78  9.734 2000  0.1     0.02   0.000924894  test.emir2_bin
ra3         0.00302345  m1      4.78  9.834 4.78  9.734 2000  0.1     0.02   0.000924894  test.emir1_bin
ra3         0.00202345  m1      4.78  9.834 4.78  9.734 2000  0.1     0.02   0.000924894  test.emir0_bin

ra2         0.00223525  via2    4.78  9.897 4.78  9.834 1500  NA      NA     0.000924894  test.emir0_bin
ra2         0.00203525  via2    4.78  9.897 4.78  9.834 1500  NA      NA     0.000924894  test.emir1_bin
ra2         0.00193525  via2    4.78  9.897 4.78  9.834 1500  NA      NA     0.000924894  test.emir2_bin

---------- "VDD" NET: Vref = 0.900000V ----------
max

Resistor    ir value    Layer   X1    Y1    X2    Y2    Res   Length  Width  Ipeak   emir_bin
                                                  (u)   (u)   (mA)
rx13        0.0965851   m1      4.78  9.834 4.78  9.734 2000  0.1     0.02   0.0214634    test.emir2_bin
rx13        0.0865851   m1      4.78  9.834 4.78  9.734 2000  0.1     0.02   0.0214634    test.emir0_bin
rx13        0.0765851   m1      4.78  9.834 4.78  9.734 2000  0.1     0.02   0.0214634    test.emir1_bin
```

In the above report, the IR results for the resistors `"ra3"` and `"ra2"` for net `VG`, and resistor, `"rx13"` for net, `VDD` from all `emir_bin files` are reported. By default, for consolidated reports, the `emirutil` variable, `worstResult` is set to `true`. Therefore, the report only includes the worst result for each resistor, which is the first row of information for each resistor shown in the above report.

© 2020 All Rights Reserved.

However, for the above report, to view all results, the `worstResult` variable is set to `false`. In this case, the IR results for the resistors are sorted so that the worst case is reported first and best case is reported last.

Reporting a List of Devices from a Subnode

Voltus-Fi-XL supports tracing of all devices or tap points starting from a metal subnode. This feature lets you list the device tap nodes that are nearest to a given subnode on a metal layer. For this, specify the name of the subnode present on any metal layer and the software will list the tap nodes that are nearest to that subnode. The term “nearest” used here refers to the shortest distance of tap nodes from the subnode in terms of the number of resistors.

Note: To use this feature, the names of the subnodes for which you want to trace the tap nodes must be provided.

The output is a report file containing the tap name of each subnode that is specified in the input.

Use Model

This feature is only supported in batch mode. To use this feature, the following commands and variables are used.

- load_ir_results

  Loads the results of the IR drop analysis

- The following variables are provided to specify the tap name patterns to discard while tracing and also to specify the number of taps to be reported while tracing.

  - `subnode_to_tap_path_trace_discard_pattern`
  - `subnode_to_tap_path_trace_number_of_taps_to_report`

  These variables can be set using the `set_variable` command in the command file. For details on how to use the `set_variable` command, see Using the set_variable Command in the Voltus-Fi-XL Command File in the “Variables” chapter.

  - `subnode_to_tap_path_trace_discard_pattern` <*pattern*>

  This variable is used to specify the patterns in the tap names to discard while tracing. You can specify multiple patterns by separating the pattern names with a space. Wildcards are also supported for specifying pattern names. When this variable is specified, all the taps that contain any of the specified patterns will not be reported during tracing.

© 2020 All Rights Reserved.

For example,

```
set_variable subnode_to_tap_path_trace_discard_pattern "MPC"
```

- ❑ `subnode_to_tap_path_trace_number_of_taps_to_report <number>`

  This variable is used to control the number of taps that will be reported by the command. The default value is `1`. If more than one tap is specified, the software will report taps of unique devices, which means that the taps of different fingers of the same device will not be reported.

  For example,

```
set_variable subnode_to_tap_path_trace_number_of_taps_to_report 4
```

- ■ print_subnode_to_tap_path_report

  Prints the report containing the tap node names for the specified subnodes. The syntax of the command is provided below.

```
print_subnode_to_tap_path_report
-net {netname}
-subnode {subnode_name}
-filename {report_file_name}
-never_traverse_top {true | false}
-report_complete_path {true | false}
```

For details of the command parameters, see load_ir_results in the "Batch Mode Execution" chapter.

Sample Command File and Results File

A sample command file and a report file generated from this command file are shown below.

Sample command file

```
load_ir_results emir0_bin.gz
set_variable subnode_to_tap_path_trace_discard_pattern "MPC"
set_variable subnode_to_tap_path_trace_number_of_taps_to_report 6
print_subnode_to_tap_path_report -net VDD -subnode VDD:36919 -filename
subnode_tap_path.rpt
```

Sample results file

```
SUBNODE TO NEAREST TAP PATH RESULTS
BINARY FILE         = emir0_bin.gz
REPORT FILE CREATED = 2016-Sep-24 04:08:45 (2016-Sep-24 11:08:45 GMT)
VDD:36919          23.556  0.68
                   X18_unmatched:s 23.574  0.525
                   XI13/MP1:s  24.03  0.915
                   XI27/MP1:s  24.078  0.915
                   MN6:d  23.622  0.525
```


© 2020 All Rights Reserved.

```
MP0:s  23.793  0.945
MP8:s  23.745  0.945
```


© 2020 All Rights Reserved.

Reporting On-Resistance (RON) Values

Overview

Voltus-Fi-XL reports the RON values for all layers between specified pairs of pins in a design. The software reports the combined resistance contribution of both pins in the pair and the contribution from each pin. The resistance values are reported in ohms.

For this feature, `report_vfi_XL_ron` command has been provided. For details, see report_vfi_XL_ron in the “Batch Mode Execution” chapter.

Generating the RON Report

This flow is supported only in the batch mode. To enable this flow, perform the following steps:

- Load the IR drop/EM analysis results.
- Run the following command:

```
report_vfi_XL_ron
-pin_pair file_name
-output_file file_name
```

Example:

Use the following commands to report RON for IR drop analysis using the pin-pair file, `pin_pair.txt` in the output file, `output.rpt`:

```
load_ir_results …input.raw/input.emir0_bin
report_vfi_XL_ron -pin_pair_file pin_pair.txt -output_file output.rpt
```

Output

The RON report includes the following information:

- Voltage source names and their corresponding terminal/pin names, voltage values, and current values
- Total layer-based wire RON for both pins in the pin-pair combination
- Total layer-based wire RON for each pin in the pin-pair combination

© 2020 All Rights Reserved.

Sample Pin-Pair File

```
VN1 VN2
```

Sample RON Report

A sample RON report is shown below.

© 2020 All Rights Reserved.

Figure 8-59 Sample RON Report

```
Pins and voltage-source used:

Vsource      Terminal Name   Voltage[V]  Current[A]
-------------------------------------------------------
---            P1            31.4        -0.269506
---            P2            0            0
---            P34           32           0.857299
---            VN1           37          -1.36653e-11
---            VN2           37          -4.11738e-11
---            VN3           37           6.66367e-12
-------------------------------------------------------
VN1/VN2
-------------------------------------------------------
Wire & Via   Name   Resistance (Ohms)
-------------------------------------------------------
metale       (ra)   2.071444e-01  (29.597%)
utm          (rb)   1.103522e-01  (15.767%)
viae         (rsb)  8.830884e-03  (1.262%)
m3           (rc)   6.944030e-03  (0.992%)
m2           (rd)   6.357516e-03  (0.908%)
m1           (re)   4.770599e-03  (0.682%)
gv           (rrb)  1.402366e-03  (0.200%)
v3           (rqb)  3.274210e-04  (0.047%)
v2           (rpb)  1.249013e-04  (0.018%)
lwires       (rlw)  2.769806e-07  (0.000%)
-------------------------------------------------------
Total:              3.474145e-01  (49.640%)
-------------------------------------------------------

VN1
-------------------------------------------------------
Wire &    Via Name     Resistance (Ohms)
-------------------------------------------------------
utm          (rb)   2.067393e-02  (2.954%)
metale       (ra)   1.310327e-02  (1.872%)
viae         (rsb)  2.029128e-03  (0.290%)
m3           (rc)   3.128705e-04  (0.045%)
gv           (rrb)  2.632291e-04  (0.038%)
m2           (rd)   2.339333e-04  (0.033%)
m1           (re)   1.717799e-04  (0.025%)
v3           (rqb)  8.433439e-05  (0.012%)
v2           (rpb)  5.306026e-05  (0.008%)
lwires       (rlw)  2.769806e-07  (0.000%)
-------------------------------------------------------
Total:              3.749304e-02  (5.357%)
-------------------------------------------------------

VN2
-------------------------------------------------------
Wire & Via Name     Resistance (Ohms)
-------------------------------------------------------
metale       (ra)   1.940411e-01  (27.725%)
utm          (rb)   8.967832e-02  (12.814%)
viae         (rsb)  6.801756e-03  (0.972%)
m3           (rc)   6.631159e-03  (0.947%)
m2           (rd)   6.123583e-03  (0.875%)
m1           (re)   4.598819e-03  (0.657%)
gv           (rrb)  1.139137e-03  (0.163%)
v3           (rqb)  2.430866e-04  (0.035%)
v2           (rpb)  7.184106e-05  (0.010%)
lwires       (rlw)  0.000000e+00  (0.000%)
-------------------------------------------------------
Total:              3.099214e-01  (44.283%)
-------------------------------------------------------
```


© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

9

EM Analysis Results

- Overview on page 274
- Types of EM Analysis
- Design Resistor EM Current Analysis on page 279
- Viewing the EM Analysis Results on page 279
- Displaying Current Directions for EM Plots on page 289
- Reporting the Direction of Currents in the EM Result File on page 290
- Generating EM Reports in Voltus-Fi-XL on page 291
- Types of EM Analysis Plots on page 292
  - RJ JMAX – Current Density/Limit and J_MAX – Current Density Plots on page 293
  - JAVG – Current Density/Limit and J_AVG Current Density Plots on page 294
  - JABSAVG – Current Density/Limit and J_ABSAVG – Current Density Plots on page 296
  - JACRMS – Current Density/Limit and J_ACRMS – Current Density Plots on page 298
  - JACPEAK – Current Density/Limit and J_ACPEAK – Current Density Plots on page 300

© 2020 All Rights Reserved.

Overview

Voltus™-Fi Custom Power Integrity Solution-XL uses the simulation database generated by Spectre APS/XPS simulators and displays the results of the electromigration (EM) analysis on the Virtuoso layout.

It also generates text and html reports of the analyses and lets you query the analyses results to view specific violations in the layout. This is used to debug the high EM violation regions in the design.

Batch mode support is provided for specifying variables for EM analysis, loading EM analysis results and generating reports. For details of the batch commands used to load and print EM analysis result reports, see load_em_results and print_em_report.

Note: For N3 node, the VPS_NG100 beta license is required for performing EM analysis.

© 2020 All Rights Reserved.

Types of EM Analysis

The following analysis types are available in Voltus-Fi-XL:

Average (avg) Analysis

Calculates the Current Density violations based on the average value of the DC current for metal lines, vias, and contacts.

$$I_{avg} = \left[\left(\int_0^{\tau} I(t)dt \right) / \tau \right]$$

where “T” is time and “`I(t)`” is value of current.

Root Mean Square (rms) Analysis

Calculates the Current Density violations based on the root mean square (RMS) of the AC current for metal lines, vias, and contacts.

$$I_{rms} = \left[\left(\int_0^{\tau} I(t)^2 dt \right) / \tau \right]^{1/2}$$

where “T” is time and “`I(t)`” is value of current.

Peak (max) Analysis

For this analysis, the Current Density violations are calculated based on the maximum DC current for metal lines, vias, and contacts. The $I_{peak(limit)}$ value is specified using the `em_jmax_dc_peak` and `em_jmax_ac_peak` constructs in the ICT file.

AC-Peak (AC Peak) Analysis

For this analysis, the Current Density violations are calculated based on the peak AC current for metal lines, vias, and contacts. It is applied to periodic AC or pulsed DC signals.

© 2020 All Rights Reserved.

$\mathbf{I}_{peak_ac(limit)}=\mathbf{I}_{peak(limit)}/(r)^{1/2}$

where the $I_{peak(limit)}$ is specified in the ICT file

If the construct `em_peak_with_duty_ratio explicit` is specified in the process block of the ICT-EM file, then the limits specified using the `em_jmax_dc_peak` and `em_jmax_ac_peak` constructs are AC-Peak limits. In this case, same limit values are reported for peak (max) and AC_peak analysis. An example of the process section of ICT file is provided below.

```
process "gpdk" {
        em_conductor_unit mA/um
        em_via_area_unit  mA
        em_via_unit mA
        em_peak_with_duty_ratio explicit
}
```

Note: To perform AC-Peak analysis in Voltus-Fi-XL using the ADE L flow, ensure that while setting up simulation, you have enabled the *EM Current Analysis* options *max* and *avg* in the *Analysis* tab of the EM/IR Analysis Setup form.

Different Settings for AC-Peak Analysis

There are different types of AC-peak analyses based on the `dynamicACPeak` variable setting.

When dynamicACPeak is not Set

When the `dynamicACPeak` variable is not set or set to `false`, which is the default value, the values of duty ratio "`r`" is as follows:

$r=\mathbf{I}_{abs\text{-}avg}/\mathbf{I}_{peak}$

Note: The $I_{abs\text{-}avg}$ and I_{peak} values are from simulation results.

The value of "`Td`" is calculated in the following order of priority:

1. If the net frequency is specified either using the `toggleNetFrequency` or `toggleNetFrequencyFile` variable, then

   `Td = r/F`

   where `F` is the net frequency specified using `toggleNetFrequency` or `toggleNetFrequencyFile`

© 2020 All Rights Reserved.

2. If the simulation is run with specified time window and the net frequency is not specified using either `toggleNetFrequency` or `toggleNetFrequencyFile`, then

   ```
   Td = r * time window period
   ```

3. If neither the simulation time window nor the net frequency is specified using either `toggleNetFrequency` or `toggleNetFrequencyFile`, then

   ```
   Td = r * T
   ```

   where `T` is the simulation period

When dynamicACPeak is Set to True

When the `dynamicACPeak` variable is set to `true` or `multiPeak`, the value of duty ratio "`r`" is as follows:

$r = \mathbf{T}_d / \mathbf{T}_{total}$

Where

$\mathbf{T}_{total}$= total transient time

$\mathbf{T}_d$= the time duration in micro second of the total "On Time" period, where `abs(I)>(max(abs(I)/2.0)` during transient analysis.

Voltus-Fi-XL does not calculate $\mathbf{T}_d$ as `max(abs(I)/2.0)` period. It calculates $\mathbf{T}_d$ by measuring the pulse at half-energy level because of performance reasons. The variable, `pwc_threshold` is specified in the EMIR configuration file to define the current threshold for finding the start of the pulse.

When you set `dynamicACPeak = true` or `multiplePeak`, it is important to set '`ipwc`' analysis in the EMIR configuration file when running Spectre. PWC stands for pulse-wise constant, where the actual current waveform from simulation is converted to a pwc waveform. This new waveform is used for 'Td' calculation.

The following command is specified in the configuration file:

```
net analysis=[ipwc] pwc_threshold=1e-6
```

$\mathbf{T}_d$ has different values for the `true/multiPeak` values of `dynamicACPeak`:

- When `dynamicACPeak` is `true`, $\mathbf{T}_d$=the time duration of maximum peak
- When `dynamicACPeak` is `multiPeak`, $\mathbf{T}_d$=sum of time durations of different peaks, $\mathbf{T}_{d1}+\mathbf{T}_{d2}+\mathbf{T}_{dN}$....

© 2020 All Rights Reserved.

Note: `Td` for a particular peak is added to the final `Td` only if the peak current is greater than or equal to percentage threshold of maximum current. This threshold is specified by the `multiplePeakPercentThreshold` variable.

Note: The software replaces the value of "r" with the value of `applyRThreshold` if the value of "r" is < than that of `applyRThreshold`. This is because a small "r" value results in an unreasonable increase in the I_{peak_ac} limit. To avoid this scenario, use the `applyR` keyword to reset the value of "r" in the ICT file or change the value of the `applyRThreshold` variable, either in the Variables form or in batch mode.

Important

Calculating T_d for multiple clock cycles: In case the design is simulated for multiple clock cycles and you want to perform `dynamicACPeak` analysis with `multiplePeak` setting, then it is recommended to set either of the following options in the Variables form of the EM tab of the Voltus-Fi GUI:

- toggleNetFrequency or
- toggleNetFrequencyFile

You can also specify the options in the `emir.conf` file using the `emirutil` command. For details see, EMIR Control File Options Supported in Voltus-Fi-XL.

The effective T_d will be calculated as follows:

```
(Td1+Td2+....+TdN)/number of cycles
```

Avg-Abs (avgabs)

Calculates the Current Density violations based on the average of the absolute current for metal lines, vias, and contacts.

$$I_{abs\text{-}avg} = \left[\left(\int_0^{\tau} |I(t)|\, dt \right) / \tau \right]$$

where "T" is time and "`I(t)`" is value of current.

© 2020 All Rights Reserved.

Design Resistor EM Current Analysis

By default, the EM current analysis is only performed on parasitic resistors defined in the RC section of the xDSPF file. If required, EM analysis for design resistors in the xDSPF instance section can be enabled. This analysis can be applied to primitive resistor elements, or resistors defined with subckt definitions.

```
net design_res_models=[resistor] analysis=[iavg] (primitive resistor)
net design_res_models=[name=rhim] analysis=[iavg] (resistor subckt rhim)
```

For the EM current analysis the parameters `$l`, `$w`, and `$lvl` are expected to be defined for primitive resistor calls, and `l` and `w` for subckt resistor calls in the xDSPF instance section. If the parameter names are differently defined in the xDSPF file—even if only the casing is different—the mapping can be performed in the conf file statement.

```
net design_res_models=[name=rhim_m l=lr w=wr] analysis=[iavg]
net design_res_models=[name=rhim_m l=L w=W] analysis=[iavg]
```

Layer information, and units for `l/w` can be defined in the EMIR conf file statement.

```
net design_res_models=[name=rhim_m layer=mt1 unit=1] analysis=[iavg]
```

The design resistor related EM results are reported in the EM analysis report file.

Viewing the EM Analysis Results

Before plotting EMIR results, ensure that the `emir` mode was enabled while performing the simulation. Also, check that the simulation result file is available in the output directory of the simulation results. The naming convention of the simulation result file is, `*.emir#_bin`. For example, your result file could be named, `xps.emir0_bin`.

Follow these steps to view the EM analysis results.

- In the Voltus-Fi-XL console, choose *IR/EM Analysis – Rail Analysis Results.* The IR/EM Results form opens. Click the EM tab to plot the EM analysis results. This form is shown below.

  Note: Structural analysis is not enabled with EM analysis results.

© 2020 All Rights Reserved.

Figure 9-1 IR/EM Results Form – EM Tab

- In the *Results* group box, specify the shrink factor, if any, by which the xDSPF was shrunk in the *Shrink factor* field.
- Click *Customize Display* to select the nets for which you want to view the display. The Customize Display pop-up window opens. You can only customize the display for selected nets before the results are loaded. The *Customize Display* button is disabled once the results are loaded.

© 2020 All Rights Reserved.

- Specify the *State Directory* / *Results File*, which is a file that stores the results of the APS/XPS simulation. The naming convention of the result file is `*.emir#_bin.`
- Select the + button to specify multiple result files.
- Specify the technology file in the *Tech File* field. The following technology files are supported in the Voltus-Fi-XL flow:
  - The interconnect technology (ICT) file
  - The Quantus technology file `(qrcTechFile)`
  - The EM data file `(emDataFile)`

  Note: For both the ICT file and qrcTechfile flow, Voltus-Fi-XL supports the use of encrypted files.
- Specify the *Layer Map File.* This file is used to map the layer names in the simulation database to the layer names in the technology file specified in the *Tech File* field. This is optional and is only required if the layer names in the simulation database are different from the ones in the technology file. You can also create a new layer map file or edit an existing file, using the edit button provided next to the field. For more information about the layer map file, see Layer Map File in the "File Formats" chapter.

  Note: If the above layer map file is not provided and you click *Load Results*, a pop-up-window with the message, "Map file not specified. EM analysis will run without layer mapping", opens. Click *OK* to proceed.
- Specify the *EM Only ICT File*, which includes information about process and EM models to be used for EM analysis. This is an optional file that can be provided with the `qrcTechFile` flow. When provided, the `process` and `em_model` information will be picked up from the specified `EMOnlyICTFile`.
- Specify either the *QRC Run* or the *DFII Layermap*. When either of these options is specified, the plots displayed on the layout show solid shape highlighting. For more information, see Solid Shape Highlighting.
  - Select *QRC Run.* Two text fields with browse buttons appear in the form. Click on the Browse buttons to open the pop-up windows, *Select QRC Run Directory* and

© 2020 All Rights Reserved.

Select QRC Run Name to specify the path to the QRC *Run Directory* and the QRC *Run Name*, respectively.

The display now generated will be based on the shape database generated by Quantus.

Note: When you switch between the IR and EM tabs, the selected *QRC Run* remains intact. This means that the Quantus run selections, run directory and run name fields, made on one tab remain populated for the other tab even after clearing the analysis results.

For more information about the setup requirements for the Quantus run flow, see Solid Shape Highlighting.

- Specify *DFII Layermap*. This layer map file is the APS/XPS-to-DFII layer map file. When this file is specified, the plots displayed in the layout show solid shape highlighting. You can also create a new layer map file or edit an existing file, using the edit button provided next to the field. This layer map file is the same file that is used in structural analysis. For more information about the file format and description, see the DFII Layer Map File section in the “File Formats” chapter.

You can also specify this file by using the `vfiDfiiLayerMapFile` environment variable either in the `.cdsenv` or the `.cdsinit` file. For more information, refer to the Environment Variables chapter.

Note: If you do not specify this file and click *Load Results*, a pop-up-window opens to verify that you want to proceed without the map file. If you click *Yes*, the plots are drawn using stick display.

- For the DFII layer map flow, click *Finer Gradient* to view the finer gradient for the selected layers. When you click this button, the *Finer Gradient Layer Selection* pop-up window opens. In this window, all the layers are selected by default. Click *Deselect*

© 2020 All Rights Reserved.

All and select the PGDB layers for which you want to view the finer gradient and click *OK*. For details, see Displaying Finer Color Gradient for Selected Layers in IR and EM Plots.

Note: Perform this step before loading the EM Results. After loading the results, when you click *Show Plot*, you can see the finer gradient for the selected layers.

- Click *Variables* to open the Variables form. In this form, you can view and/or edit the variables that are used for EM analysis. Double-click the value of any variable to edit it. For variables that require a path to a file or a directory to be specified, click the *Browse* button next to the variable field and specify the path.

© 2020 All Rights Reserved.

Figure 9-2 The Variables Form

After updating the values, click *OK*. To restore the default values, click *Default*.

You can also set above variables in batch mode by using the `set_variable` command.

Note: For descriptions of the variables that can be set in the above form, see the "Variables" chapter.

© 2020 All Rights Reserved.

- Specify the *Type* of EM analysis to plot the current density violations. The available options are *max*, *avg*, *avgabs*, *AC Peak*, and *rms*. You can select more than one analysis type at a time. By default, *max* and *avg* are selected.
  - **max**: calculates the Current Density violations based on the maximum DC current
  - **avg**: calculates the Current Density violations based on the average value of the DC current
  - **avgabs**: calculates the Current Density violations based on the average value of absolute current
  - **AC Peak**: calculates the Current Density violations based on peak AC current

    Note: This analysis type is not supported in the `emDataFile` flow. It is only supported in the `qrcTechFile` flow. Even if the analysis type is selected in the `emDataFile` flow, the corresponding plot types, JACPEAK and J_ACPEAK, will not be available in the plot types list.
  - **rms**: calculates the Current Density violations based on the root mean square (RMS) value of the AC current

  For details of the above analysis types, see Types of EM Analysis.

  Note: When you specify any of the above analysis types, ensure that the rules for the selected analysis type are available in the technology file specified in the *Tech File* field. If the EM rules are not available, the analysis result file will be empty.

- Click *Load Results* to load the results of the EM Analysis. Once the results are loaded, the list in the *Plot Results* group box is enabled. The list of plots corresponds to the selected analyses types. In addition to the plots specific to the chosen analysis type, Peak Tap Current and Peak Resistor Current plot types are available for all EM analyses.
  - **TC - Peak Tap Current**: analyzes and reports peak tap currents
  - **RC - Peak Resistor Current**: analyzes and reports peak resistor currents

  For details of the above plots, see TC, TCAVG, TCRMS – Tap Current Plots and RC, RCAVG, and RCRMS – Resistor Current Plots in the “IR Drop Analysis Results” chapter.

  You can select from the available plot types. For more information about the different EM type plots, see Types of EM Analysis Plots.

- Click *Clear Results* to clear the results.
- Click *All Power Nets* to view EM plots for all power nets.
- Click *All Signal Nets* to view EM plots for all signal nets.

© 2020 All Rights Reserved.

- Click *Select All Nets* to select all the nets listed in the table. For example, if you select *Power Nets* and click *Select All Nets*, all power nets listed in the table are selected. The list box lists *Net Type* and the corresponding *Net Name*.

  Note: You can search for specific nets for which you want to plot results by typing the net name in the text field provided in the form. The list box is updated to display information about the specified net.

  The net name is case-sensitive. This is shown in the image below.

- Click *Show Only Failed* to plot results only for those nets that failed the EM check. Selecting this option will set the minimum value for the slider to 1.
- Click *Show EM%* to display the EM violations as percentages. The *Min* - *Max* slider range displays the EM violations as a percentage of Current Density and Current Density limit (J/J_{limit}) values. A percentage of 100 and above indicates EM violations. This is shown in image below.

  Note: The *Show Only Failed* and *Show EM%* options are disabled when non-ratio type plots are selected in the *Plot Results* group box. For details of plot types, see Types of EM Analysis Plots.

© 2020 All Rights Reserved.

Figure 9-3 Viewing EM Violations as Percentages

Note: If a net contains multiple pins, the pins on the same net may be shorted together. For more information, see the Port Order Handling section in the “Postlayout Simulation” chapter of the *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

© 2020 All Rights Reserved.

- Click *Show Plot* to view the plot in the layout. When you click *Show Plot*, the IR/EM Results form expands to show the available plot display options. In this form, you can customize the EM analysis plot displays. You can use these options to customize the displayed plots.

  These options are explained in detail in Specifying the Display Options for IR/EM Plots section in the “IR Drop Analysis Results” chapter. The figure below shows the display options in the expanded IR/EM Results form and the EM results plot for all signal nets.

Figure 9-4 Viewing the EM Plot for All Signal Nets

© 2020 All Rights Reserved.

- Click *Clear Plot* to clear the plot for the selected net from the Virtuoso layout. For this, select the net for which you want to clear the plot in the list box and then click *Clear Plot*.

Displaying Current Directions for EM Plots

When you load EM results and click *Show Plot,* the expanded IR/EM Results form opens. In this form, there are options provided for displaying the current direction between nodes and for customizing the size of the arrows showing the direction of the flow of current. In the image above, the *Display* group box shows the following options:

- *Current Direction*: Check this option to enable the display of current directions in the layout for any EM plot type.
- *Display Size*: Specify the size of the symbols that are used for displaying the current directions. You can increase or decrease the size of the symbols based on your requirements. The acceptable range is between 0 and 5. If you specify a value greater than 5, the software will set the display size to the maximum possible size, which is 5.

When the *Current Direction* option is enabled, the plot in the layout shows white highlights for the current direction flow of selected nets. The highlight appears on top of the plot highlights. The current direction plot honors the following user settings:

- Layer selection specified in the *Layers* group box
- The *Select Nets* option in the *Type* group box of the IR/EM Results form, and
- *Show Only Failed* check in the IR/EM Results form

Note: The *Min/Max* slider range setting is not honored by the current direction plot. The plot displays current directions for all nodes.

- For metal layers, arrows are used to indicate the flow of current in the directions; "E", "W", "N", "S", "NE", and so on. For example, if the direction of current is from West to East, the arrow will be horizontal and will point from left-to-right while if the direction of current is South to North, the arrow will be vertical and will point from bottom-to-top.
- For vias, the direction of the current is from layer -> to layer. A cross symbol (X) is used to show currents flowing down, while a diamond-shaped dot shows currents flowing up.

  This is shown in the image below.

© 2020 All Rights Reserved.

Figure 9-5 Current Flow Direction Representation in the layout

Reporting the Direction of Currents in the EM Result File

You can print information about the direction of current flow for each layer in the EM result file. This current direction is reported in the “Current Direction” column in the report.

You can specify the reporting of current direction in batch mode in either of the following ways:

- Specify the following in the `vfibatch` command file:

  ```
  set_variable idirn true
  ```

- Specify the following in the EMIR control file:

  ```
  emirutil idirn=true
  ```


© 2020 All Rights Reserved.

Note: For more information on how to set variables, see “Variables” chapter. For more information about the available EMIR control file options, see EMIR Control File Options Supported in Voltus-Fi-XL in the “Data Preparation” chapter.

The direction of current is printed in the following manner:

- For metal layers, the direction of current is printed as “E”, “W”, “N”, “S”, “NE”, and so on. For example, if the report prints the direction of current as “E”, it means the current is flowing from West to East. If the report prints the direction of current as “N”, it means the current is flowing from South to North.
- For vias, the direction of current is printed as “from layer -> to layer”. For example, in below figure, current direction for `via2` is `metal3->metal2`.

Sample EM Analysis Results File

```
ELECTROMIGRATION ANALYSIS RESULTS

VERSION = 1
BINARY FILE          = ./emir_simulation/aps.direct.raw/input.emir0_bin
RESULTS FILE CREATED = 2013-Nov-18 01:17:39 (2013-Nov-18 09:17:39 GMT)
USER SUPPLIED VALUES:
   RESULTS TYPE      = TRANSIENT
   TRANSIENT START   = 0
   TRANSIENT STOP    = 1.2e-07
   SIM TEMPERATURE   = 27 C
   AVERAGE           = nil
   AVERAGE ABS       = nil
   AVERAGE POS       = nil
   ACPEAK            = nil
   CUSTOM            = nil
   RMS               = nil
   PEAK              = t

---------------------- NET "bias100" ------------------------------
max

Pass/Fail %  Resistor  layer   Current   Width   PathLength I limit X1     Y1    X2    Y2    J/JMAX     Res      ViaArea No of        width/#via CurrentDirection
                                (mA)      (um)     (um)      (mA)                                                    (nm^2)  vias needed    (um/#)
pass-75.5    rg1974    metal3  1.61816   2        103.149   6.6    0.25   1.75  0     0     0.245176   0.008084 NA      NA           0.490352        S
pass-79.9    rg1973    metal3  1.32752   2        103.149   6.6    0.55   1.75  0.2   1.75  0.20114    0.00972  NA      NA           0.40228         E
pass-91.0    rh3592    metal2  0.119003  0.6      89.909    1.32   41.95  11.4  41.7  11.13 0.0901539  0.01622  NA      NA           0.0540923       NE
pass-98.7    rn17660   via2    0.0282677 NA       89.909    2.2    6.745  10.95 6.74  10.90 0.0128489  2.66     16900   1            1           metal3->metal2
pass-98.7    rn17662   via2    0.0282677 NA       89.909    2.2    7.025  10.90 7.02  10.90 0.0128489  2.66     16900   1            1           metal3->metal2
```

Video

To view the demonstration on plotting the EM analysis results in Voltus-Fi, see Viewing and Querying the EM Analysis Results in Voltus-Fi-XL video.

Access to this video will depend on the availability of a web browser and a Cadence Online Support account.

Generating EM Reports in Voltus-Fi-XL

You can generate both textual and html reports for EM analysis results in Voltus-Fi-XL from the simulation results by specifying the EMIR control file (`emir.conf`), the EMIR database file, and the output directory.

For more information about generating EM reports in Voltus-Fi-XL, see Reports Generated Using the EMIR Control File.

© 2020 All Rights Reserved.

For more information about default reports generated for EM analysis and to view a sample EM analysis report, see Default Reports Generated for EMIR Analysis.

Types of EM Analysis Plots

Following EM Plots are available for viewing and debugging:

- RJ JMAX – Current Density/Limit and J_MAX – Current Density Plots
- JAVG – Current Density/Limit and J_AVG Current Density Plots
- JABSAVG – Current Density/Limit and J_ABSAVG – Current Density Plots
- JACRMS – Current Density/Limit and J_ACRMS – Current Density Plots
- JACPEAK – Current Density/Limit and J_ACPEAK – Current Density Plots

Note: In addition to the above, the TC and RC plots are common to all analysis types. However, the TCAVG, TCRMS, RCAVG, and RCRMS plots are only available in the `qrcTechFile` flow and the EM Only ICT File flow. They are not supported in the `emDataFile` flow.

Note: For details, see TC, TCAVG, TCRMS – Tap Current Plots and RC, RCAVG, and RCRMS – Resistor Current Plots in the “IR Drop Analysis Results” chapter.

The different EM analysis plot types available and the corresponding analysis types are listed below:

- TC plots – max, avg, avgabs, AC Peak, rms analyses
- TCAVG, TCRMS plots – max, avg, avgabs, AC Peak, rms analyses (only available in the `qrcTechFile` flow)
- RC plots – max, avg, avgabs, AC Peak, rms analyses
- RCAVG, RCRMS plots – max, avg, avgabs, AC Peak, rms analyses (only available in the `qrcTechFile` flow)
- RJ JMAX and J_MAX plots – max analysis
- JAVG and J_AVG plots – avg analysis
- JABSAVG and J_ABSAVG plots – avgabs analysis
- JACRMS and J_ACRMS plots – rms analysis
- JACPEAK and J_ACPEAK – AC Peak analysis (only available in the `qrcTechFile` flow)

© 2020 All Rights Reserved.

RJ JMAX – Current Density/Limit and J_MAX – Current Density Plots

These plots are applicable to the “max” analysis type. The J_MAX plot reports the Current Density (J) value based on peak currents. It is used to locate high Current Density areas in the design.

The RJ JMAX or the Current Density/Limit plot reports the ratio of the wire Current Density to the Current Density limit that is allowed by the process – (J/JMAX) based on peak currents. So, all wires should have a ratio of less than 1.0. You can debug or fix the Current Density violations that have a ratio of more than1.0. The key advantage is that the value of this ratio is consistent across designs.

Information Displayed in Annotation Browser

The EM/IR tab of Annotation Browser displays the following columns for the Current Density plot types for peak currents:

- Value
- Layers
- Short Message
- Density
- Current
- Current Limit (mA)
- Needed Width
- Width
- Length
- Resistance
- Via Area
- Number of Vias
- Needed Number of Vias
- Net

The plot below shows the Current Density limit (J/JMAX) violation values based on peak currents for the selected power nets.

© 2020 All Rights Reserved.

Figure 9-6 RJ JMAX Plot for Specified Power Nets – Displaying Customized Violation Range

JAVG – Current Density/Limit and J_AVG Current Density Plots

These plots are applicable to the “avg” analysis type. The J_AVG plot analyzes and reports the Current Density (J) values based on average currents while the JAVG plot reports the J/JMAX violation values based on the average currents.

Information Displayed in Annotation Browser

The EM/IR tab of Annotation Browser displays the following columns for the Current Density plot types for average currents:

- Value

© 2020 All Rights Reserved.

- Layers
- Short Message
- Density
- Current
- Current Limit (mA)
- Needed Width
- Width
- Length
- Resistance
- Via Area
- Number of Vias
- Needed Number of Vias
- Net

The plot below shows the average Current Density limit (J/JMAX) violation values based on average currents for the selected nets. The violation range is customized using the *Min Max* sliders in the IR/EM Results form.

© 2020 All Rights Reserved.

Figure 9-7 JAVG Plot for Selected Nets – Displaying the Customized Violation Range

JABSAVG – Current Density/Limit and J_ABSAVG – Current Density Plots

These plots are applicable to the “avgabs” analysis type. The J_ABSAVG plot analyzes and reports the Current Density (J) values based on the average of the absolute current while the JABSAVG plot reports the J/JMAX violation values based on the average of the absolute current.

Information Displayed in Annotation Browser

The EM/IR tab of Annotation Browser displays the following information for the Current Density plots for the average of the absolute currents:

- Value
- Layers
- Short Message

© 2020 All Rights Reserved.

- Density
- Current
- Current Limit (mA)
- Needed Width
- Width
- Length
- Resistance
- Via Area
- Number of Vias
- Needed Number of Vias
- Net

The plot below shows the Current Density limit (J/JMAX) violation values based on the average of the absolute currents for all signal nets. The violation range is customized by using the *Min Max* sliders in the IR/EM Results form.

© 2020 All Rights Reserved.

Figure 9-8 JABSAVG Plot for All Signal Nets – Displaying the Customized Violation Range

JACRMS – Current Density/Limit and J_ACRMS – Current Density Plots

These plots are applicable to the “rms” analysis type. The J_ACRMS plot analyzes and reports Current Density (J) values based on the root mean square (RMS) value of the AC currents while the JACRMS plot reports the J/JMAX violation values based on the RMS value of the AC currents.

Information Displayed in Annotation Browser

The EM/IR tab of Annotation Browser displays the following columns for Current Density plots based on the RMS value of the AC currents:

- Value
- Layers

© 2020 All Rights Reserved.

- Short Message
- Density
- Current
- Current Limit (mA)
- Needed Width
- Width
- Length
- Resistance
- Via Area
- Number of Vias
- Needed Number of Vias
- Net

The plot below shows the Current Density (J/JMAX) violation values based on the RMS value of the AC currents for all power nets. The violation range is customized by using the *Min Max* sliders in the IR/EM Results form.

© 2020 All Rights Reserved.

Figure 9-9 JACRMS - Plot – Displaying the Customized Violation Range for All Power Nets

JACPEAK – Current Density/Limit and J_ACPEAK – Current Density Plots

These plots are applicable to the “AC Peak” analysis type, which is only available in the `qrcTechFile` flow. These plots are not available in the list for the `emDataFile` flow even if the analysis type “AC Peak” is selected. The J_ACPEAK plot analyzes and reports Current Density (J) values based on the peak value of the AC currents while the JACPEAK plot reports the J/JMAX violation values based on the peak value of the AC currents.

Information Displayed in Annotation Browser

The EM/IR tab of Annotation Browser displays the following columns for Current Density plots based on the peak value of the AC currents:

- Value

© 2020 All Rights Reserved.

- Layers
- Short Message
- Density
- Current
- Current Limit (mA)
- Needed Width
- Width
- Length
- Resistance
- Via Area
- Number of Vias
- Needed Number of Vias
- Net

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

10

Self-Heating Effect Analysis

- Overview on page 305
  - Self-Heating Effect Analysis on page 305
  - Thermal-Aware DC EM Analysis on page 306
  - Hierarchical SHE Analysis on page 306
- Data Requirements on page 307
- Compatible Software Versions on page 308
- Flow Overview on page 308
- Performing the Analysis on page 309
  - Running Pegasus or PVS LVS
  - Running Quantus Parasitic Extraction
  - Running Post-Layout Simulation using Spectre APS on page 311
  - Viewing the Analysis Plots in Voltus-Fi-XL on page 312
- Types of Plots on page 313
  - Thermal-DeltaT Regions on page 314
  - Thermal-DeltaT Tiles on page 315
  - RJ_SH - Current Density/Limit - max (SH) on page 316
  - JAVG_SH - Current Density/Limit - avg (SH) on page 316
  - JACRMS_SH - Current Density/Limit - rms (SH) on page 317
- Re-Evaluation of EM Limit Based on dTemp on page 317
- EMIR Control File Options on page 318

© 2020 All Rights Reserved.

- Batch Mode Support for Running SHE Analysis and Thermal-Aware DC EM Analysis on page 326
- Command for Encrypting Data in param.sh File on page 329
- Generating Reports on page 330

© 2020 All Rights Reserved.

Overview

The following topics are covered in this section:

- Self-Heating Effect Analysis
- Thermal-Aware DC EM Analysis
- Hierarchical SHE Analysis

Self-Heating Effect Analysis

As semiconductor technology scales down to sub-micron dimensions, power density increases, resulting in temperature increases—self-heating effect (SHE)—in devices. Semiconductor components are sensitive to temperature variations.

The main effects of self-heating are as follows:

- Degraded carrier mobility, which leads to a reduced on-current of transistors and, therefore, slower speed
- Higher interconnect metal resistivity, which results in longer delays
- Increased failure rate and reduced reliability of electronic devices, with secondary effects such as interconnect electromigration

In digital circuits, the key consequences are longer delays due to increased interconnect resistance and reliability issues due to electromigration. Analog circuits are more sensitive to temperature variations, which may degrade matching properties. Therefore, thermal simulations and modeling play an important role in the design of integrated circuits today.

The self-heating effect is driven by front-end-of-line (FEOL) self heat and back-end-of-line (BEOL) self heat.

- The FEOL is the first part of IC fabrication where the individual devices such as transistors, capacitors, and resistors are patterned in the semiconductor. FEOL covers everything up to, but not including, the deposition of metal interconnect layers.
- The BEOL is the second part of IC fabrication where the individual devices get interconnected with the wiring on the wafer. Both contribute towards the maximum rise in temperature or deltaT that causes the self-heating effect.

© 2020 All Rights Reserved.

Thermal-Aware DC EM Analysis

The thermal-aware DC EM flow considers the local thermal effect – this includes both joule-heating and self-heating – in DC EM checks.

The thermal-aware DC EM flow is enabled by setting the following variable:

In the Variables form on the EM tab of the IR/EM Results form by specifying the following variable:

```
coupledSH true
```

In batch mode by specifying the following command:

```
set_variable coupledSH true
```

Voltus-Fi-XL supports the SHE analysis flow and the thermal-aware DC EM flow visualization. You can view the results of the these analyses by plotting them on the Virtuoso layout to show the worst deltaT violations. You can then debug your design to reduce the self-heating effect. In this chapter, the subsequent sections will provide details of the SHE analysis flow that are also relevant to the thermal-aware DC EM flow. Additional parameters and commands for the latter flow will be highlighted within these sections.

Hierarchical SHE Analysis

Voltus-Fi supports the generation of reports that are used by Voltus for Hierarchical SHE analysis. For this, the following option of the `emirutil` command is specified in the configuration file:

```
emirutil selfheating_report=[type=mos-region | metal filename=<filename>
layer=<layername> tile=<no.of divisions in y-direction, no. of divisions in
x-direction>]
```

For details of the above command, see EMIR Control File Options. The SHE reports that are generated are specified while performing hierarchical SHE analysis, using the Voltus command, `analyze_self_heat`.

Important

This feature is only available when the VTS-Fi-AA license is checked out. This is an advanced license that is not available as part of the Voltus-Fi-XL license.

© 2020 All Rights Reserved.

Data Requirements

The table below lists the inputs that are required for, both the SHE analysis and the thermal-aware DC EM analysis in Voltus-Fi-XL:

Table 10-1 Data Required for SHE and Thermal-Aware DC EM Flow

Data	Purpose	Details
Model cards	Simulation	Spice model cards, which support SHE analysis for simulation
Detailed Standard Parasitic Format (xDSPF) file	Simulation	A detailed standard parasitic extraction file that contains x, y coordinates and l (length), and w (width) values. This file is required post simulation.
qrcTechFile or the ICT file	EMIR and SHE analysis	A file containing current limits for each process layer. This file is usually available through a foundry. **Note:** This file should contain the `Irms` equation.
EMIR control file (`emir.config`)	Simulation	A configuration file needed to run EMIR and SHE analysis with Spectre® APS solver run.
`param.sh` file	SHE analysis	File for deltaT FEOL and BEOL calculations. **Note:** The information contained in the `param.sh` file is proprietary. To use this file, you need a collateral from the foundry.
XML file	SHE analysis	A file created after simulation and provided as an input file for the Voltus-Fi-XL SHE flow. It contains `deltaT` information for all devices in the layout.

Important

To run the self-heating flow and the thermal-aware DC EM flow in batch mode, it is mandatory to set the following environment variable:

© 2020 All Rights Reserved.

```
setenv CDS_XVNC_TENBASE '+'
```

This variable is not required for run the SHE flow and the thermal-aware DC EM flow in the GUI.

This limitation will be removed in a future release of the software.

Compatible Software Versions

The table below lists the compatible software versions that are needed for running the SHE analysis flow and the thermal-aware DC EM flow:

Product Name	Release Version
PVS/Pegasus	PVS 13.1.1 HF1 and later to Pegasus 18.20
EXT	EXT 13.2.0 and later
MMSIM/SPECTRE	MMSIM 13.1.1 ISR10 and later to SPECTRE 16.1.0 onwards

Note: It is recommended that if any of the above release versions are not available, a later version of the product be used.

Flow Overview

The diagram below illustrates the Pegasus/PVS LVS – Quantus – Virtuoso® Analog Design Environment L for Netlist and Config file generation – Spectre® APS – Voltus-Fi-XL flow for SHE analysis and thermal-aware DC EM analysis.

© 2020 All Rights Reserved.

Figure 10-1 The High-Level Flow

Performing the Analysis

The following key tasks are performed for the SHE analysis and the thermal-aware DC EM flow:

- Running Pegasus or PVS LVS
- Running Quantus Parasitic Extraction
- Running Post-Layout Simulation using Spectre APS
- Viewing the Analysis Plots in Voltus-Fi-XL

These steps are detailed in subsequent sections.

© 2020 All Rights Reserved.

Running Pegasus or PVS LVS

This is the first step, which involves verifying the layout connectivity and creating an LVS database that is required for the subsequent Quantus parasitic extraction.

Note: Pegasus, the new physical verification system, will be available in the 18.2 physical verification release.

Running Quantus Parasitic Extraction

This is the second step, which involves using the Quantus engine to create the xDSPF file for all the device parameters and interconnect parasitics.

Data Requirements for Extraction Setup

While setting up extraction for SHE analysis, ensure the following:

- The Quantus command file should include the following command:

  ```
  output_setup \
  -keep_temporary_files "true"
  ```

  When set to `true`, the above command causes Quantus to leave the temporary files in the directory specified by the `output_setup -temporary_directory_name` command.

  When set to `false`, Quantus removes the temporary files when extraction has completed. By default this option is set to `false`.

  For more information about the Quantus command file options, see the “Quantus Command Files” chapter in the *Quantus Extraction Users Manual*.

- The Quantus version used for performing SHE analysis in Voltus-Fi-XL should be either the same or higher than that being used in the extraction step
- The layout should open within Virtuoso without any pop-up window.
- The OD shapes should be present in the layout for the OD layers specified in the file mentioned through the `shParamFile` option.

GUI Use Model

In the main menu of the Virtuoso Layout Suite, choose Quantus. Choose the Run Pegasus – Quantus or PVS – Quantus option. The Quantus Parasitic Extraction Run Form opens. Run extraction.

© 2020 All Rights Reserved.

For detailed information about all the settings in the Quantus Parasitic Extraction Run form, see the “Quantus Graphical Interface” chapter in *Quantus Extraction Users Manual*.

Note: Pegasus, the new physical verification system, will be available in the 18.2 physical verification release.

The output includes the following:

- An xDSPF file, which contains all the parasitic information of the design – all the nets, x- and y-coordinates of transistors, parasitic resistors, and nets connected.
- A Quantus output directory, which contains the shape database.

Running Post-Layout Simulation using Spectre APS

Once extraction is done, run post-layout simulation using the EMIR control file `(emir.config)` file to create the SHE analysis reports. Before performing simulation, ensure the following in the ADEL or the Simulation window:

- Setup transient analysis.
- Choose *Setup – Simulation File – Model Files* to add the SHE Models. These models are required to create the `*.dtemp#.xml` file that contains information about the increase in temperature for each device.
- Select *Simulation – Reliability – Setup*. The Reliability form opens. In this form, specify the following:
  - Select *Enable Reliability* to enable reliability simulation.
  - Select *Enable TMI Flow.*
  - In the *self-heating/aging(tmiShe)* cyclic field, select *self-heating only (tmiShe=1).* This option enables only the TMI self-heating mode.

    Note: This option is available in IC616ISR8 and ICADV12.1ISR10 releases with MMSIM13.1ISR10 or above versions.

Now, you can start the simulation either by using *Simulation – Run* option or clicking the “run simulation” button provided in the right panel of the ADEL window.

Note: For detailed information about how these options are set, see “Using the Reliability Simulator Interface” chapter in the *Virtuoso Analog Design Environment L User Guide*.

Note: The EMIR control file options required to be specified for SHE analysis are detailed in the EMIR Control File Options section. These options can also be specified in the Variables form that opens from the EM tab of the IR/EM Results form.

© 2020 All Rights Reserved.

Viewing the Analysis Plots in Voltus-Fi-XL

Once the simulation run is complete, the simulation database saved in the `*.emir#_bin` file is loaded into Voltus-Fi-XL to view the SHE analysis plots. The SHE analysis and thermal-aware DC EM analysis plots are available on the EM tab of the IR/EM Results form.

The steps for populating the EM tab of the IR/EM Results form are detailed in the Viewing the EM Analysis Results section of the "EM Analysis Results" chapter.

In addition, for the SHE analysis flow, ensure that the following inputs are provided on the EM tab of the IR/EM Results form:

- Enable the *QRC Run*.
- In the QRC Run Location pop-up window, provide the details of the *QRC Run Directory* and *Run Name*.
- Click *Variables* to open the Variables form. In this form, specify the following variables:
  - shParamFile
  - shXMLFile

  Note: For a sample `shParamFile`, see Self-Heating Effect Analysis File in the "File Formats" chapter.

  Note: To enable the thermal-aware DC EM analysis, in addition to the above variables, the following variable needs to be set to `true`:

  - coupledSH
- Select *rms* and *avg*.

This form is shown below.

© 2020 All Rights Reserved.

Figure 10-2 The IR/EM Results Form

- In the Plot Results group box, the self-heating effect analysis plots are available in the list. These are detailed in the section below.

Types of Plots

The types of self-heating effect analysis and thermal-aware DC EM analysis plots that are available in the list are detailed below.

© 2020 All Rights Reserved.

Figure 10-3 Plot Types

Thermal-DeltaT Regions

Reports the increase in temperature (deltaT) for all OD and metal layers in the entire design on the Virtuoso layout. This plot provides a detailed view of the temperature variations on the design. This is shown in the figure below.

© 2020 All Rights Reserved.

Figure 10-4 The Thermal-DeltaT Regions Plot

Thermal-DeltaT Tiles

Reports the increase in temperature (deltaT) per tile based on the number of tiles, specified in the IR/EM Results form, on the Virtuoso layout. For this plot, specify the dimensions for the tiles in the `Tiles` text box on the *EM* tab of the IR/EM Results form. The design in the layout is divided into sections based on the number of tiles that are specified, for example, 15x20.

On the Virtuoso layout, the tiles are colored on the basis of the maximum temperature increase in each tile. This is shown in the figure below.

© 2020 All Rights Reserved.

Figure 10-5 The Thermal-DeltaT Tiles Plot

RJ_SH - Current Density/Limit - max (SH)

Analyzes and reports the J/JMAX violations based on peak current after performing the SHE analysis.

Note: This plot is not available for the thermal-aware DC EM flow

JAVG_SH - Current Density/Limit - avg (SH)

Analyzes and reports the J/JMAX violations based on the average currents after performing the SHE or thermal-aware DC EM analysis.

© 2020 All Rights Reserved.

JACRMS_SH - Current Density/Limit - rms (SH)

Analyzes and reports J/JMAX violations based on the RMS value of the AC currents after performing the SHE analysis.

Note: This plot is not available for the thermal-aware DC EM flow.

Re-Evaluation of EM Limit Based on dTemp

The EM effect is re-evaluated based on the new temperature. In the SHE flow, this temperature is the sum of the simulation temperature and the changed temperature due to the self-heating effect. In the thermal-aware DC EM flow, this temperature is the sum of the simulation temperature and the changed temperature due to joule heating and the self-heating effect.

The `Current_Limit` is calculated based on this new temperature and compared with the actual current flowing through the metal resistor to find the EM pass or fail.

The `shEMEffect` variable is used to re-evaluate the EM limit based on individual metal BEOL_T and tile BEOL_T temperatures. This variable is set using the `set_variable` command in the Voltus-Fi command file:

To re-evaluate EM on individual metal BEOL_T:

```
set_variable shEMEffect beolT
```

To re-evaluate EM limit based on tile BEOL_T temperature:

```
set_variable shEMEffect tiles_no.of divisions in y-direction X no.
of divisions in x-direction
```


© 2020 All Rights Reserved.

EMIR Control File Options

A complete list of the EMIR control file options that are supported in Voltus-Fi-XL is provided in the table, EMIR Control File Options Supported in Voltus-Fi-XL, in the “Data Preparation” chapter.

In addition, there are some options that are specific to the SHE flow and the thermal-aware DC EM flow. The table below provides a list of options that are specified in the `emir.conf` file for these analyses.

© 2020 All Rights Reserved.

Table 10-2 Supported EMIR Control File Options for Self-Heating Effect Analysis

Keyword	Option Set	Explanation	Default Value
net	analysis=[selfheating]	Turns on the self-heating effect analysis for the specified device under test (DUT). **Note**: Ensure that the Irms option is specified in the control file.	none
	selfheating_skipinst=[notDUT]	Specifies the instances on which SHE analysis will not be performed.	none
	selfheating_skipmodels=[<models to skip>]	Specifies the models on which SHE analysis will not be performed.	none
emirutil	shParamFile="param.sh"	Specifies the self-heating effect analysis parameter file, which is required for calculating the SHE analysis results. A sample shParamFile format is provided in the "File Formats" chapter.	none
	filter_em_sh_threshold=*value*	Defines the SHE EM analysis threshold value for the resistors being reported. The value for filter_em_sh_threshold is the J/Jlimit value. filter_em_sh_threshold=1 reports only those resistors that have a J/Jlimit value above 1. **Note**: Specifying this option is equivalent to filtering all resistors that have passed the EM analysis check.	none

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`shXMLFile=[<file>]`	Specifies the XML file to be used in the SHE analysis flow. This file contains information about the temperature change for every instance. **Note:** From Spectre16.1 ISR7 release onwards, the path of the XML file need not be specified in the emir configuration file. The software is enhanced to automatically locate and use the file for SHE analysis. Specify the `shXMLFile` option only when you want to override the existing XML file.	`none`
	`shEMEffect=[beolT | tiles_no.of divisions in y-direction X no. of divisions in x-direction]`	Specifies the basis for the re-evaluation of EM effect. `beolT`: specifies that re-evaluation will be done on the basis of individual metal BEOL_T. `tiles_no.of divisions in y-direction X no. of divisions in x-direction`: specifies that re-evaluation will be done on the basis of tile BEOL_T. For example: `shEMEffect=tiles_10x15`	`none`
	`extendSHRegion=[true | false]`	Specifies whether or not the OD shape extension is to be done. **Note:** This parameter is applicable only to the thermal-aware DC EM flow.	`false`
	`extendODRange=value`	Specifies the range of the OD shape expansion. The unit of this parameter is um (micro meter). **Note:** This parameter is applicable only to the thermal-aware DC EM flow.	`1.8`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`extendJouleHeating=[true | false]`	Specifies whether or not metal-shape expansion is to be done for joule-heating calculation. **Note:** This parameter is applicable only to the thermal-aware DC EM flow.	`false`
	`extendMetalRange=`*`value`*	Specifies the number of metal pitch for metal expansion. **Note:** This parameter is applicable only to the thermal-aware DC EM flow.	`0`
	`extendHRRange=`*`value`*	Specifies the range of the design resistor shape expansion. The unit of this parameter is um (micro meter). **Note:** This parameter is applicable only to the thermal-aware DC EM flow. The default value may be different for different process nodes.	`1.8`
	`Tambient=`*`value`*	Specifies the ambient temperature for self-heating analysis. If this value is not specified, the software calculates it as `Tj-deltaT`.	`Tj-deltaT`
	`disableODSH=[true | false]`	When set to `true`, disables the self-heating effect of devices in the SHE flow.	`false`
	`disableHRSH=[true | false]`	When set to `true`, disables the self-heating effect of design resistors in the SHE flow.	`false`
	`coupledSH=[true | false]`	When set to `true`, specifies that the thermal-aware DC EM flow should be enabled.	`false`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`view=[lib=<lib> cell=<cell> view=<layout>]`	Specifies the Virtuoso Layout view. **Note:** For the software to recognize the cell, view and library, ensure that "`cds.lib`" is either present in the directory in which EMIR simulation is run or its path is specified using the `cds.lib` command described below.	`none`
	`cds.lib="lib_path"`	Specifies the path of the cds.lib. It is used to specify the path when the `cds.lib` is not in the current working directory. For example, `emirutil cds.lib="/home/test/cds.lib"`	
	`qrc_output=[runDir=dir_name runName=qrc_run_name]`	Specifies the Quantus run directory and run name. This parameter is required for self-heating effect analysis.	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`selfheating_report=[type=mos-region | metal filename=<filename> layer=<layername> tile=<no.of divisions in y-direction, no. of divisions in x-direction>]`	Generates the self-heating effect report, which is a text file. You can specify whether the report should contain deltaT information for the specified mos-region or metal and resistor or tile-based by using the relevant parameters. Example: `emirutil selfheating_report=[type=mos-region filename=mos-region-tiles.txt tiles=15x10]` **Note:** To generate hierarchical SHE analysis reports using this command, the VTS-Fi-AA license is required to be checked out. This is an advanced license that is not available as part of the Voltus-Fi-XL license. If you do not want to perform hierarchical self-heating analysis, ensure that this option is not specified in the configuration file. The SHE EM analysis reports can be generated by specifying the `print_em_report` option in the configuration file with the required `-type` option. No special license is required for generating these reports. This option is detailed below.	`none`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value

	`print_em_report=[net=[all_nets | all_power | all_signal | netname] type=[acrms_sh | avg_sh | peak_sh | acpeak_sh] filename=`*`output_file_name`*`]`	Generates the EM report with the self-heating effect. You can specify the following:
		`filename`: create report with the specified filename.
		`net`: create report for specified nets - all power nets (`all_power`), all signal nets (`all_signal`), all nets (`all_nets`), or net (`netname`).
		`type` - create report for specified analysis type; `acrms_sh, avg_sh, acpeak_sh` or `peak_sh.`
		For example:
		`print_em_report=[net=all_nets type=acrms_sh filename= acrms_sh.txt]`
			`none`

	`genRthReport=[true | false]`	Generates a thermal resistance report with cell name and its corresponding resistance `(Rth)` value. **Note:** This parameter is applicable only to the thermal-aware DC EM flow. For example: `emirutil genRthReport=true`	`false`
The options listed below are specified only for the SHE flow with heat-sink effect			
`emirutil`	`enableSHHeatSink=[true | false]`	When set to `true`, specifies that the heat-sink effect should be included in the SHE analysis. **Note:** If this parameter is set to `true`, ensure that heat-sink models are available.	`true`

© 2020 All Rights Reserved.

Keyword	Option Set	Explanation	Default Value
	`extendHeatSinkRange=value`	Specifies the expanded region of the OD. The heat sink structure inside this region can only act as heat sink for the specified OD shape. **Example**: `emirutil extendHeatSinkRange=0.5`	`0.4`
	`connectingODLayers="layer1 layer2 layer#"`	Specifies the list of layers to identify connecting ODs for heat sink effect. The default value is shown below. `emirutil connectingODLayers="M0_OD M0_OD1 M0_OD2 MVIA0_1 MVIA0_2 M1"`	`M0_OD M0_OD1 M0_OD2 MVIA0_1 MVIA0_2 M1`
	`shParamFile="param.sh"`	Specifies the self-heating effect analysis parameter file, which is required for calculating the SHE analysis results with heat-sink effect. **Note:** For SHE with heat-sink effect, this file should include the heat-sink rules.	`none`
	`print_selfheating_report=[type=heatsink filename=filename]`	When specified, generates the heat-sink self-heating factor report. A sample report is provided below. **Note:** No special license is required for generating this report.	`none`

Heat-Sink Self-Heating Sample Report:

```
-----------------------------------------------------------------------------
VERSION: <version>
HEADER:
OD_Name  llx  lly  urx  ury  od_area  od_dT  od_dT_p  od_dT_p_be  od_dT_p_fe
connecting_wire  layer  totalA_on_region
connecting_via  layer  totalA_on_region
```


© 2020 All Rights Reserved.

```
connecting_OD  od_cell  od_dT  od_area  layer
---------------------------------------------------------------------------
```

For more information about how to set the emirutil command options, see Using the emirutil Command in the EMIR Control File in the “Variables” chapter.

Batch Mode Support for Running SHE Analysis and Thermal-Aware DC EM Analysis

You can run the SHE analysis and thermal-aware DC EM analysis in batch mode.

For SHE analysis, set the following variables using the set_variable command:

```
set_variable shParamFile param_sh.file
set_variable shXMLFile xml_file
set_variable shEMEffect {beolT | tiles_no.of divisions in y-direction X no.of divisions in x-direction}
```

For thermal-aware DC EM analysis, in addition to the above commands, specify the following:

```
set_variable coupledSH true
set_variable extendJouleHeating true
set_variable extendSHRegion true
```

For more information about the above variables, see the “Variables” chapter.

The commands listed below are common for both flows, SHE and Thermal-aware DC EM analysis:

- load_view

  Specifies the layout to be used for SHE analysis. The syntax is provided below.

  ```
  load_view \
  -libname lib \
  -cellname cell \
  -viewname layout
  ```

- qrc_run_directory

  Specifies the location of the Quantus run directory. The syntax is provided below.

  ```
  qrc_run_directory run_directory
  ```

- qrc_run_name

  Specifies the Quantus run name. The syntax is provided below.

  ```
  qrc_run_name qrc_run_name
  ```

- load_em_results

© 2020 All Rights Reserved.

Loads the results of EM analysis. The syntax is provided below.

```
load_em_results \
-tech_file tech_file_name \
[-layer_mapfile layer_map_filename] \
results_file_name
```

- print_em_report

  Prints the EM analysis reports. The syntax is provided below.

```
print_em_report \
-net -all_nets \
-type [acrms_sh | avg_sh | peak_sh] \
-filename output_file_name
```

- print_thermal_report

  Prints the thermal reports for the hierarchical self-heating EM analysis.

```
print_thermal_report \
-type {mos-region | metal} \
[-tiles tile_matrix] \
-filename output_file_name
```

For details of the above commands, see the "Batch Mode Execution" chapter.

Example

The following set of commands report the results of SHE analysis for cell, `cell1` of library, `test_lib`. The SHE analysis results will be reported in the `./acrms_sh.txt` file. The OD regions temperature will be reported in the `mos-region.txt` file:

```
load_view
     -libname test_lib \
     -cellname cell1 \
     -viewname layout

qrc_run_directory svdb
qrc_run_name design

load_em_results \
     -tech_file qrcTechFile
     -layer_mapfile contactmapfile/input.emir0_bin

print_em_report
     -net all_nets \
     -type acrms_sh \
     -filename ./acrms_sh.txt
```


© 2020 All Rights Reserved.

```
print_thermal_report \
     -type mos-region \
     -filename ./mos-region.txt
```


© 2020 All Rights Reserved.

Command for Encrypting Data in param.sh File

Voltus-Fi supports encrypting the contents of the `param.sh` file to hide confidential data. Use the below command to encrypt the contents of the file using the `vfibatch` command:

```
vfibatch -cmd <cmd-file-name>
```

The contents of the `cmd-file-name` file are as follows:

```
encrypt_file
-input_file_name <filename>
-output_file_name <filename>
```

Where,

`-input_file_name` is the name of the file whose contents are to be encrypted

`-output_file_name` is the encrypted file as viewed by the user. It contains the contents of the input file with confidential data in encrypted form.

In the input file, the text to be encrypted is placed inside below tags:

```
<Encryption Start>...<Encrypton End>
```

A sample input file whose contents are to be encrypted is shown below:

```
line1
line2
line3
line4
<Encryption Start>
Encrypt line1
Encrypt line2
<Encryption End>
line5
line6
```

The output file for the above input file is shown below:

```
line1
line2
line3
line4
<Encryption Start>
   u    s    f    i’{BèŸô`ʔïáâ×Ù û¢:#zšå„Và´-£¬?~    v    o    l    trcn3 tpeni ä
```


© 2020 All Rights Reserved.

```
EsÙ∀š¬›^ÐÞAryù™   v   o   l*ª w±Ä›¯}1-?kB9dP…Ö
<Encryption End>
line5
line6
```

Generating Reports

The text reports for both SHE and thermal-aware DC EM analysis are only available in batch mode using the sequence of commands detailed in the above section. The types of reports that are generated are detailed below:

- Thermal reports containing temperature rise of OD region and metal resistors.

  The format of the OD region report is as follows:

  Figure 10-6 OD Region Report Format

```
design: design_name
area: (x1 y1) (x2 y2)
report format: <delta_temperature> <od_area>
coordinate unit: um
delta_temperature          od_area
```

  For thermal-aware DC EM analysis, when the option `extendSHRegion` is set to true, the OD region report format is as follows:

  Figure 10-7 OD Region Report Format for Thermal-Aware DC EM Flow

```
# design: design_name
# area: (x1 y1) (x2 y2)
# report format: <delta_temperature> <od_area>  <extended od_area>
# coordinate unit: um
```

  The format of the metal resistors report is as follows:

  Figure 10-8 Metal Resistors Report Format

```
# LayerName          Irms(mA)     Alpha      Beta T_FEOL(C)  T_RMS(C) T_BEOL(C)  Metal-region
#
```


© 2020 All Rights Reserved.

- Tiles-thermal reports containing temperature rise of OD region and metal resistors based on the specified tile matrix.

  The format of the metal resistors report (tile-based) is as follows:

Figure 10-9 Metal Resistors Tile-Based Report Format

```
-----------------------------------------------------------------------
DIE_STACK_TEMPERATURE_MAP
DIE_AREA <x y> <x y> <x y>
NUMBER_OF_LAYERS #

LAYER <layername>
TEMPERATURE
XCOR   x1  x2  x3  x4  x5  x6....xn
YCOR   y1  y2  y3  y4  y5  y6....yn





END_TEMPERATURE
END_LAYER
-----------------------------------------------------------------------
```

  The format of the OD region (tile-based) report is as follows:

Figure 10-10 OD Region Tile-Based Report Format

```
-----------------------------------------------------------------------------------------------------
#Tiles MOS-region report
"lower left x" "lower right y " "upper right x" "upper right y" "number of ROWS" "number of columns"
<deltaT for each tile in the matrix>
-----------------------------------------------------------------------------------------------------
```

- The SHE EM reports containing dTemp of each parasitic resistor and EM report (including `pass/fail`, `em_limit`, and `needed width`) both with and without the self-heating effect. The following reports are generated:
  - `acrms_sh.txt`: AC RMS EM report both with and without the self-heating effect
  - `avg_sh.txt`: Average EM report both with and without the self-heating effect
  - `peak_sh.txt`: Peak EM report both with and without the self-heating effect

  The formats of these report file are provided below.

© 2020 All Rights Reserved.

Figure 10-11 AC RMS EM Report Format

```
ELECTROMIGRATION ANALYSIS RESULTS (acrms_sh)

Tj  = 70
deltaT = 5

                                                |   Without SH Effect   |       With SH Effect    |
 Resistor    layer     X    Y    current width(um)/| %fail curr limit    minW | %fail curr limit    minW | dT(SH)    feolT   beolT coupled
                                   (mA)    viaCount |          (mA)       (um) |          (mA)       (um) |
---------------------+----+----+----------+--------+-----------------+--------+-------+---------+--------+--------+-------+-------+--------
```

Figure 10-12 Average EM Report Format

```
ELECTROMIGRATION ANALYSIS RESULTS (avg_sh)

Tj  = 70

                                                |     Without SH Effect        |        With SH Effect        |
 Resistor     layer     X    Y    current width(um)/| %fail     curr limit    minW | %fail     curr limit    minW | Tj(SH)   feolT   beolT
                                   (mA)    viaCount |              (mA)       (um) |              (mA)       (um) |
-------------+---------+----+----+----------+--------+----------+----------+--------+-----------+---------+--------+--------+-------+--------
```

Figure 10-13 Average EM Report Format for Thermal-Aware DC EM Analysis

```
ELECTROMIGRATION ANALYSIS RESULTS (avg_sh)

Tj  = 110
deltaT = 5
Tenv = Tj - deltaT
beolT = coupledT + joule_heatingT
Tmetal = Tenv + beolT

                                                |      With Thermal Effect |
 Resistor   layer      X     Y    current width(um)/| %fail  curr limit   minW |  feolT coupledT joule_HeatingT   beolT
           Tmetal
                                   (mA)    viaCount |            (mA)       (um) |
 --------+--------+-------+------+------+--------+---------+-------+-----+-----+-------+-------+---------------+------
```

Figure 10-14 Peak EM Report Format

```
# ELECTROMIGRATION ANALYSIS RESULTS (peak_sh)

# Tj  = 70
#

#                                               |     Without SH Effect        |        With SH Effect        |
#  Resistor   layer     X    Y    current width(um)/| %fail     curr limit    minW | %fail     curr limit    minW | Tj(SH)   feolT   beolT
#                                   (mA)    viaCount |              (mA)       (um) |              (mA)       (um) |
#-----------+---------+----+----+----------+--------+----------+----------+--------+-----------+---------+--------+--------+-------+-------
```

- Thermal resistance report with cell name and thermal resistance (Rth) value. This is only generated in the thermal-aware DC Em flow.

© 2020 All Rights Reserved.

Figure 10-15 Thermal Resistance Report Format for Thermal-Aware DC EM Analysis

© 2020
All Rights Reserved.

© 2020 All Rights Reserved.

11

ESD Analysis

- Overview
- Specifying ESD Analysis Settings
- xDSPF File Optimization
  - Generating Optimized xDSPF File
- xDSPF File Optimization
  - Effective Resistance Checks Flow
  - Key Features of ESD Effective Resistance Checks
  - Prerequisites for Effective Resistance Checks
  - Running Effective Resistance Checks
- ESD Current Density Checks
  - ESD Current Density Checks Flow
  - Key Features of ESD Current Density Checks
  - Types of Analysis for ESD Current Density Checks
  - Prerequisites for ESD Current Density Checks
  - ESD Modeling
  - Specifying ESD Models
  - ESD Testbench
  - Specifying ESD Testbench
  - ESD Zaps
  - Creating Custom ESD Zaps
  - Running Current Density Checks

© 2020 All Rights Reserved.

- Rule-Based ESD Checks
  - ESD Rule File
  - Running Rule-Based ESD Checks
- ESD Form Description

Overview

Electrostatic Discharge (ESD) is defined as the transient electric current between two objects at different electrostatic potentials causing an ESD event. Due to shrinking geometries of nanometer designs, devices are at a higher risk to ESD exposure and are prone to damage.

The ESD analysis feature in Voltus-Fi-XL lets you check whether every power or ground pin or bump has an ESD device along the potential ESD paths or not. In addition, you can analyze the device placement to ensure that the device does not violate the effective resistance limit.

Given a set of pins and ESD devices in ESD analysis, you can find the effective resistance between each pin and ESD device pair. The objective of ESD analysis is to identify and highlight the pins that have an effective resistance higher than the specified threshold value. It also checks whether the total effective resistance from a power pin to an ESD device to a ground bump is within the specified threshold value. In addition, static verification using known current density limits is used to analyze the current flow from the pad to the ESD devices.

To prevent an ESD event, you can place the ESD devices or clamps at power/ground pads to provide an efficient discharge path from one pin or bump to another.

Voltus-Fi-XL supports the following checks in ESD analysis in batch mode and GUI:

- Effective resistance checks
- Current density checks

The following features of ESD analysis are beta features and available only with the VPS_NG100 license:

- Current density checks in batch mode
- GUI support for ESD analysis

For more information, contact your Cadence representative.

© 2020 All Rights Reserved.

Specifying ESD Analysis Settings

Before running ESD analysis, you need to specify the settings in the Voltus-Fi GUI. The steps are detailed below.

1. Opening ESD Form
2. Specifying Mandatory Inputs
3. Specifying Optional Inputs
4. Loading DSPF File
5. Selecting Instances for Analysis
6. Specifying Advanced Settings

Opening ESD Form

1. In the Voltus-Fi-XL console, choose *ESD Analysis – ESD Analysis*.

© 2020 All Rights Reserved.

2. Open the *ESD Setup* tab of the ESD form.

Specifying Mandatory Inputs

1. In the *ESD Common Setup* section, specify the nets in *Power Nets*, *Ground Nets*, and *Signal Nets*.
2. Specify a filename for *DSPF File*.
3. Specify the ESD cells in *ESD transistors/diodes*.
4. In the *Output Dir* box, specify the directory in which the reports generated for ESD analysis will be saved.

© 2020 All Rights Reserved.

5. Specify a filename for *DFII layer map file*.
6. Specify a filename for *Model File.*
7. In the *qrcTechFile/ICT File* box, specify the Quantus technology file or the ICT file that contains the EM limits for ESD current density calculations.

Specifying Optional Inputs

1. In the *Optional Inputs* section, specify a filename for *CDL File*.
2. Specify the ESD CDL cells in *CDL Cells*.
3. In the *Common EMIR conf File* box, specify the configuration file to be used for all zap simulations.
4. Specify a filename for *EM Only ICT File*.

Loading DSPF File

1. Click *Load DSPF* to load the DSPF file that is specified in the *DSPF File* field in the *ESD Common Setup* group box.

   If an optimized xDSPF file is present inside the output directory, Voltus-Fi uses the file for ESD analysis. If an optimized xDSPF file is not present inside the output directory, Voltus-Fi displays a message prompting you to optimize the file. You can either proceed without optimizing the xDSPF file or generate an optimized xDSPF file. It is recommended to generate an optimized xDSPF for faster Spectre simulation.

   After the DSPF file is loaded, the *DSPF summary* window displays the number of nodes, resistors, and ESD instances.

Once the fields in the *ESD Setup* group box are populated and the DSPF file is loaded, the *ESD Instances* table lists ESD instances of all ESD macro cells in a tree view.

Selecting Instances for Analysis

You can select instances for effective resistance and current density checks. The *ESD Instances* table lists ESD instances of all ESD macro cells and all fingers of the listed ESD

© 2020 All Rights Reserved.

instances in a tree view.

To select instances for effective resistance and current density checks, do the following:

1. Click *Select All Instances* to select all instances, or click the check boxes corresponding to specific instances.
2. Click *Update* to update the ESD device instances that are to be processed.

© 2020
All Rights Reserved.

When an instance is selected, all related fingers are highlighted on the layout. You can select a single finger to highlight it in the layout.

Specifying Advanced Settings

With advanced settings, you can specify the number of threads to use for multi-threading, the maximum number of computers to use for distributed processing, or the maximum number of computers and the number of threads to use for superthreading. You can also specify the Spectre simulation settings.

Local Multithreading

1. Click *Advanced Settings* in the *ESD Setup* tab.

© 2020 All Rights Reserved.

The Advanced Settings form opens.

2. On the *Run Settings* tab, select *Multi CPU* from the *Simulation Run Mode* list.
3. Specify a value for *Number of Local Processors*.
4. Click *Ok* or *Apply*.

Remote Multithreading, Distributed Processing, and Superthreading

1. On the *Run Settings* tab, select *LSF* from the *Simulation Run Mode* list.
2. Specify a value for *Remote Hosts*.

   This value is required for both distributed processing and superthreading.

© 2020 All Rights Reserved.

3. Specify a value for *CPU's per Remote Host*.

   Note: For superthreading, this value must be used in conjunction with the value of *Remote Hosts*.

4. In the *LSF Configuration* section, specify any additional arguments you need to provide to the LSF job submission command in the *Args* box.

5. Specify a resource string for the LSF queue in the *Resource* box.

6. Specify a value for *Queue* for the LSF configuration.

7. Specify a value for *Timeout*.

8. Click *Generate*.

   The *Effective Command* box shows the effective command string compiled from the values in the *LSF Configuration* section. This value is used as the submission string to LSF. You can modify this value for custom distributed mode settings.

9. Click *Ok* or *Apply*.

Spectre Simulation Settings

1. Open the *Simulation Settings* tab of the Advanced Settings form.

2. In the *Spectre Simulator Options* box, specify the Spectre setup simulation file. For example, `highvoltage=yes`.

3. In the *Spectre Command Line Options* box, specify the options to be used while invoking Spectre. For example, `+aps +postlayout=hpa`. For static DC simulation, you can also add `+fastdc`.

4. In the *Spectre Transient Options* field, specify the options to be used while running the `tran` simulation. For example, `method=trap`.

   Note: Type the values, or click *Default* to enter the default values and click *Apply.* When you click *Default*, any previously specified settings for the particular field are cleared.

5. In the *Spectre Threads* box, specify the number of threads to be used while running simulation.

6. Click *Ok* or *Apply*.

Related Topics

- Setting Up ESD Analysis in Voltus-Fi-XL (video)

© 2020 All Rights Reserved.

xDSPF File Optimization

Voltus-Fi uses the specified full-chip flat xDSPF file and optimizes it for faster Spectre simulation. The optimized xDSPF file consists of all ESD protection devices and nets, parasitics, and all other components that are related to the ESD discharge paths for all input net ports or pins.

After an xDSPF file is optimized, Voltus-Fi automatically runs the SPF Checker utility on the optimized file to ensure that the DSPF file is compatible with the simulation requirements.

Generating Optimized xDSPF File

You can generate an optimized xDSPF file in both batch and GUI modes.

Batch Mode

To generate an optimized xDSPF file in batch mode, do the following:

- Provide information about the `esd_optimizer` command in the command file, which is specified using the `vfibatch` command.

  `vfibatch -cmd` *`cmd-file-name`*

GUI Mode

To generate an optimized xDSPF file in GUI mode, do the following:

1. In the *ESD Setup* tab of the ESD form, specify the ESD analysis settings and load the DSPF file.

© 2020 All Rights Reserved.

2. In the *ESD Optimizer* tab, click *Run Optimizer*.

After the optimizer run is complete, click *Load DSPF* on the *ESD Setup* tab to use the optimized DSPF file.

© 2020
All Rights Reserved.

ESD Effective Resistance Checks

Effective Resistance Checks Flow

The following diagram illustrates the flow of ESD effective resistance checks in Voltus-Fi-XL:

Key Features of ESD Effective Resistance Checks

The following features are supported for the ESD effective resistance checks in Voltus-Fi-XL:

- **Discarding R-effective of Pin to First ESD Device when Reporting pin-to-pin**: The following option of the `analyze_esd_generic` command is used to discard the effective resistance of the pin to the first ESD for signal nets:

  `-discard_pin_to_first_ESD_for_signal_net {true|false}`

- **Recognizing ESD Instances in a Circuit**: When the ESD-cell is also used for non–ESD purposes, Voltus-Fi falsely recognizes the non-ESD instances as ESD. To avoid

© 2020 All Rights Reserved.

this, the following option is provided in the `analyze_esd_generic` command to let you identify ESD-instances by name:

```
-esd_instance_name_string "clamp ESD"
```

When the above option is specified, only instances that contain the words, 'clamp' or 'ESD', in their names are recognized as ESD-instances. The following message is displayed in the Voltus-Fi log file:

```
INFO: All instances that do not contain the following not case-sensitive token-
string will be discarded from ESD-analysis.
clamp ESD
```

- **Merging Fingers**: By default, Voltus-Fi merges fingers for resistance checks. Specify the following option of the `analyze_esd_generic` command if you do not want to merge fingers:

  ```
  -merge_fingers false
  ```

- **CDL Processing for ESD Analysis**: The CDL ESD cells can be specified in the Voltus-Fi GUI for ESD analysis. You can specify multiple ESD cell names in the *Cells* field in the *CDL* group box of the ESD form. Multiple cell names can be specified separated by a space. This input is optional.

Voltus-Fi automatically finds all the leaf-level instances present inside the specified ESD CDL cells. The leaf-level instances present inside the CDL cells are matched inside the DSPF and used for ESD analysis. For example, you have specified `ESD_cell1` and `ESD_cell2` in the schematic CDL netlist. The figures below show the top-level circuit before and after the CDL cells are processed.

© 2020
All Rights Reserved.

Top-Level Circuit before CDL Processing

© 2020 All Rights Reserved.

Top-Level Circuit after CDL Processing

Prerequisites for Effective Resistance Checks

The following table lists the required inputs for running effective resistance checks in Voltus-Fi-XL.

Input	Details
DSPF file	It provides the complete parasitic information of the design. This file is generated using Quantus.
ESD model information	It is included in the DSPF file.

© 2020
All Rights Reserved.

Input	Details
Technology file or qrcTechFile	It contains the EM limits for ESD effective resistance calculations.
Net names	Power/ground/signal nets that are required for pin information.
Output directory	Directory where the ESD analysis reports are written.
Circuit Design Language (CDL) file	The schematic netlist.

Running Effective Resistance Checks

You can run effective resistance checks in both batch and GUI modes.

Batch Mode

To run effective resistance checks in batch mode, do the following:

- Provide information about the `analyze_esd_generic` command in the command file, which is specified using the `vfibatch` command.

  ```
  vfibatch -cmd cmd-file-name
  ```

GUI Mode

To run effective resistance checks in GUI mode, do the following:

1. In the *ESD Setup* tab of the ESD form, specify the ESD analysis settings and load the DSPF file.

© 2020 All Rights Reserved.

2. Open the *ESD Run* tab.

3. Click the *Resistance Checks* tab.

4. Specify the threshold values for *Pin-to-Pin Threshold* and *Pin-to-Clamp Threshold*.

   The Pin-to-Clamp resistance check threshold value is also applicable for the Clamp-to-Clamp resistance check. By default, the threshold value is set to `10 Ohms`.

5. Select *Finger Merge* to merge the MOS fingers in the resistance check reports. This option is selected by default.

6. Select *Split Pins Shorting* to short the split pins.

© 2020 All Rights Reserved.

7. Click *Run R-effective Checks* to run the effective resistance checks.

The table is populated with the reports for the different resistance checks. You can click an ESD device in the table to view all its descendants in the layout window as fly-lines. The following figure shows all the fly-lines for the effective resistance of the ESD device selected in the table. The green fly-lines indicate “pass” while the red fly-lines indicate a “fail” status.

© 2020
All Rights Reserved.

You can also click a pin in the table to view the fly-line for that specific pin to the ESD point.

Related Topics

- Performing ESD Effective Resistance Checks in Voltus-Fi-XL (video)

© 2020 All Rights Reserved.

ESD Current Density Checks

ESD Current Density Checks Flow

The following diagram illustrates the high-level flow of ESD current density checks in Voltus-Fi-XL:

Key Features of ESD Current Density Checks

The following features are supported for the ESD current density (CD) checks in Voltus-Fi-XL:

- **Common EMIR Configuration File**: An option is provided to specify a common configuration (conf) file to be used for all zap simulations. This configuration file contains common settings, for example, geo-unit.
- **CDL Processing for ESD Analysis**: The CDL ESD cells can be specified in the Voltus-Fi GUI for ESD analysis. You can specify multiple ESD cell names in the *Cells*

© 2020 All Rights Reserved.

field in the *CDL* group box of the ESD form. Multiple cell names can be specified separated by a space. This input is optional.

Voltus-Fi automatically finds all the leaf-level instances present inside the specified ESD CDL cells. The leaf-level instances present inside the CDL cells are matched inside the DSPF and used for ESD analysis. For example, you have specified `ESD_cell1` and `ESD_cell2` in the schematic CDL netlist. The figures below show the top-level circuit before and after the CDL cells are processed.

© 2020
All Rights Reserved.

Top-Level Circuit before CDL Processing

© 2020
All Rights Reserved.

Top-Level Circuit after CDL Processing

- **Waveform Analysis**: Voltus-Fi GUI supports waveform analysis when dynamic ESD zap simulation is run. The waveform display is provided using the layout window of the Virtuoso Visualization and Analysis XL software.

Types of Analysis for ESD Current Density Checks

There are different types of ESD analyses for current density checks as per the industry standards. Voltus-FI supports static and dynamic ESD analyses.

- **Static ESD Analysis**: It uses the DC simulation models for current density checks.
- **Dynamic ESD Analysis**: It can be either human body model (HBM) or charged device model (CDM). While running ESD analysis, Voltus-Fi-XL creates an input stimulus that

© 2020 All Rights Reserved.

is injected into the pin of the circuit. The nature of the input stimulus is similar to an actual ESD event that occurs in a real-life event.

The figure below shows the overview model of creating a stimulus and injecting into the pin for dynamic ESD analysis.

The testbench creates a stimulus that models the HBM or CDM event.

- **HBM ESD Analysis**: The human body model (HBM) analysis captures the ESD event in the pin of an IC occurring from the human touch, such as touching with a finger. The figure below shows the characteristic of the input stimulus that occurs in the circuit pin. The peak current in this event is a few amperes (A). The rise time is less than 10ns and

© 2020 All Rights Reserved.

the decay time is approximately 150ns. Voltus-Fi creates an HBM stimulus internally using an RLC circuit.

- **CDM ESD Analysis**: The charged device (CDM) model captures the ESD event that occurs when a chip is mounted during assembly and manufacturing. For example, a package may be inductively charged up when it is transported along a conveyor belt. Then the discharge may happen when a pin comes in contact with any metal. The figure below shows the characteristic of the input stimulus. The CDM event is known to produce

© 2020 All Rights Reserved.

a spike of currents, in the range of peak `50A`, and have a sub-nanosecond rise and decay time.

The CDM events are generated when the package is inductively charged up, therefore, it is important that this testing model should include the substrate network in effect during simulation. The CDM testbench created by Voltus-Fi automatically takes into account the substrate network to inject high current into the circuit.

It is also important to capture package parameters in CDM analysis. The testbench contains package RLC parameters. Voltus-Fi lets you control several characteristics of the stimulus such as peak current, transient period, package effect, and so on. Voltus-Fi can internally generate CDM stimulus to a circuit.

Prerequisites for ESD Current Density Checks

The following table lists the required inputs for performing ESD current density checks in Voltus-Fi-XL.

Input	Details
Inputs required for ESD analysis	

© 2020
All Rights Reserved.

Input	Details
DSPF file	It provides the complete parasitic information of the design. This file is generated using Quantus.
ESD model information	It is included in the DSPF file.
Net names	Power/ground/signal nets that are required for pin information.
Technology file or qrcTechFile	It is required for reading the technology data for ESD analysis.
Layer map file	It provides mapping between the layer names in the DSPF file and the technology file.
ESD current	ESD static current to be applied for simulation.
ICT file with EM limits	It specifies the process and EM model information for EM analysis.
Models	Spectre model statements.
Circuit Design Language (CDL) file	The schematic netlist.
Inputs required for testbench settings	
Testbench file	It is specified only for the in-house testbench flow. Voltus-Fi-XL can generate a testbench for ESD analysis. The testbench produces stimulus for the purpose of injection into the pin. With this option of specifying the testbench file, you can specify your own testbench in either Spice or Spectre format. If you provide a testbench, Voltus-Fi-XL does not generate any testbench for ESD analysis.
Package parameters file	It specifies the package parameters for the tool-generated testbench flow. The testbench generated by Voltus-Fi-XL considers package RLC values (resistance, inductance, and capacitance, respectively). You can specify package parameters to be used in the testbench by specifying the package parameters file.

© 2020 All Rights Reserved.

Setting Quantus for xDSPF Generation

For ESD analysis, while performing extraction in the batch mode, ensure that the `-reduction` option of the `extract_xdspf` command is set to `false`. This disables the `reduce_i_cards` option in the CCL file while generating the xDSPF file.

While performing extraction in the GUI mode, ensure that the *Reduce I-Cards* option is disabled. This disables the `reduce_i_cards` option in the CCL file.

This is done because if a net is “capacitance only”, it is qualified for reduced I card or instance card treatment. If the `reduce_i_cards` option is set to `true`, the instance cards and their corresponding dummy resistors are not printed in the results.

Figure 11-1 Placing ESD Devices for ESD Protection

The current density check flow can be summarized in the following manner:

- The ESD intended paths for each ESD event (zap) are simulated.
- The results for each zap are displayed.
- The current values at all ESD instances are reported by Voltus-Fi-XL.
- EM analysis can be performed for ESD current paths for every ESD event.

© 2020 All Rights Reserved.

ESD Modeling

Transmission Line Pulse (TLP) modeling is required for running current density checks. The following types of modeling of ESD devices are supported in Voltus-Fi:

- **Spectre/Spice Models**: This mode is used in both static and dynamic ESD analysis. In this mode, Voltus-Fi does not generate any ESD models. It uses the ESD models present in the models library specified for simulation. These may be general Spice models.
- **Vt, Ron, Roff Parameter Models**: This mode is used in DC simulation mode (static ESD analysis). In this mode, Voltus-Fi generates ESD models for MOS and diode devices based on the following parameters:

  $R_{on}+$ $R_{on}-$: On-resistance voltage

  $R_{off}+$ $R_{off}-$: Off-resistance voltage

  $V_{t1}+$ $V_{t1}-$: Trigger voltage

  The diagrams below provide details about how the ESD models are generated for two- and three-terminal devices:

  Two-Terminal Diode ESD Device

© 2020
All Rights Reserved.

Three-Terminal ESD Device

- **TLP-IV Models**: This mode is used in CDM and HBM simulation modes (dynamic ESD analysis). In this mode, Voltus-Fi generates ESD models for MOS and diode devices based on the snapback IV (current voltage) parameters defined in the model data file.

  In this type of modeling, there are two options:

  - **Models generated using user-provided IV data**: If IV data is present for ESD devices, specify it in the *TLP IV Model File* field and click *Load Cell IV Data from File*.

    The TLP data file format is as follows:

    ```
    DeviceName IV={[L/W/Area] I1,V1 I2,V2 I3,V3 …}
    <ESD MOS name> mos IV=(L=1e-05; W=2e-05; 0.00025,1.25; 0.00035,1.215;
    0.005,1.2325; )
    <ESD Diode Name> pdiode/ndiode IV=(Area=1e-05; 0.00025,1.25;
    0.00035,1.215; 0.005,1.2325; )
    ```

    Each line specifies the MOS or diode name (`pdiode/ndiode`) along with the IV data. In addition to specifying the IV data, you can also optionally specify the `L` (Length), `W` (Width), and `A` (Area).

    Voltus-Fi supports linear scaling of the IV data (current values). When a device is instantiated during the Spectre simulation, the current scaling feature allows you to scale the current values provided in the TLP data file.

    The IV data scaling for MOS is based on `W` (width), and for diode is based on `A` (area). L does not affect the current change. Depending on the `W` value for ESD protection clamps/MOS devices and the `A` value for ESD protection diode devices,

© 2020 All Rights Reserved.

the current values provided in TLP IV parameters are linearly scaled based on the ratio of:

- Scale factor for Clamps/ MOS = (`W` of device in DSPF)/ (`W` of device in TLP IV File)
- Scale factor for Diodes = (`A` of diode in DSPF)/ (`A` of diode in TLP IV File)

An example of IV data scaling is as follows:

Following are the IV parameters specified in the TLP IV file:

```
nch_18_mac IV=(L=0.18 W=300; 0,0 0.01,2 0.02,3 0.03,4 0.1,2.5 0.5,3
1.2,3.5)
Diode_18   IV=(Area=100; 0,0 0.2,1 0.5,2 0.8,3 1.1,4)
```

Depending on the `W` and `A` of devices in DSPF, the current values will be scaled to:

```
nch_18_mac  IV=(L=0.18 W=600; 0,0 0.02,2 0.04,3 0.06,4 0.2,2.5 1,3 2.4,3.5)
Diode_18    IV=(Area=50; 0,0 0.1,1 0.25,2 0.4,3 0.55,4)
```

- **Models generated using no IV data**: If no IV data is present for ESD devices, ensure that *TLP-IV Models* is selected in the *Model Type* drop-down and leave the above *TLP IV Model File* field empty. Voltus-Fi will use heuristics to generate models for the ESD device.

Specifying ESD Models

To specify a model type for ESD analysis, do the following:

1. Open the *ESD Model* tab of the ESD form.

© 2020 All Rights Reserved.

2. Select an option from the *Model Type* list.

ESD Testbench

Testbench specification is needed when you run current density checks for the CDM and HBM simulation modes (dynamic ESD analysis). The testbench consists of resistance, inductance, and capacitance (RLC) circuits that create a stimulus in the zap pins for CDM and HBM circuit testing. In Voltus-Fi GUI, you can specify either an in-house testbench or a tool-generated testbench for ESD analysis.

© 2020
All Rights Reserved.

- **Tool-Generated Testbench**: Voltus-Fi generates a testbench for CDM and HBM analysis. The RLC circuit of the testbench generates a stimulus to the input zap pin. You can change the nature of the stimulus waveform by changing the RLC values in the testbench circuit.
- **In-House Testbench**: You can specify an in-house testbench if you do not want to use the Voltus-Fi-generated HBM or CDM testbench.

Specifying ESD Testbench

You can specify either an in-house testbench or a tool-generated testbench for CDM and HBM analysis.

Specifying In-House Testbench

To specify an in-house testbench, do the following:

1. Open the *ESD Run* tab of the ESD form.
2. Click any of the tabs: *CDM Mode* or *HBM Mode*.
3. Click *CDM Testbench Settings* or *HBM Testbench Settings*.

© 2020 All Rights Reserved.

The Testbench Settings form opens.

4. Select *Load In-House Testbench*.
5. Select an option from the *Testbench Format* list.
6. Specify a value in the *TestBench File* box.
7. Click *Ok* or *Apply*.

Specifying Tool-Generated Testbench

To specify a tool-generated testbench, do the following:

1. Open the *ESD Run* tab of the ESD form.
2. Click any of the tabs: *CDM Mode* or *HBM Mode*.
3. Click *CDM Testbench Settings* or *HBM Testbench Settings*.

© 2020
All Rights Reserved.

The Testbench Settings form opens.

4. Select *Load Tool Generated Testbench*.
5. Specify values for the following or click *Defaults* to enter the default values:

   CDM Mode: *LO* (inductance), *RO* (resistance), *CO* (capacitance), and *C1* (substrate effect)

   HBM Mode: *RO* (resistance), *R1* (resistance), and *CO* (capacitance)

   Alternatively, specify a value in the *Load from File* box to load values of the above parameters from a file.
6. Click *Ok* or *Apply*.

ESD Zaps

In Voltus-Fi GUI, the *Zap Simulation Table* displays zap details for current density checks. Each row in the table specifies a zap and displays information about the polarity of the

© 2020 All Rights Reserved.

injection current (positive or negative), the name of the static current injection pin, the reference ground pin, and the simulation status.

There are two options for adding zaps in the GUI: tool-generated zaps and custom zaps.

Tool-generated Zaps

These zaps are generated by default. They are created from the power nets, ground nets, and signal nets specified in the *ESD Setup* tab. The table in the *Tool Gen Zaps* tab is automatically populated with information of all zaps.

Custom Zaps

You can add custom zaps in the table by specifying their current/voltage polarity, the injection pin, and reference pin information. This option is used in the following scenarios:

- To short multiple injection and reference pins in one zap during simulation.
- To perform ESD checks on a pair of injection and reference pins that are not generated by default.

Creating Custom ESD Zaps

You must specify the custom zaps and then load the DSPF file again to save the changes. To create custom zaps, do the following:

1. Open the *ESD Run* tab of the ESD form.
2. Click any of the tabs: *DC Mode*, *CDM Mode*, or *HBM Mode*.

© 2020 All Rights Reserved.

3. In the *Zap Simulation Table*, click *Custom Zaps*.

4. Click the *Add Row* (+) button.

   A row is added to the table.

5. Specify a value in the *Injection Polarity* column: POSITIVE or NEGATIVE.

6. Specify values in the *Injection Pin* (static current injection pin) and *Reference Pin* (reference ground pin) columns.

   You can specify multiple pins separated by a space. Wildcards are not supported. When multiple pins are specified as either injection or reference, these pins are shorted internally during simulation.

7. On the *ESD Setup* tab, click *Load DSPF* to create a zap using these inputs. The inputs are saved.

Running Current Density Checks

The ESD Analysis GUI in Voltus-Fi-XL lets you set up the ESD current density analysis, specify the ESD devices, select the ESD instances, select the simulation job, and view the ESD current density reports. The GUI provides the following features to analyze currents:

- Slider with filter range
- Annotation Browser with markers
- Layer selection to view resistors

There are two use models for current density checks in the GUI. The first, in which Spice models are provided and the second, in which the Spice models are not provided. In the latter case, the tool generates TLP models for ESD analysis.

© 2020 All Rights Reserved.

The high-level GUI flow for ESD current density checks for the two use models is detailed in the flow diagram below.

Figure 11-2 ESD Current Density Checks Flow in Voltus-Fi-XL GUI

You can run current density checks in both batch and GUI modes.

Batch Mode

To run current density checks in batch mode, do the following:

- Provide information about the `analyze_esd_current_density` command in the command file, which is specified using the `vfibatch` command.

  ```
  vfibatch -cmd cmd-file-name
  ```

GUI Mode

In GUI mode, you can run current density checks for the following: DC simulation mode (static ESD analysis), CDM and HBM simulation modes (dynamic ESD analysis), and PERC. The steps are detailed below.

© 2020 All Rights Reserved.

Running Current Density Checks for DC Mode

To run current density checks for the DC simulation mode, do the following:

1. In the *ESD Setup* tab of the ESD form, specify the ESD analysis settings and load the DSPF file.
2. On *ESD Model* tab, select *Static ESD Device Models*.
3. Select an option from the *Model Type* list: *Spectre/Spice Models* or *Vt/Ron/Roff Models*.
4. Open the *ESD Run* tab.

5. Click the *DC Mode* tab.
6. Specify whether simulation should be run by creating voltage source or injecting current by selecting the *DC Voltage (V)* or *DC Current (A)* options, respectively. The default voltage is `200 Volts` and the default current is `6 Ampere`.
7. In the *DC Zap Simulation Table*, specify the zaps for simulation. Either select the tool-generated zaps or create custom zaps.

© 2020
All Rights Reserved.

8. Click *Start Simulation*.

Running Current Density Checks for CDM Mode

To run current density checks for the CDM simulation mode, do the following:

1. In the *ESD Setup* tab of the ESD form, specify the ESD analysis settings and load the DSPF file.
2. On *ESD Model* tab, select *Dynamic ESD Device Models*.
3. Select an option from the *Model Type* list: *Spectre/Spice Models* or *Vt/Ron/Roff Models*.
4. Open the *ESD Run* tab.

5. Click the *CDM Mode* tab.
6. Specify whether simulation should be run by creating voltage source or injecting current by selecting the *CDM Voltage (V)* or *CDM Current (A)* options, respectively. The default voltage is `300 Volts` and the default current is `6 Ampere`.

© 2020
All Rights Reserved.

7. Specify a value in nanosecond (ns) for *Transient Time*.
8. Click *CDM Testbench Settings* to specify a testbench.
9. In the *CDM Zap Simulation Table*, specify the zaps for simulation. Either select the tool-generated zaps or create custom zaps.
10. Click *Start Simulation*.

Running Current Density Checks for HBM Mode

To run current density checks for the HBM simulation mode, do the following:

1. In the *ESD Setup* tab of the ESD form, specify the ESD analysis settings and load the DSPF file.
2. On *ESD Model* tab, select *Dynamic ESD Device Models*.
3. Select an option from the *Model Type* list: *Spectre/Spice Models* or *Vt/Ron/Roff Models*.
4. Open the *ESD Run* tab.

© 2020
All Rights Reserved.

5. Click the *HBM Mode* tab.
6. Specify whether simulation should be run by creating voltage source or injecting current by selecting the *HBM Voltage (V)* or *HBM Current (A)* options, respectively. The default voltage is `2000 Volts` and the default current is `1.6 Ampere`.
7. Specify a value in nanosecond (ns) for *Transient Time*.
8. Click *HBM Testbench Settings* to specify a testbench.
9. In the *HBM Zap Simulation Table*, specify the zaps for simulation. Either select the tool-generated zaps or create custom zaps.
10. Click *Start Simulation*.

PERC – Programmable Electrical Rule Check (PERC) Interface

Click *Launch PERC* to launch the Cadence® Pegasus™ or PVS PERC (Programmable Electrical Rule Check) interface.

Note: PERC performs electrostatic discharge, multi-power domains, and advanced electrical rule checks. The violations of these rules can affect the functionality of the electrical equipment. With PERC, these violations are caught at early design development stages. Pegasus PERC provides a functionality to code various checks, which take into account the following information:

- Device connectivity
- Device property
- Design hierarchy
- User-defined grouping of the nets by net-tag or voltage attribute

You can run Pegasus or PVS PERC on the schematic or layout design. For details, see the "Running PERC" chapter in the *Pegasus Interactive User Guide* or *Virtuoso IPVS User Guide*.

© 2020 All Rights Reserved.

Performing Simulation

Once all the information for the zaps is provided, you can perform the simulation as follows:

1. Select *Start Simulation* to start the simulation for a specific zap. The zap is selected in the table. The simulation runs in the background.
2. Select *Check HeartBeat* to check the status of the simulation being performed. A pop-up window named, *Simulation status*, opens. Once the simulation is complete, the Simulation Status column in the table shows the status as completed.

Figure 11-3 Checking the Simulation Status

3. Click *Stop Simulation* to stop the simulation for all the zaps running in the background.

© 2020
All Rights Reserved.

4. Click *Select or Deselect All Simulation jobs* to simulate ESD events for all or selected power/ground/signal-PAD zap combinations. By default, all combinations are selected.

Figure 11-4 Deselecting All Simulation Jobs

5. Click *ESD CD Report* to view the ESD current density pass/fail report for all instances. For details about the report format and a sample report, see Current Density Check Reports.

For each zap, you can perform the following actions using the buttons provided in the table:

- Loading the Simulation Results
- Viewing the ESD Current Reports
- Viewing Waveforms for Dynamic Simulation
- Viewing the Simulation Log Information
- Viewing the Device Ron Information

© 2020 All Rights Reserved.

Loading the Simulation Results

Click *Load Results* to load the results of the simulation in Voltus-Fi-XL. When you click this option, the EM tab of the IR/EM Results form opens.

Figure 11-5 Loading the Results in Voltus-Fi-XL

Viewing the ESD Current Reports

Click *ESD Currents* to view the report for the ESD currents of all ESD instances for the specific zap. This report shows the peak current through each terminal of the ESD instances on the simulated net.

© 2020 All Rights Reserved.

Figure 11-6 Sample Current Report for ESD Terminal

ESD Devices	Current (Amps)	Voltage (Volts)	Layer Name
MX772/X0/M0#d	1.594e-03	3.871e-01	cont_ndiff
MX772/X0/M0#g	0.000e+00	0.000e+00	mwires
MX772/X0/M0#s	-8.274e-01	1.063e+01	cont_ndiff
MX771/X0/M0#d	1.649e-03	2.480e-02	cont_ndiff
MX771/X0/M0#g	0.000e+00	0.000e+00	mwires
MX771/X0/M0#s	-5.648e-01	1.102e+01	cont_ndiff
DX829/X1/D0#PLUS	4.564e-04	4.206e-03	cont_pdiff
DX829/X1/D0#MINUS	0.000e+00	4.564e-07	dwires
DX829/X0/D0#PLUS	2.303e-04	5.285e-01	cont_pdiff
DX829/X0/D0#MINUS	0.000e+00	2.969e-02	dwires
DX828/X1/D0#PLUS	4.572e-04	4.210e-03	cont_pdiff
DX828/X1/D0#MINUS	0.000e+00	4.572e-07	dwires

Viewing Waveforms for Dynamic Simulation

Click *Show Waveform* to view the waveforms for the dynamic zap simulation. The waveform opens in Virtuoso® Visualization and Analysis XL.

© 2020 All Rights Reserved.

Figure 11-7 Viewing the Waveform for Dynamic Simulation Analysis

Viewing the Simulation Log Information

Click *View Log* to view the simulation log file while the simulation is running.

© 2020 All Rights Reserved.

Figure 11-8 Viewing the Log File

```
Spectre (R) Circuit Simulator
Version 18.1.0.mid.398 64bit -- 10 Jun 2019
Copyright (C) 1989-2019 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence and Spectre are registered tradem

Includes RSA BSAFE(R) Cryptographic or Security Protocol Software from RSA Security, Inc.

User: anuk   Host: sjfqos019   HostID: 12AC2F23   PID: 28196
Memory  available: 52.5689 GB  physical: 406.3835 GB
Linux   : Red Hat Enterprise Linux Workstation release 6.5 (Santiago)
CPU Type: Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz
All processors running at 3000.1 MHz
        Socket: Processors
        0:       0,  2,  4,  6,  8, 10, 12, 14, 16, 18
        1:       1,  3,  5,  7,  9, 11, 13, 15, 17, 19

System load averages (1min, 5min, 15min) : 199.5 %, 198.8 %, 198.7 %
HPC is enabled


Simulating `ESD_HBM.sp' on sjfqos019 at 1:41:39 AM, Tue Jun 11, 2019 (process id: 28196).
Current working directory: /home/anuk/VoltusFi_workshop_opamp_ESD_workshop/CD_static_simulation/temp_dir/HBM/POS_AVDD_AVSS
Command line:
     \
        /vols/mmsim/cm/1810_isr/dev/install/lnx86/latest/tools.lnx86/bin/spectre  \
        -64 ESD_HBM.sp +aps +emir=ESD.conf -outdir ./

Loading /vols/mmsim/cm/1810_isr/dev/install/lnx86/latest/tools.lnx86/cmi/lib/64bit/5.0/libinfineon_sh.so ...
Loading /vols/mmsim/cm/1810_isr/dev/install/lnx86/latest/tools.lnx86/cmi/lib/64bit/5.0/libphilips_o_sh.so ...
Loading /vols/mmsim/cm/1810_isr/dev/install/lnx86/latest/tools.lnx86/cmi/lib/64bit/5.0/libphilips_sh.so ...
Loading /vols/mmsim/cm/1810_isr/dev/install/lnx86/latest/tools.lnx86/cmi/lib/64bit/5.0/libsparam_sh.so ...
Loading /vols/mmsim/cm/1810_isr/dev/install/lnx86/latest/tools.lnx86/cmi/lib/64bit/5.0/libstmodels_sh.so ...
Reading file:  /home/anuk/VoltusFi_workshop_opamp_ESD_workshop/CD_static_simulation/temp_dir/HBM/POS_AVDD_AVSS/ESD_HBM.sp

Current resident memory used = 57.11 Mbytes.


Reading EMIR configuration file:  /home/anuk/VoltusFi_workshop_opamp_ESD_workshop/CD_static_simulation/temp_dir/HBM/POS_AVDD_

~~~~~~~~~~~~~~~~~~~~~~~~~~~
EMIR Analysis Configuration
~~~~~~~~~~~~~~~~~~~~~~~~~~~

include         /home/anuk/VoltusFi_workshop_opamp_ESD_workshop/CD_static_simulation/temp_dir/DiffOpAmp.dspf.emir_conf
nets            name=[XO.*] analysis=[vmax imax sigvmax vavg iavg] reftype=[max]
emirutil        techfile = /home/anuk/VoltusFi_workshop_opamp_ESD_workshop/qrcTechFile
```

Viewing the Device Ron Information

Click *Device Ron* to view the *ESD Device Ron* pop-up window. The device Ron report is generated using the DC mode. This window displays the name of the *Trigger pin*, *Reference pin*, *Total R* and the *Polarity*. Clicking on any instance in this window, highlights the instance on the layout. Click *Save* to save the report in a text file.

Note: For viewing the Ron Reports in CDM/HBM simulation, first generate the report using the DC simulation mode.

© 2020 All Rights Reserved.

Figure 11-9 Viewing the Device Ron Report

ESD Devices	Device Type	Ron (Ohms)	W	L	Nfin	X	Y
MX771/X0/M0	g45n2nvt	10	1.2e-06	8e-07	1	-23	48.9
MX772/X0/M0	g45n2nvt	7.14	1.2e-06	8e-07	1	-22.7	85.7
DX828/X0/D0	g45pd2svt	NA	0	0	1	20	22
DX828/X1/D0	g45pd2svt	NA	0	0	1	20.8	33.1
DX829/X0/D0	g45pd2svt	NA	0	0	1	52.8	23.5
DX829/X1/D0	g45pd2svt	NA	0	0	1	53.5	34.6

Rule-Based ESD Checks

Voltus-Fi performs rule-based resistance and current density checks. You can specify the rules for computing effective resistance and performing current density or EM analysis in a single rule file. Voltus-Fi runs resistance and current density checks on the specified nets and clamps for the rules specified in the rule file, and generates a pass/fail summary report and a detailed report for each rule. In this flow, extraction is run only once for all methods, thereby saving time and memory.

Voltus-Fi performs the following types of ESD checks:

- **Pin-to-clamp**: Effective resistance is computed from a pin to all clamps or ESD devices.
- **Clamp-to-clamp**: Effective resistance is computed between all clamps for each net.
- **EM**: Current density checks are performed for the DC Simulation model using current value as the input stimulus.

© 2020 All Rights Reserved.

ESD Rule File

The following section lists the syntax of an ESD rule file, parameters of the rule file, and examples of different rule types.

Rule File Syntax

The rule syntax is not case-sensitive. The syntax of the rule file is:

Resistance Checks

```
rule rule_name
type {clamp2clamp | pin2clamp}
power -nets net1 net2 net3
ground -nets net1 net2 net3
signal -nets net1 net2 net3
clamp_types -from type1 type2 type3 ... -to type1 type2 type3 ...
reff_threshold value
reporting {reverse | both}
end rule
```

EM Checks

```
rule rule_name
type em
source -type {power | ground} -nets net1 net2 net3... -pins pin1 pin2 pin3...
     -report_threshold value
sink -type {power | ground} -nets net1 net2 net3... -pins pin1 pin2 pin3... -
     report_threshold value
clamp_types type1 type2 type3...
current value
end rule
```

Rule File Parameters

© 2020 All Rights Reserved.

Parameter	Description
`clamp_types -from` *`type1 type2 type3 ...`* `-to` *`type1 type2 type3 ...`*	
	Specifies the type of clamp for which the rule is run. The different clamp types are stored in PGV. For `clamp2clamp`, you need to specify the `-from` and `-to` parameters to specify the starting and ending clamp types. For the other types of checks, you need to specify individual clamp cell types. The `-from` and `-to` parameters are not required.
`current` *`value`*	
	Specifies the zap current value. This parameter is applicable only to the `em` rule type.
`ground -nets {`*`net1 net2 net3...`* `| all}`	
	Specifies the list of ground nets to be analyzed. You can specify `all` to include all ground nets.
`power -nets {`*`net1 net2 net3...`* `| all}`	
	Specifies the list of power nets to be analyzed. You can specify `all` to include all power nets.
`reporting {reverse | both}`	
	Specifies the reporting format that can be used for sorting of the rule. The default format is clamp to pin. It checks if any clamp is failing from clamp to pin. ■ `reverse` - reports effective resistance from pin to clamp. It checks if any pin is failing from pin to clamp. ■ `both` - reports effective resistance both from clamp to pin and pin to clamp. This parameter is applicable only to the `pin2clamp` rule type.
`rule` *`rule_name`*	
	Specifies the name of the rule.
`reff_threshold` *`value`*	

© 2020 All Rights Reserved.

Parameter	Description
	Specifies the resistance threshold for the rule to pass (rule pass criteria). A pass status is given when the effective resistance is below the specified threshold value, and a fail status is given when the effective resistance is above the specified threshold value. This parameter is applicable to the `clamp2clamp` and `pin2clamp` rule types.
`signal -nets {net1 net2 net3... | all}`	
	Specifies the list of signal nets to be analyzed. You can specify `all` to include all signal nets.
`sink -type {power | ground} -nets {net1 net2 net3... | all} -pins {pin1 pin2 pin3... | all} -report_threshold value`	
`source -type {power | ground} -nets {net1 net2 net3... | all} -pins {pin1 pin2 pin3... | all} -report_threshold value`	
	For `em` check, source (the point where current is injected), and sink (the point from where the current comes out) details are provided. For both source and sink, you need to specify the type of net (power/ground), the nets to be analyzed, and whether pins are to be shorted. You can specify EM rules for all nets or specific nets with zaps and sinks on pins. In the `em` rule check, the software runs a static simulation and checks current through each metal line against the ESD EM techfile.
`type {clamp2clamp | pin2clamp | em}`	
	Specifies the rule type.

Rule File Examples

The following are examples of different rule types.

Rule for computing clamp-to-clamp resistance

© 2020 All Rights Reserved.

```
Rule c2c_rule1
type clamp2clamp
power -nets AVDD
ground -nets AVSS
signal -nets inm
clamp_types -from "g45pd2svt" "g45n2nvt" -to  "g45pd2svt" "g45n2nvt"
reff_threshold 3
end Rule
```

Rule for computing pin-to-clamp resistance: Example 1

```
Rule p2c_rule1
type pin2clamp
clamp_types "g45n2nvt"
power -nets "AVDD"
ground -nets "AVSS"
signal -nets all
reff_threshold 2.5
reporting both
end rule
```

Rule for computing pin-to-clamp resistance: Example 2

```
Rule p2c_rule2
type pin2clamp
clamp_types "g45n2nvt" "g45pd2svt"
power -nets all
ground -nets all
signal -nets all
reff_threshold 2.5
reporting reverse
end rule
```

EM rule: Example 1

© 2020 All Rights Reserved.

```
Rule em_rule1
type em
source -type power -nets AVDD inm
sink -type ground -nets AVSS inp
clamp_types "g45n2nvt" "g45pd2svt"
current 13
end rule
```

EM rule: Example 2

```
Rule em_rule2
type em
source -type power -nets all -report_threshold 2
sink -type ground -nets all -report_threshold 1
clamp_types "g45n2nvt" "g45pd2svt"
current 6
end rule
```

Running Rule-Based ESD Checks

Batch Mode

To run rule-based checks in batch mode, do the following:

- Provide information about the `analyze_esd_network` command in the command file, which is specified using the `vfibatch` command.

  `vfibatch -cmd` *`cmd-file-name`*

GUI Mode

To run rule-based checks in GUI mode, do the following:

1. In the *ESD Setup* tab of the ESD form, specify the ESD analysis settings and load the DSPF file.

© 2020 All Rights Reserved.

2. Click the *ESD Rule Check* tab.

3. Specify a rule file. Either type the file name in the *Rule File* field or click the *Browse* button to select a file. Use the *Edit* button to edit the file.

4. Click *Load* to load rules from the rule file.

5. Select *Finger Merge* to merge the MOS fingers in the resistance check reports. This option is selected by default.

6. In the *Rule Database* group box, select the rules to be run.

7. Click *Run Rules*.

© 2020 All Rights Reserved.

Batch Mode Support for ESD Analysis

This section includes the following topics:

- Batch Mode Support for Effective Resistance Checks
- Batch Mode Support for Current Density Checks
- Batch Mode Support for Rule-Based Checks

Batch Mode Support for Effective Resistance Checks

In batch mode, the resistance checks are performed using the `analyze_esd_generic` command. This command reports the effective resistance from the power/ ground pins and signal nets to the nearest ESD device.

The syntax of the command is provided below.

```
analyze_esd_generic
     -power_nets {pwr_net_names}
     -ground_nets {gnd_net_names}
     -signal_nets {signal_net_names}
     -pgpin2clamp_threshold threshold_value
     -loop_threshold threshold_value
     -target_models {model_name_1 ron_1 model_name_2 ron_2…}
     -dspf_file filename
     -tech_file file_name
     -output_dir dir_name
     -discard_pin_to_first_ESD_for_signal_net {true|false}
     -esd_instance_name_string {instance_names}
     -merge_fingers {true | false}
     -cdl_file file_name
```

Note: The effective resistance is calculated after taking into account the parasitics in the netlist. The Ron of the ESD device is added to the effective resistance while reporting the loop resistance.

For details of the syntax and examples of the command, see analyze_esd_generic in the “Batch Mode Execution” chapter.

Batch Mode Support for Current Density Checks

The current density checks are performed using batch mode command, `analyze_esd_current_density`. This command reports the current values at all ESD instances.

© 2020 All Rights Reserved.

The syntax of the command is provided below.

```
analyze_esd_current_density
     -esd_conf_file conf_filename
     -run_async {true | false}
     -simulation_mode {DC CDM HBM}
     -zap_injection_current_polarity {positive | negative}
     -zap_injection_pins list_of_pins
     -zap_injection_reference_pins list_of_pins
     -use_remote_machines_as_many_zaps {true | false}
```

For details of the above command, see analyze_esd_current_density in the “Batch Mode Execution” chapter.

Batch Mode Support for Rule-Based Checks

Rule-based ESD checks are performed using the batch mode command, `analyze_esd_network`. This command performs resistance and current density checks based on the set of rules defined in the rule file.

The syntax of the command is provided below.

```
analyze_esd_network
     -power_nets {pwr_net_names}
     -ground_nets {gnd_net_names}
     -signal_nets {signal_net_names}
     -target_models {model_name_1 ron_1 model_name_2 ron_2…}
     -dspf_file filename
     -output_dir dir_name
     -merge_fingers {true | false}]
     -tech_file file_name
     -discard_pin_to_first_ESD_for_signal_net {true | false}
     -rule_file file_name
     -model_file file_name
     -dfIIlayermap_file file_name
```

For details of the syntax and examples of the command, see the analyze_esd_network command in the “Batch Mode Execution” chapter.

The following data for the ESD analysis is read from the ESD conf file specified using the `analyze_esd_generic` command:

- The power/ground/signal nets: The `-power_nets`, `-ground_nets`, and `-signal_nets` options specify the names of the power/ground/signal nets for which the ESD current density check is to be performed. For details, see Specifying the Power/Ground/Signal Nets.

© 2020 All Rights Reserved.

- DSPF file: The `-dspf_file` option specifies the name of the DSPF file that is generated using Quantus. This file contains information about model names, including the ESD model names required for the ESD analysis
- Technology file: The `-tech_file` option specifies the technology file containing the EM limits for ESD validation.
- Model file: The `-model_file` option specifies the Spice model file that contains references to all the Spice models required to simulate the DSPF circuit.
- Output directory: The `-output_dir` option specifies the name of the output directory in which the reports generated for the ESD analysis are written.

Specifying the Power/Ground/Signal Nets

For the following three options, `-power_nets, -ground_nets`, and `-signal_nets` options, keep the following in mind:

- The options support wild-card UNIX style naming.
- If a specified net name is not present in the DSPF, the net name is ignored and no zap combinations are generated for the missing net. Also, if a pin is present in the input DSPF file and is not specified using these three options, Voltus-Fi displays a message listing the missing pins that are not being ESD analyzed.
- For all three options, pins can be specified using the standard regular UNIX style expressions. For example, specify '*' to specify all pins for an option. While using regular expression, it is possible that after regular expressions (regex) are resolved, the same pins appear in more than one of the three options. In such cases, preference is given in the following order; power, ground, and then signal. This ensures that after regex resolution, the same pin does not appear in more than one option.

For example, if you specify '*' for all three options; `-power_nets, -ground_nets`, and `-signal_nets`, then all pins will be considered as power-pins.

© 2020 All Rights Reserved.

Using Distributed Processing in Voltus-Fi-XL

The computation of resistance and current density checks is CPU-intensive. Voltus-Fi supports distributed processing to perform these checks.

Use the following commands to set up the ESD current-density distributed processing:

- To set up a run with multiple processes on the same machine, use the following command:

  ```
  setMultiCpuUsage -localCpu 10
  ```

  The above command will set up the ESD-CD to run ten jobs in parallel.

- To set up a run with multiple processes in LSF (Load Sharing Facility), use the following set of commands:

  ```
  setMultiCpuUsage -localCpu 10 -remoteHost 2
  ```

  ```
  setDistributeHost -lsf -queue vormetric -args {-P lnx64}
  -resource {OSREL==EE50}
  ```

  The above commands will set up the ESD-CD to run two jobs in parallel in the LSF farm as per the queue and other specifications provided using the `setDistributeHost` command.

  For detailed information about the above commands, see analyze_esd_generic in the “Batch Mode Execution” chapter.

Outputs of ESD Analysis

The following reports are generated for ESD Analysis in Voltus-Fi-XL:

- Effective Resistance Check Reports
- Current Density Check Reports
- Rule-Based Check Reports

Effective Resistance Check Reports

The following reports are generated for effective resistance checks:

- `The Pin to Clamp Report`
- `The Clamp-to-Clamp Report`

© 2020 All Rights Reserved.

- The Pin-to-Pin Report

The report formats and samples are provided below.

The Pin to Clamp Report

In this report, the effective resistance is reported for each pin and all ESD devices.

Sample filename: `pin2clamp.rpt`

Format:

```
Net name
Node Name         Reffective          Pass/Fail
ESD Macro Device
```

Format Description:

`Net Name` specifies the net names for the terminals/nodes for which the effective resistance is being reported.

`Node Name` specifies the terminal names of the ESD device.

`Reffective` specifies the effective resistance of the terminal.

`Pass/Fail` specifies the status of the terminal – a `pass` status is given when the effective resistance of the terminal is below the specified threshold value and a `fail` status is given when the effective resistance of the terminal is above the specified threshold value.

`ESD Macro Device` specifies the name of the ESD device for which the effective resistance is being reported.

A part of a sample report is shown below.

© 2020 All Rights Reserved.

Figure 11-10 The Pin-to-Clamp Sample Report

```
****************** ESD Resistance checks ******************************
 Power Nets : VDDIO
 Ground Nets : VSSIO
 Signal Nets : PAD_1 PAD_2
 Threshold : 10.000000
 Finger-merging : false
 Report created : 2017-Feb-08 12:07:37 (2017-Feb-08 11:07:37 GMT)
 Host name : euvclo39
*********************************************************************

 Net : VDDIO

        ----------------------------------------------------------------
                       Node Name            Reffective       Pass/Fail
        ----------------------------------------------------------------
        -------------------
        ESD macro device : g45n2svt
        -------------------
                   MXu_clamp1/MM0#d           3.280230            pass
                   MXu_clamp1/MM0#g          17.880000            fail
                   MXu_clamp1/MM0@1#d         3.281230            pass
                   MXu_clamp1/MM0@1#g        20.386700            fail
                   MXu_clamp1/MM0@10#d        2.629670            pass
                   MXu_clamp1/MM0@10#g       20.707200            fail
                   MXu_clamp1/MM0@11#d        2.630670            pass
                   MXu_clamp1/MM0@11#g       24.079700            fail
                   MXu_clamp1/MM0@12#d        2.499250            pass
                   MXu_clamp1/MM0@12#g       20.843700            fail

        -------------------
        ESD macro device : g45p2svt
        -------------------
                        MXu_pad1/MM0#g      105.279000            fail
                        MXu_pad1/MM0#s        0.905011            pass
                      MXu_pad1/MM0@1#g       17.998500            fail
                      MXu_pad1/MM0@1#s        0.914313            pass
                     MXu_pad1/MM0@10#g       19.206600            fail
                     MXu_pad1/MM0@10#s        0.904510            pass
                     MXu_pad1/MM0@11#g       95.324500            fail
                     MXu_pad1/MM0@11#s        0.948836            pass
                     MXu_pad1/MM0@12#g       96.526200            fail
                     MXu_pad1/MM0@12#s        0.945040            pass
                     MXu_pad1/MM0@13#g       18.004400            fail
```

The Clamp-to-Clamp Report

In this report, the effective resistance is reported between two clamps for every net.

Sample filename: `clamp2clamp.rpt`

Format:

```
Net name
Clamp_terminal          Reffective          Pass/Fail
Clamp:
```

Format Description:

© 2020 All Rights Reserved.

`Net Name` specifies the net name for the terminals for which the effective resistance is being reported.

`Clamp-terminal` lists the names of all terminals to which Reffective is reported.

`Reffective` specifies the effective resistance of the terminal.

`Pass/Fail` specifies the status of the terminal - a `pass` status is given when the effective resistance of the terminal is below the specified threshold value and a `fail` status is given when the effective resistance of the terminal is above the specified threshold value.

`Clamp` specifies the name of the clamp terminal from which the effective resistance is being reported.

Figure 11-11 The Clamp-to-Clamp Sample Report

```
******************* ESD Resistance checks **************************
 Power Nets : VDDIO
 Ground Nets : VSSIO
 Signal Nets : PAD_1 PAD_2
 Threshold : 10.000000
 Finger-merging : true
 Report created : 2017-Jun-27 22:32:15 (2017-Jun-28 05:32:15 GMT)
 Host name : sjfnl166
*******************************************************************************

NET     :                  VDDIO

                                 Clamp-terminal      Reffective       Pass/Fail
                 ----------------------------------------

         Clamp   :       MXu_clamp1/MM0#d

                               MXu_clamp1/MM0#g       23.211700            fail
                               MXu_clamp2/MM0#d        2.419820            pass
                               MXu_clamp2/MM0#g       20.503500            fail
                                 MXu_pad1/MM0#g       18.004900            fail
                                 MXu_pad1/MM0#s        0.908434            pass
                                 MXu_pad1/MM1#g       17.846800            fail
                                 MXu_pad1/MM1#s        0.863932            pass
                                 MXu_pad2/MM0#g       19.315200            fail
                                 MXu_pad2/MM0#s        1.007880            pass
                                 MXu_pad2/MM1#g       21.622500            fail
                                 MXu_pad2/MM1#s        0.940568            pass
         Clamp   :       MXu_clamp1/MM0#g

                               MXu_clamp1/MM0#d        2.652340            pass
                               MXu_clamp2/MM0#d        2.707060            pass
                               MXu_clamp2/MM0#g       20.790700            fail
                                 MXu_pad1/MM0#g       18.004900            fail
                                 MXu_pad1/MM0#s        0.908445            pass
```

The Pin-to-Pin Report

In this report, the total effective resistance is reported for all the paths, from the power pin to the ESD device to the ground pin.

© 2020 All Rights Reserved.

Sample filename: `pin2pin.rpt`

Format:

```
Net name
Effective Resistance to Ground Net
pin-ESD-reff    ESD-terminal  ESD-ESD-reff  ESD-terminal  ESD-pin-reff total-reff
--------------------
Pin1
Pin2
--------------------
```

Format Description:

`Net Name` specifies the net name for terminals for which the effective resistance is being reported.

`Effective Resistance to Ground Net` specifies the total effective resistance from the power/signal pin to the ground pin.

`pin-ESD-reff` specifies the effective resistance between the power pin and the ESD device.

`ESD-terminal` specifies the name of the ESD terminal.

`ESD-ESD-reff` specifies the effective resistance of the terminal.

`ESD-pin-reff` specifies the effective resistance between the terminal and the ground pin.

`total-reff` specifies the total effective resistance from the power pin to ESD to ground pin.

`Pin1` specifies the “from” pin.

`Pin2` specifies the “to” pin.

The table below the pins shows all paths from `Pin1` to `Pin2`.

© 2020 All Rights Reserved.

Figure 11-12 Pin-to-Pin Sample Report

```
******************* ESD Resistance checks **************************
 Power Nets : VDDIO
 Ground Nets : VSSIO
 Signal Nets : PAD_1 PAD_2
 Threshold : 10.000000
 Finger-merging : true
 Report created : 2017-Jun-27 22:32:15 (2017-Jun-28 05:32:15 GMT)
 Host name : sjfnl166
*******************************************************************************


        pin-ESD-reff  ESD-terminal      ESD-ESD-reff    ESD-terminal    ESD-pin-reff  total-reff
      ---------------------------------------------------------------------------------------------


---------------------------
Pin1     :    PAD_1
Pin2     :     VSSIO
Max total-reff    :    3.21
Status    :   pass
---------------------------
         0.86    MXu_pad1/MM2#d              1.28    MXu_pad1/MM3#d            1.07    3.21
         0.86    MXu_pad1/MM2#d              0.00    MXu_pad1/MM2#s            1.07    1.93

---------------------------
Pin1     :    PAD_2
Pin2     :     VSSIO
Max total-reff    :    6.07
Status    :   pass
---------------------------
         2.14    MXu_pad2/MM0#g              2.74    MXu_clamp2/MM0#d          1.18    6.07
         2.14    MXu_pad2/MM0#g              2.77    MXu_clamp1/MM0#d          1.08    5.98
         1.87    MXu_pad2/MM1#g              2.74    MXu_clamp2/MM0#d          1.18    5.80
         1.87    MXu_pad2/MM1#g              2.77    MXu_clamp1/MM0#d          1.08    5.72
         2.14    MXu_pad2/MM0#d              1.90    MXu_pad2/MM3#d            1.12    5.17
         1.87    MXu_pad2/MM1#d              1.90    MXu_pad2/MM3#d            1.12    4.90
         2.14    MXu_pad2/MM0#d              1.40    MXu_pad2/MM2#d            1.15    4.69
         1.87    MXu_pad2/MM1#d              1.40    MXu_pad2/MM2#d            1.15    4.42
         0.86    MXu_pad2/MM2#d              1.28    MXu_pad2/MM3#d            1.12    3.26
         0.86    MXu_pad2/MM2#d              0.00    MXu_pad2/MM2#d            1.15    2.01

---------------------------
Pin1     :    VDDIO
Pin2     :     VSSIO
Max total-reff    :    22.04
Status    :   fail
---------------------------
         18.12    MXu_pad2/MM0#g              2.74    MXu_clamp2/MM0#d          1.18    22.04
         18.12    MXu_pad2/MM0#g              2.77    MXu_clamp1/MM0#d          1.08    21.96
         17.92    MXu_pad2/MM1#g              2.74    MXu_clamp2/MM0#d          1.18    21.85
```

Current Density Check Reports

Voltus-Fi supports current density checks for ESD analysis in the GUI and in batch mode. The format and example of the report generated by Voltus-Fi-XL for this check is provided below.

Sample filename: `ESD_consolidated_report.rpt`

Format:

```
Trigger pin: DVDD
Reference pin: DVSS
Polarity: POSITIVE
```


© 2020 All Rights Reserved.

```
COMPLETE
FAIL
NET "DVDD"

%failed  resistor  layer  current   width  pathLength density limit needed
width/#via  X1   Y1   X2  Y2  resistance
```

Format Description:

`Trigger pin` specifies the pin where the HBM/CDM static current is applied.

`Reference pin` specifies the ground pin through which the current is discharged.

`Polarity` specifies the polarity (+ or -) of the current.

`%failed` specifies the number of resistors that failed the EM check.

`resistor` specifies the name of the resistor that failed the EM check.

`layer` specifies the layer on which the failed resistor is located.

`current` specifies the current flowing through the failed resistor.

`width` specifies the width of the failed resistor.

`pathLength` specifies the Blech length of the failed resistor.

`density limit needed` specifies the current density threshold value, which is required to pass the EM check.

`width/#via` specifies the ratio of the resistor width to the number of vias.

`X1, Y1, X2, Y2` specify the location of the resistor.

`resistance` specifies the resistance value in Ohms.

A part of the sample report is shown below.

© 2020 All Rights Reserved.

Figure 11-13 The ESD Current Density Sample Report

```
                                   ESD Current Density Report

%failed    resistor  layer  current    width  pathLength density   limit    needed width/#via  X1    Y1    X2    Y2    resistance
                                (A)     (um)   (um)     (A/um)    (A/um)     (um/#)              (um)  (um)  (um)  (um)  (ohm)
-----------------------------------------------------------------------------------------------------------------------------------

Trigger pin : DVDD Reference pin : DVSS Polarity : POSITIVE          COMPLETE          FAIL
        NET "DVDD"
                fail+1693.4  r1267071  via1   8.493650e-04 0     0       0.000000e+00 0.000000e+00 0            27.939 35.441 26.938 27.665 7
        NET "DVSS"
                fail+582.8   r1618311  via1   3.233707e-04 0     0       0.000000e+00 0.000000e+00 0            41.444 5.47   58.716 0.93   7

Trigger pin : DVDD Reference pin : DVSS Polarity : NEGATIVE          COMPLETE          FAIL
        NET "DVDD"
                fail+1669.0  r1267071  via1   8.378116e-04 0     0       0.000000e+00 0.000000e+00 0            27.939 35.441 26.938 27.665 7
        NET "DVSS"
                fail+2201.9  r1678218  via1   1.090162e-03 0     0       0.000000e+00 0.000000e+00 0            17.308 0.336  0.574  19.074 7

Trigger pin : VDD Reference pin : DVSS Polarity : POSITIVE          COMPLETE          FAIL
        NET "VSS"
                fail+14448.5 r781953   via1   6.890147e-03 0     0       0.000000e+00 0.000000e+00 0            52.03  67.078 50.401 66.199 7
        NET "VDD"
                fail+1254.4  r831677   via1   6.414336e-04 0     0       0.000000e+00 0.000000e+00 0            145.496 63.333 141.948 35.791 7
        NET "DVSS"
                fail+2165.2  r1678218  via1   1.072791e-03 0     0       0.000000e+00 0.000000e+00 0            17.308 0.336  0.574  19.074 7

Trigger pin : VDD Reference pin : DVSS Polarity : NEGATIVE          COMPLETE          FAIL
        NET "VDD"
                fail+1263.6  r831677   via1   6.458107e-04 0     0       0.000000e+00 0.000000e+00 0            145.496 63.333 141.948 35.791 7
        NET "DVSS"
                fail+10627.7 r1372986  via1   5.080639e-03 0     0       0.000000e+00 0.000000e+00 0            116.912 2.43   129.321 2.43   7
```

Rule-Based Check Reports

The following reports are generated for rule-based ESD checks:

- The Clamp-to-Clamp Rule Report
- The Pin-to-Clamp Rule Report
- Current Density Check Rule Report
- Summary Report

The Clamp-to-Clamp Rule Report

In this report, the effective resistance is reported between two clamps for every net.

Sample filename: `c2c_rule1.reff`

Format:

© 2020 All Rights Reserved.

```
#<clamp cell> <x y (um)> <pin> <layer> <clamp instance>
#- <pass/fail> <R ohm> <threshold ohm> <clamp cell> <x y (um)> <pin> <layer> <clamp
instance>
```

Format Description:

`clamp cell` specifies the from-clamp cell name.

`x y` specifies the location of the from-clamp cell.

`pin` specifies the name of the ESD pin of the from-clamp cell.

`layer` specifies the metal layer of the ESD pin of the from-clamp cell.

`clamp instance` specifies the name of the clamp instance.

`pass/fail` specifies the status of the to-clamp instance - a pass status is given when the clamp-to-clamp effective resistance is below the specified threshold value and a fail status is given when the effective resistance is above the specified threshold value.

`R ohm` specifies the clamp-to-clamp effective resistance.

`threshold ohm` specifies the minimum allowed effective resistance between the from-clamp and to-clamp cells.

`clamp cell` specifies the to-clamp cell name.

`x y` specifies the location of the to-clamp cell.

`pin` specifies the name of the ESD pin of the to-clamp cell.

`layer` specifies the metal layer of the ESD pin of the to-clamp cell.

`clamp instance` specifies the name of the to-clamp instance.

A part of a sample report is shown below.

© 2020 All Rights Reserved.

Figure 11-14 The Clamp-to-Clamp Rule Sample Report

```
##########################################################################################
# File Info    : Clamp2Clamp Effective Resistance Report
# Rule Name    : c2c_rule1
# Date created : 2019-Nov-14 02:31:18 (2019-Nov-14 10:31:18 GMT)
##########################################################################################


#NET AVDD

#<clamp cell> <x y (um)> <pin> <layer> <clamp instance>
#- <pass/fail> <R ohm> <threshold ohm> <clamp cell> <x y (um)> <pin> <layer> <clamp instance>
g45pd2svt (   20.775000 33.055000   ) AVDD NA DX828/X1/D0
- pass 0.0020000001 3.0000000 g45pd2svt (   53.500000 34.585000   ) AVDD NA DX829/X1/D0
- fail 19.345218 3.0000000 g45n2nvt (  -23.000000 48.855000   ) AVDD NA MX771/X0/M0
- fail 12.081248 3.0000000 g45n2nvt (  -22.740000 85.680000   ) AVDD NA MX772/X0/M0

g45pd2svt (   53.500000 34.585000   ) AVDD NA DX829/X1/D0
- pass 0.0020000001 3.0000000 g45pd2svt (   20.775000 33.055000   ) AVDD NA DX828/X1/D0
- fail 19.345218 3.0000000 g45n2nvt (  -23.000000 48.855000   ) AVDD NA MX771/X0/M0
- fail 12.081248 3.0000000 g45n2nvt (  -22.740000 85.680000   ) AVDD NA MX772/X0/M0

g45n2nvt (  -23.000000 48.855000   ) AVDD NA MX771/X0/M0
- fail 19.345218 3.0000000 g45pd2svt (   20.775000 33.055000   ) AVDD NA DX828/X1/D0
- fail 19.345218 3.0000000 g45pd2svt (   53.500000 34.585000   ) AVDD NA DX829/X1/D0
- fail 31.240666 3.0000000 g45n2nvt (  -22.740000 85.680000   ) AVDD NA MX772/X0/M0

g45n2nvt (  -22.740000 85.680000   ) AVDD NA MX772/X0/M0
- fail 12.081248 3.0000000 g45pd2svt (   20.775000 33.055000   ) AVDD NA DX828/X1/D0
- fail 12.081248 3.0000000 g45pd2svt (   53.500000 34.585000   ) AVDD NA DX829/X1/D0
- fail 31.240666 3.0000000 g45n2nvt (  -23.000000 48.855000   ) AVDD NA MX771/X0/M0
```

The Pin-to-Clamp Rule Report

In this report, the effective resistance is reported for each pin and all ESD devices.

Sample filename: `p2c_rule1.reff`

Format:

```
#Reporting from pin to clamp
#<pin> <x y (um)> <layer>
#- <pass/fail> <R ohm> <threshold ohm> <clamp cell> <x y (um)> <pin> <layer> <clamp instance>


#Reporting from clamp to pin
#<clamp cell> <x y (um)> <pin> <layer> <clamp instance>
#- <pass/fail> <R ohm> <threshold ohm> <x y (um)> <layer> <bump>
```

Format Description:

© 2020 All Rights Reserved.

If the effective resistance is computed from pin to clamp:

`pin` specifies the pin name.

`x y` specifies the location of the pin.

`layer` specifies the metal layer on which the pin is located.

`pass/fail` specifies the status of the clamp instance - a pass status is given when the pin-to-clamp effective resistance is below the specified threshold value and a fail status is given when the effective resistance is above the specified threshold value.

`R ohm` specifies the pin-to-clamp effective resistance.

`threshold ohm` specifies the minimum allowed effective resistance between a power or ground pin and an ESD device pair.

`clamp cell` specifies the name of the ESD clamp cell.

`x y` specifies the location of the clamp cell pin node. If there are multiple nodes on the pin and are shorted, this is the location of the shorted pin.

`pin` specifies the name of the ESD pin.

`layer` specifies the metal layer of the pin whose x,y location is given.

`clamp instance` specifies the name of the ESD clamp instance.

If the effective resistance is computed from clamp to pin:

`clamp cell` specifies the from-clamp cell name.

`x y` specifies the location of the from-clamp cell.

`pin` specifies the name of the ESD pin of the from-clamp cell.

`layer` specifies the metal layer of the ESD pin of the from-clamp cell.

`clamp instance` specifies the name of the clamp instance.

`pass/fail` specifies the status of the pin instance - a pass status is given when the clamp-to-pin effective resistance is below the specified threshold value and a fail status is given when the effective resistance is above the specified threshold value.

`R ohm` specifies the clamp-to-pin effective resistance.

© 2020 All Rights Reserved.

`threshold ohm` specifies the minimum allowed effective resistance between the from-clamp and to-pin cells.

`x y` specifies the location of the to-pin cell.

`layer` specifies the metal layer of the ESD pin of the to-pin cell.

`bump` specifies the bump name.

A part of a sample report is shown below.

Figure 11-15 The Pin-to-Clamp Rule Sample Report

```
##############################################################################################
# File Info    : Pin2Clamp Effective Resistance Report
# Rule Name    : p2c_rule1
# Date created : 2019-Nov-14 02:31:18 (2019-Nov-14 10:31:18 GMT)
##############################################################################################


#NET AVDD

#Reporting from pin to clamp

#<pin> <x y (um)> <layer>
#- <pass/fail> <R ohm> <threshold ohm> <clamp cell> <x y (um)> <pin> <layer> <clamp instance>
AVDD  (  -30.110000 79.422500   ) dwires
- fail 19.344218 2.5000000 g45n2nvt (  -23.000000 48.855000   ) AVDD NA MX771/X0/M0
- fail 12.080248 2.5000000 g45n2nvt (  -22.740000 85.680000   ) AVDD NA MX772/X0/M0

#Reporting from clamp to pin

#<clamp cell> <x y (um)> <pin> <layer> <clamp instance>
#- <pass/fail> <R ohm> <threshold ohm> <x y (um)> <layer> <bump>
g45n2nvt ( -23.000000 48.855000 ) AVDD NA MX771/X0/M0
- fail 19.344218 2.5000000 (  -30.110000 79.422500   ) dwires AVDD
g45n2nvt ( -22.740000 85.680000 ) AVDD NA MX772/X0/M0
- fail 12.080248 2.5000000 (  -30.110000 79.422500   ) dwires AVDD
```

Current Density Check Rule Report

Sample filename: `VDD.rj.avg.rpt`

Format:

```
I/Ilimit    I(mA) Ilimit(mA)  Layer  Location   Length(um)  Width(um)
ViaArea(um^2) Via Cuts    Fix Width/Via  Benefit Factor  J(A/cm^2)   Jmax(A/cm^2)
Resistor Id
```

Format Description:

© 2020 All Rights Reserved.

`I(mA)` specifies the current value that flows through the metal or via resistor and which is calculated by the tool.

`Ilimit(mA)` specifies the current limit value that is provided by the foundry depending upon its width and thickness.

`Layer` specifies the metal or via layer name on which the resistor is located.

`Location` specifies the coordinates of the resistor.

`Length` specifies the length of the resistor.

`Width` specifies the width of the resistor.

`Via Area` is the drawn dimension via area in square microns.

`Via Cuts` is the number of via cuts.

`Fix Width/ Via` The `Fix Width/Via` is recalculated by modifying the existing via width to eliminate EM violation. If `I/Ilimit` is greater than 1, which is a violation, `Fix Width/Via` will be `violation * via width`

`Benefit Factor` When certain constraints are met, its EM limit can be 2X, 3X, or multiple X of the base EM limit. These multiples are called Benefit Factor.

`J(A/cm^2)` is the current density value.

`Jmax(A/cm^2)` is the current density limit.

`Resistor Id` is the unique ID assigned to a resistor.

A part of a sample report is shown below.

© 2020 All Rights Reserved.

Figure 11-16 Current Density Check Rule Sample Report

```
# Design: DiffOpAmp_esd
# Net name: AVDD
# EM analysis: Avg
# EM Reporting Threshold: 0.9
# Total Violations: 567
#
# Report format:#
# I/Ilimit    I(mA)       Ilimit(mA)  Layer       Location                      Length(um)  Width(um)   ViaArea(um^2) Via Cuts    Fix Width/Via Benefit Factor J(A/cm^2)   Jmax(A/cm^2) Resistor Id
-------------------
```

`Status` specifies the PASSED, FAILED, or ERROR status of the rule. It will have the status PASSED if all the nets under this rule pass, and will have the status FAILED if even one net fails. The ERROR status is displayed if an error occurs during the run of the rules.

`Net` specifies the nets analyzed under this rule.

`Type` specifies the type of the net being analyzed.

`FromClamps` specifies the number of clamps analyzed under the from section of the rule on each net.

`ToClamps` specifies the number of clamps analyzed under the to section of the rule on each net.

`Total Paths` specifies the number of paths analyzed. If from and to clamps are of the same clamp type, total paths = `#FromClamps * (#Toclamps - 1)`. If from and to are not of same clamp type, total paths = `#Fromclamps * #Toclamps`.

`Status` specifies the status of each net.

`MinRes` specifies the minimum resistance value for each net.

`MaxRes` specifies the maximum resistance value for each net.

- Pin-to-Clamp Rule

```
Rule p2c_rule1
type pin2clamp
clamp_types g45n2nvt
power -nets AVDD
ground -nets AVSS
signal -nets all
reff_threshold 2.5
reporting both
End Rule

Status FAILED

Net                 Type        Clamps      Pins        Short       TotalPaths  Status      MinRes      MaxRes
-------------------------------------------------------------------------------------------------------------------

AVDD                power       2           1           FALSE       2           FAIL        12.0802     19.3442
AVSS                ground      2           1           FALSE       2           FAIL        9.42209     22.4207
inm                 signal      0           1           FALSE       0           NA          NA          NA
inp                 signal      0           1           FALSE       0           NA          NA          NA
```

`Status` specifies the PASSED, FAILED, or ERROR status of the rule. It will have the status PASSED if all the nets under this rule pass, and will have the status FAILED if even one net fails. The ERROR status is displayed if an error occurs during the run of the rules.

`Net` specifies the nets analyzed under this rule.

`Type` specifies the type of the net being analyzed.

`Clamps` specifies clamp cells on each net.

© 2020 All Rights Reserved.

`Pins` specifies pins on each net.

`Short` specifies if pins were shorted.

`Total Paths` specifies the number of paths analyzed. If pins are shorted, total paths is equal to the number of clamps. If pins are not shorted, total paths is equal to `#Pins * #Clamps`.

`Status` specifies the status of each net.

`MinRes` specifies the minimum resistance value for each net.

`MaxRes` specifies the maximum resistance value for each net.

- Current Density Check Rule

```
Rule em_rule1
type em
source -type power -nets AVDD inm -short_pins true
sink -type ground -nets AVSS inp -short_pins true
clamp_types g45n2nvt g45pd2svt
current 13
End Rule
Status FAILED

Domain                        Net                Type        Threshold   Violations  Worst(I/Ilimit) I(mA)       Ilimit(mA)  Layer
Status
-------------------------------------------------------------------------------------------------------------------------------------------

AVDD_short_AVSS_short_AVDD    AVDD               power       0.9         567         1323.41         2.64947     1323.41     metal1_conn
FAILED
AVDD_short_AVSS_short_AVSS    AVSS               ground      0.9         47          8.30468         0.172737    8.30468     via1
FAILED
AVDD_short_inp_short_AVDD     AVDD               power       0.9         1155        1441.34         2.88556     1441.34     metal1_conn
FAILED
AVDD_short_inp_short_inp      inp                ground      0.9         19          840.651         3.93698     840.651     metal1_conn
FAILED
inm_short_AVSS_short_AVSS     AVSS               ground      0.9         62          2501.5          6.5039      2501.5      via1
FAILED
inm_short_AVSS_short_inm      inm                power       0.9         3           1387.09         6.4961      1387.09     metal1_conn
FAILED
inm_short_inp_short_inm       inm                power       0.9         31          1308.59         6.12846     1308.59     metal1_conn
FAILED
inm_short_inp_short_inp       inp                ground      0.9         31          149.398         0.69967     149.398     metal1_conn
FAILED
```

`Status` specifies the PASSED, FAILED, or ERROR status of the rule. It will have the status PASSED if all the nets under this rule pass, and will have the status FAILED if even one net fails. The ERROR status is displayed if an error occurs during the run of the rules.

`Domain Status` specifies the status of the domain.

`Net` specifies the nets analyzed under this rule.

`Type` specifies the type of the net being analyzed.

`Threshold` specifies the effective resistance values that are greater than the specified threshold value.

`Violations` specifies the number of violations on each domain.

© 2020 All Rights Reserved.

`Worst(I/Ilimit)` specifies the current value or the current limit value, whichever is greater.

`I(mA)` specifies the current value that flows through the metal or via resistor and which is calculated by the tool.

`Ilimit(mA)` specifies the current limit value that is provided by the foundry depending upon its width and thickness.

`Layer` specifies the metal or via layer name on which the resistor is located.

© 2020 All Rights Reserved.

Sample ICT File with EM Limits

Following is a sample ICT file with EM limits:

```
process "test_esd" {
}
conductor "M1" {
em_model {
        em_jmax_dc_peak EQU 65*w
        }
}
conductor "M2" {
em_model {
        em_jmax_dc_peak EQU 65*w
        }
}
conductor "M3" {
em_model {
        em_jmax_dc_peak EQU 65*w
        }
}
conductor "M4" {
em_model {
        em_jmax_dc_peak EQU 65*w
        }
}
conductor "M5" {
em_model {
        em_jmax_dc_peak EQU 77*w
        }
}
}
```


© 2020 All Rights Reserved.

ESD Form Description

ESD Form - ESD Setup Tab

Field	Description
ESD Common Setup	
Power Nets	Power nets for which ESD analysis is to be run. Multiple power nets can be specified separated by a space. Wildcards are supported.
Ground Nets	Ground nets for which ESD analysis is to be run. Multiple ground nets can be specified separated by a space. Wildcards are supported.
Signal Nets	Signal nets for which ESD analysis is to be run. Multiple signal nets can be specified separated by a space. Wildcards are supported.
DSPF File	DSPF file generated using Quantus. The file contains information about model names, including the ESD model names required for ESD analysis. A DSPF file in gzip format is also supported.
ESD transistors/diodes	ESD protection clamp cells that are used in the DSPF netlist. Multiple ESD cells can be specified separated by a space. Wildcards are not supported. Typically, the ESD cells are of two types, diode and ESD MOS device.
DFII layer map file	File containing mapping information between the layer names in the DSPF file and the technology file.
Model File	File containing references to all the models required to simulate the DSPF circuit. The file should contain information about models for all devices, both ESD and non-ESD. However, if it does not contain information about the models for ESD devices, Voltus-Fi generates TLP models for the ESD macro devices.
qrcTechFile / ICT File	Quantus technology file or the ICT file that contains the EM limits for ESD current density calculations.

© 2020 All Rights Reserved.

Field	Description
Optional Inputs	
CDL File	Schematic CDL netlist. If the netlist is specified, Voltus-Fi considers the extracted DSPF file to be ESD-cells flat. In this case, Voltus-Fi locates all the diodes/devices present in the ESD cells inside the CDL netlist and looks for the same diodes/devices in the DSPF file. **Note:** Ensure that the DSPF file contains the schematic namespace and not the layout namespace.
CDL Cells	ESD CDL cells. Multiple cells can be specified separated by a space.
Common EMIR conf File	File to be used for all zap simulations.
EM Only ICT File	File containing process and EM model information for EM analysis of ESD zap.
Use Optimized DSPF	Indicates that an optimized xDSPF file will be used for faster ESD analysis.
Load DSPF	Loads the DSPF file.
Advanced Settings	Opens the Advanced Settings form. This form is used to specify the number of threads to use for multithreading, or the maximum number of computers to use for distributed processing, or the maximum number of computers and the number of threads to use for superthreading. It is also used for specifying the Spectre simulation settings.
ESD Instances	
Instances	ESD instances of all ESD macro cells and all fingers of the listed ESD instances. The ESD instances are displayed in a tree view under the two cell type nodes: ESD Cell Devices and CDL Cell Devices. You can expand a cell type node to view all the instances of the selected cell type and expand an instance to view all the fingers of that instance.

© 2020 All Rights Reserved.

Field	Description
Select/Deselect	Selects or clears an instance or a finger of an instance.
Select All Instances	Selects all instances listed in the table.
Unselect All Instances	Clears all selected instances in the table.

Advanced Settings Form - Run Settings Tab

Field	Description
Run Settings	
Simulation Run Mode	Mode to distribute the simulation run. The following options are available: ■ *Multi CPU* (default): Distributes the run to multiple CPUs (multi-threading) on the local machine. ■ *LSF*: Distributes the run to different machines. The LSF mode supports multi-threading, distributed processing, and superthreading. On selecting the LSF option, additional fields in the form are enabled in which you can add the LSF resources.
Multi CPU Configuration	
Number of Local Processors	Number of CPUs to be used for a zap simulation.
Distributed Configuration	
Remote Hosts	Number of clients to be used for a zap simulation. This field is required for both distributed processing and superthreading.
CPU's per Remote Host	Number of CPUs used by clients for multithreading. **Note:** In case of superthreading, this field must be used in conjunction with the *Remote Hosts* field.
LSF Configuration	

© 2020 All Rights Reserved.

Field	Description
Args	Any additional arguments that need to be provided to the LSF job submission command.
Queue	Queue for the LSF configuration.
Resource	Resource string for the LSF queue.
Timeout	Timeout interval for a submitted job that will never start. It prevents the master machine from hanging when the clients do not come up for some reason.
Effective Command	Shows the effective command string compiled from the fields in the *LSF Configuration* section. This value is used as the submission string to LSF. You can also modify this field for custom distributed mode settings.
Generate	Generates the effective command string.
OK	Completes the command and closes the form.
Cancel	Closes the form without executing the command.
Apply	Completes the command and keeps the command active and the form on the screen.
Help	Opens help information in Cadence Help.

Advanced Settings Form - Simulation Settings Tab

Field	Description
Spectre Simulation Settings	
Spectre Simulator Options	Spectre setup simulation file options. For example, `highvoltage=yes`.
Spectre Command Line Options	Options to be used while invoking Spectre. For example, `+aps +postlayout=hpa`. For static DC simulation, you can also add `+fastdc`.
Spectre Transient Options	Options to be used while running a `tran` simulation. For example, `method=trap`.
Spectre Threads	Number of threads to be used while running a simulation.

© 2020 All Rights Reserved.

Field	Description
Default	Restores the default values in the fields, and any previously specified settings for a particular field are cleared.
OK	Completes the command and closes the form.
Cancel	Closes the form without executing the command.
Apply	Completes the command and keeps the command active and the form on the screen.
Help	Opens help information in Cadence Help.

ESD Form - ESD Optimizer Tab

Field	Description
SPEF Checker Log	
Original DSPF SPEF Checker Log	SPF checker log of the original xDSPF file.
Optimized DSPF SPEF Checker Log	SPF checker log of the optimized xDSPF file.
Run Optimizer	Starts the optimizer run.
View Log	Displays the log file for the run.
Stop Optimizer	Stops the optimizer run.

ESD Form - ESD Model Tab

Field	Description
Static ESD Device Models	TLP modeling to be used for static ESD analysis.
ESD TLP Parameters (Vt, Ron, Roff) Data	
Cell Name	Name of the ESD cell.

© 2020 All Rights Reserved.

Field	Description
Model Type	ESD model settings for static ESD analysis. The following options are available: ■ *Spectre/Spice Models*: In this mode, Voltus-Fi does not generate any ESD models. It uses the ESD model settings specified in the *Model File* box on the *ESD Setup* tab. Note that the models for ESD devices must be present in the models library specified for simulation. These may be general Spice models. When you select this mode, other TLP parameter columns in the table are disabled. ■ *Vt, Ron, Roff Models*: In this mode, Voltus-Fi generates ESD models for the devices based on the following specified parameters: `Ron+ Ron-` (on-resistance voltage), `Roff+ Roff-` (off-resistance voltage), and `Vt1+ Vt1-` (trigger voltage).
Cell Type	Type of ESD cells. Five types of ESD cells are supported: *MOS*, *NDIODE*, *PDIODE*, *DESIGN_CAP*, and *DESIGN_RES*.
TERM+ IN	Inward terminal for the forward direction of the flow of current.
TERM+ OUT	Outward terminal for the forward direction of the flow of current.
TERM- IN	Inward terminal for the reverse direction of the flow of current.
TERM- OUT	Outward terminal for the reverse direction of the flow of current.
Ron +	Device on resistance for the forward direction of the flow of current.
Roff +	Device off resistance for the forward direction of the flow of current.
Vt1 +	Trigger voltage for the forward direction of the flow of current.

© 2020 All Rights Reserved.

Field	Description
Ron -	Device on resistance for the reverse direction of the flow of current.
Roff -	Device off resistance for the reverse direction of the flow of current.
Vt1 -	Trigger voltage for the reverse direction of the flow of current.
Design Res	The ESD device is modeled as design resistor based on the resistance value specified (in ohm) in this column. Default value is `1 ohm`.
Design Cap	The ESD device is modeled as design capacitor based on the capacitance value specified (in farad) in this column. Default value is `1 farad`.
Dynamic ESD Device Models	TLP modeling to be used for dynamic ESD analysis.
ESD TLP-IV Data	
TLP IV Model File	File containing IV data of the ESD devices.
Load Cell IV Data from File	Loads IV data of the ESD devices from the TLP IV model file.
Cell Name	Name of the ESD cell.
Model Type	ESD model settings for static ESD analysis. The following options are available: ■ *Spectre/Spice Models*: In this mode, Voltus-Fi does not generate any ESD models. It uses the ESD model settings specified in the *Model File* box on the *ESD Setup* tab. Note that the models for ESD devices must be present in the models library specified for simulation. These may be general Spice models. When you select this mode, other TLP-IV parameter columns in the table are disabled. ■ *TLP I-V Models*: In this mode, Voltus-Fi generates ESD models for the devices based on the specific snapback current voltage or IV characteristics described in the model data file.

© 2020 All Rights Reserved.

Field	Description
Cell Type	Type of ESD cells. Five types of ESD cells are supported: *MOS*, *NDIODE*, *PDIODE*, *DESIGN_CAP*, and *DESIGN_RES*.
IV Data	Comma separated I,V values that are used for modeling of snapback behavior of MOS and diode devices.
Design Res	The ESD device is modeled as design resistor based on the resistance value specified (in ohm) in this column. Default value is `1 ohm`.
Design Cap	The ESD device is modeled as design capacitor based on the capacitance value specified (in farad) in this column. Default value is `1 farad`.

ESD Form - ESD Run Tab

Field	Description
Resistance Checks	
Pin-to-Clamp	Effective resistance is computed between each pin and all ESD devices within the specified threshold value.
Clamp-to-Clamp	Effective resistance is computed between all clamps for each net.
Pin-to-Pin	Effective resistance is computed for all paths—from a power pin to ESD device to ground pin—within the specified threshold value.
Pin-to-pin Threshold	Threshold value for pin-to-pin resistance checks.
Pin-to-clamp Threshold	Threshold value for pin-to-clamp resistance checks. **Note:** The Pin-to-clamp resistance check threshold value is also applicable for the clamp-to-clamp resistance check. Default value: `10 Ohms`.

© 2020 All Rights Reserved.

Field	Description
Finger Merge	Merges the MOS fingers in the resistance check reports.
Split Pins Shorting	Short the split pins.
Run R-effective Checks	Runs the effective resistance checks.
View Log	Displays the log file for the effective resistance run.
Stop R-effective Run	Stop the effective resistance checks run.
Clear R-Checks	Clears the results of the previous effective resistance checks run.
DC Mode	
DC Current (A)	Indicates that DC simulation is run by injecting current. Default value: `6 Ampere (A)`.
DC Voltage (V)	Indicates that DC simulation is run by creating voltage source. Default value: `200 Volts (V)`.
Select/Deselect All DC Simulation Jobs	Specifies whether to simulate ESD events for all or selected power/ground/signal-PAD zap combinations. By default, all combinations are selected.
Start Simulation	Runs the simulation for DC mode.
Stop Simulation	Stops the simulation for all the zaps running in the background.
Check HeartBeat	Displays the status of the simulation being performed.
ESD CD Report	Displays the report of ESD currents of all the ESD instances for the specific zap. This report shows the peak current through each terminal of the ESD instances on the simulated net.
Tool Gen Zaps	These zaps are generated by default. They are created from the power nets, ground nets, and signal nets specified in the *ESD Setup* tab.

© 2020 All Rights Reserved.

Field	Description
Custom Zaps	Lets you add custom zaps by specifying their current/voltage polarity, the injection pin, and reference pin information.
CDM Mode	
CDM Current (A)	Indicates that CDM simulation is run by injecting current. Default value: `6 Ampere (A)`.
CDM Voltage (V)	Indicates that CDM simulation is run by creating voltage source. Default value: `200 Volts (V)`.
Transient Time (ns)	Time taken for the circuit to change from one steady state to another steady state.
CDM Testbench Settings	Lets you specify either an in-house testbench or a tool-generated testbench for ESD analysis.
Select/Deselect All CDM Simulation Jobs	Specifies whether to simulate ESD events for all or selected power/ground/signal-PAD zap combinations. By default, all combinations are selected.
Start Simulation	Runs the simulation for CDM mode.
Stop Simulation	Stops the simulation for all the zaps running in the background.
Check HeartBeat	Displays the status of the simulation being performed.
ESD CD Report	Displays the report of ESD currents of all the ESD instances for the specific zap. This report shows the peak current through each terminal of the ESD instances on the simulated net.
Tool Gen Zaps	These zaps are generated by default. They are created from the power nets, ground nets, and signal nets specified in the ESD Setup tab.

© 2020 All Rights Reserved.

Field	Description
Custom Zaps	Multiple pins can be specified separated by a space. Wildcards are not supported. When multiple pins are specified as either injection or reference, these pins are shorted internally during simulation
HBM Mode	
HBM Current (A)	Indicates that HBM simulation is run by injecting current. Default value: `1.6 Ampere (A)`.
HBM Voltage (V)	Indicates that DC simulation is run by creating voltage source. Default value: `2000 Volts (V)`.
Transient Time (ns)	Time taken for the circuit to change from one steady state to another steady state.
HBM Testbench Settings	Lets you specify either an in-house testbench or a tool-generated testbench for ESD analysis.
Select/Deselect All HBM Simulation Jobs	Specifies whether to simulate ESD events for all or selected power/ground/signal-PAD zap combinations. By default, all combinations are selected.
Start Simulation	Runs the simulation for HBM mode.
Stop Simulation	Stops the simulation for all the zaps running in the background.
Check HeartBeat	Displays the status of the simulation being performed.
ESD CD Report	Displays the report of ESD currents of all the ESD instances for the specific zap. This report shows the peak current through each terminal of the ESD instances on the simulated net.
Tool Gen Zaps	These zaps are generated by default. They are created from the power nets, ground nets, and signal nets specified in the ESD Setup tab.

© 2020 All Rights Reserved.

Field	Description
Custom Zaps	Multiple pins can be specified separated by a space. Wildcards are not supported. When multiple pins are specified as either injection or reference, these pins are shorted internally during simulation
PERC	
Launch PERC	Launches the Cadence® Pegasus™ or PVS PERC (Programmable Electrical Rule Check) interface.

ESD Form - ESD Rule Check Tab

Field	Description
Rule File	File containing the rules.
Load	Loads rules from a rule file.
Finger Merge	Merges the MOS fingers in the resistance check reports.
Rule Database	Displays the list of rules.
Rule Detail	Displays details of the rule selected in the *Rule Database* group box. The rule details are displayed as specified in the rule file.
Run Rules	Starts the rule-based ESD checks run.
View Log	Displays the log file for the run.
Stop Rules	Stops the rule-based ESD checks run.

© 2020 All Rights Reserved.

Field	Description
Summary Report	Report of the selected rule. The summary report is displayed only after a rule-based ESD checks run is complete or if the results of the previous run are present in the output directory. The summary report displays a net-wise summary for `clamp2clamp` and `pin2clamp` rule types and a domain-wise summary for `em` type. Double-click any row or column to see the detailed report for `clamp2clamp` and `pin2clamp` rule types. Similarly, double-click any row or column to see the detailed average current report of the selected domain and net for `em` rule type.
Rule Type	Type of rule.
Rule Status	Status of the rule: PASSED, FAILED, or ERROR.

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

12

Structural Analysis

- Structural Analysis in Voltus-Fi-XL
- Checks Performed for Structural Analysis in Voltus-Fi-XL
- Top Via Missing Check in Voltus-Fi-XL
- Bottom Via Missing Check in Voltus-Fi-XL
- Missing Via Connections Check in Voltus-Fi-XL
- Via Coverage Ratio Check in Voltus-Fi-XL
- Skewed Via Ratio Check in Voltus-Fi-XL
- Running Structural Analysis in Voltus-Fi-XL
- Results of Structural Analysis in Voltus-Fi-XL
- IR/EM Results Form — Structural Analysis Tab

© 2020 All Rights Reserved.

Structural Analysis in Voltus-Fi-XL

A typical electrical IR/EM analysis, which involves extraction of the power network, current calculation, and matrix solve, is able to identify devices that have a higher IR drop than the user-defined threshold.

Debugging the source of IR drop problems is a manual process involving the study of various voltage or current plots and patterns, and then narrowing down the weak links in the power grid. This is an effort-intensive activity.

Voltus-Fi-XL supports structural analysis, which is a shape-based geometrical analysis that is used to quickly identify power-grid weaknesses in designs and mark them on the Virtuoso layout for you to view and debug. Structural analysis cuts down on the sign-off analysis time. This analysis is available in the GUI mode.

You can perform structural analysis using the results of the IR drop analysis.

Related Topics

Checks Performed for Structural Analysis in Voltus-Fi-XL, Running Structural Analysis in Voltus-Fi-XL, Results of Structural Analysis in Voltus-Fi-XL, IR/EM Results Form — Structural Analysis Tab

Checks Performed for Structural Analysis in Voltus-Fi-XL

The checks performed in structural analysis in Voltus-Fi-XL work on the basic definition of connectivity—for every metal segment, there should be connectivity from the top metal layers to the bottom metal layers unless the metal segment belongs to either the top-most metal layer or to the bottom-most metal layer.

The via connectivity checks are performed on full-metal segments and not on the individual shapes that are part of the metal segments.

The following categories of checks can be performed for routing layers:

- For unconnected shapes in power net layout
  - Top Via Missing Check in Voltus-Fi-XL
  - Bottom Via Missing Check in Voltus-Fi-XL
- For overlapping layers
  - Missing Via Connections Check in Voltus-Fi-XL

© 2020 All Rights Reserved.

- Via Coverage Ratio Check in Voltus-Fi-XL

- For weakly connected shapes in power net layout
  - Skewed Via Ratio Check in Voltus-Fi-XL

Related Topics

Structural Analysis in Voltus-Fi-XL, Running Structural Analysis in Voltus-Fi-XL, Results of Structural Analysis in Voltus-Fi-XL, IR/EM Results Form — Structural Analysis Tab

Top Via Missing Check in Voltus-Fi-XL

Top Via Missing is a check in Voltus-Fi-XL to perform structural analysis. This check highlights all metal segments that have no top via or contact shapes overlapping them on layer basis. In the following example, metal layer, M3, has via, V2, but does not have via, V3.

The highlighted shape in the example below has V2 but not V3.

© 2020 All Rights Reserved.

Related Topics

Checks Performed for Structural Analysis in Voltus-Fi-XL, Bottom Via Missing Check in Voltus-Fi-XL, Missing Via Connections Check in Voltus-Fi-XL, Via Coverage Ratio Check in Voltus-Fi-XL, Skewed Via Ratio Check in Voltus-Fi-XL

© 2020 All Rights Reserved.

Bottom Via Missing Check in Voltus-Fi-XL

Bottom Via Missing is a check in Voltus-Fi-XL to perform structural analysis. This check highlights all metal segments that have no bottom via or contact shapes overlapping them on layer basis. In the following example, metal layer, M3, has via, V3, but does not have via, V2.

The highlighted shapes in the example below have V3 but not V2.

Related Topics

Checks Performed for Structural Analysis in Voltus-Fi-XL, Top Via Missing Check in Voltus-Fi-XL, Missing Via Connections Check in Voltus-Fi-XL, Via Coverage Ratio Check in Voltus-Fi-XL, Skewed Via Ratio Check in Voltus-Fi-XL

Missing Via Connections Check in Voltus-Fi-XL

Missing Via Connections is a check in Voltus-Fi-XL to perform structural analysis. This check highlights overlapping wire segments that have no vias in the overlapping area. In the following example, the overlapping area between M4 and M3 has V3, but the overlapping area between M3 and M2 does not have V2.

© 2020 All Rights Reserved.

The highlighted overlapping shape of the metal segments in the example below has missing via.

© 2020
All Rights Reserved.

Related Topics

Checks Performed for Structural Analysis in Voltus-Fi-XL, Top Via Missing Check in Voltus-Fi-XL, Bottom Via Missing Check in Voltus-Fi-XL, Via Coverage Ratio Check in Voltus-Fi-XL, Skewed Via Ratio Check in Voltus-Fi-XL

Via Coverage Ratio Check in Voltus-Fi-XL

Via Coverage Ratio Check is a check in Voltus-Fi-XL to perform structural analysis. This check highlights segments that have less than user-specified via coverage ratio in the overlapping area. In the following example, the overlapping area between M2 and M3 has a via, but the via coverage ratio or the ratio of via area and overlapping area is less than the user-specified ratio.

The highlighted shape in the example below has a via coverage ratio less than that specified by the user.

© 2020 All Rights Reserved.

Related Topics

Checks Performed for Structural Analysis in Voltus-Fi-XL, Top Via Missing Check in Voltus-Fi-XL, Bottom Via Missing Check in Voltus-Fi-XL, Missing Via Connections Check in Voltus-Fi-XL, Skewed Via Ratio Check in Voltus-Fi-XL

Skewed Via Ratio Check in Voltus-Fi-XL

Skewed Via Ratio Check is a check in Voltus-Fi-XL to perform structural analysis. This check highlights segments in which the ratio of top and bottom connecting vias is skewed. In the following example, the ratio of top/bottom via is 2:8, which is skewed.

© 2020 All Rights Reserved.

The highlighted shape in the example below has a skewed top/bottom via ratio.

For this check, you can choose to:

- Highlight wire segments that have a top/bottom via ratio higher than the specified ratio.
- Highlight wire segments that have a bottom/top via ratio higher than the specified ratio.

Related Topics

Checks Performed for Structural Analysis in Voltus-Fi-XL, Top Via Missing Check in Voltus-Fi-XL, Bottom Via Missing Check in Voltus-Fi-XL, Missing Via Connections Check in Voltus-Fi-XL, Via Coverage Ratio Check in Voltus-Fi-XL

© 2020 All Rights Reserved.

Running Structural Analysis in Voltus-Fi-XL

Voltus-Fi-XL performs structural analysis to quickly identify power-grid weaknesses in designs and mark them on the Virtuoso layout for you to view and debug.

Prerequisites

To perform structural analysis, you first need to load the IR drop analysis results. Before you run structural analysis, ensure the following:

- In the *Layers* group box in the expanded IR/EM Results form, the layers for which you want to view the results of the structural analysis are selected.
- The list of selected layers includes metal layers because structural analysis is performed only for metal layers.

To run structural analysis, perform the following steps.

1. In the IR/EM Results form, select the *Structural Analysis* tab.

© 2020 All Rights Reserved.

The following information is displayed.

2. Select the *Layer Map File* and click *Run*.

© 2020
All Rights Reserved.

The names of layers for which data is available will appear under the field titled, *Analysis Data Valid For Layers*.

3. Click *Select Net* to select a net on the Virtuoso layout.
4. To view the data, the options provided for different check types are enabled. Select the specific check you want to perform for the selected layers.

   You can view the violations for the selected check type in the result browser. The following information is available for the selected metal layers for all checks except the missing via check: *Layer*, *Area* (u^2), *IR(V)*, *#TopVias*, *#BotVias*, and *Ratio*. Double-click the column headings of the result browser to list the results in either an ascending or descending order.
5. Click *Auto Zoom* to zoom into the violations for the specified check.
6. To change the layers for which you want to perform structural analysis, select or deselect the layers in the *Layers* group box in the expanded IR/EM Results form and click *update*.

Related Topics

Structural Analysis in Voltus-Fi-XL, Checks Performed for Structural Analysis in Voltus-Fi-XL, Performing Structural Analysis using Voltus-Fi (video), IR/EM Results Form — Structural Analysis Tab

Results of Structural Analysis in Voltus-Fi-XL

After you run structural analysis in Voltus-Fi-XL, the results are displayed in the Virtuoso layout depending on the specified check type. Provided below are examples for the following checks:

Show Unconnected from Top

The figure below shows an example of structural analysis results in the Virtuoso layout when the *Show Unconnected from Top* check is specified. Here, the metal segment (`m2`) does not have any top vias.

© 2020 All Rights Reserved.

Show Unconnected from Bottom

The figure below shows an example of structural analysis results in the Virtuoso layout when the *Show Unconnected from Bottom* check is specified. Here, the metal segment (`Metal_1`) has only top vias. It does not have any bottom vias.

© 2020 All Rights Reserved.

Bottom / Top Via Ratio

The figure below shows an example of structural analysis results in the Virtuoso layout when the *Bottom / Top via Ratio >* check is specified. In this example, the user-specified bottom/top via ratio is 8. The metal segment (`Metal_1`) has 3 top vias and 32 bottom vias. The bottom to top via ratio is 32:3, which is greater than the user-specified ratio of 8.

© 2020 All Rights Reserved.

Missing Via Connections

In this example, the overlapping area of metal segments `Metal_4` and `Metal_3`, which is highlighted in red, has a missing via, `v3`. The result browser shows the overlapping layer names and the overlapping area of the two segments. This is shown in the figure below.

© 2020 All Rights Reserved.

Related Topics

Structural Analysis in Voltus-Fi-XL, Checks Performed for Structural Analysis in Voltus-Fi-XL, Running Structural Analysis in Voltus-Fi-XL, IR/EM Results Form — Structural Analysis Tab

© 2020 All Rights Reserved.

IR/EM Results Form — Structural Analysis Tab

Field	Description
Net Header File	Specifies the net header file.

Layer Map File	Specifies the APS/XPS-to-DFII layer map file. You can create a new layer map file or edit an existing file, using the edit button provided next to the field. An example of the layer map file is provided below.
	`#<type> <extraction layer name> <dfII layer name>`
	`metal  metal7  M7`
	`via  VIA5  VIA6`
	`metal  metal6  M6`
	`via  VIA5  VIA5`
	`metal  metal5  M5`
	`via  VIA4  VIA4`
	`metal  metal4  M4`
	`via  VIA3  VIA3`
	`metal  metal3  M3`
	The order of layers in the layer map file is important. The layer map file must have layers listed in a top-to-bottom order.

Select Metal Layers for Analysis in Display Tab and then Run	Runs structural analysis.
Analysis Data Valid For Layers	Displays the names of layers for which data is available.
Select Net	Selects the net on the Virtuoso layout. **Note:** You can perform structural analysis only on one net at a time. So, if you are using the IR drop analysis results for performing structural analysis, ensure that only one net is selected on the *IR* tab. If you select *All Power Nets*, *All Signal Nets*, or multiple nets in the *Select Nets* option on the *IR* tab, an error message is displayed.
Clear	Clears any existing highlights in the layout.

© 2020 All Rights Reserved.

Show Unconnected from Top

Highlights all unconnected metal segments that have no top via/contact shapes overlapping them on layer basis. For this check, the result browser will show the following information:

Layer	Area (u^2)	IR (V)	# TopVias	# BotVias	Ratio
METAL_2	5118.98	0.955203	0	2830	0
METAL_2	5118.98	0.957325	0	2980	0

Show Unconnected from Bottom

Highlights all unconnected metal segments that have no bottom via/contact shapes overlapping them on layer basis.

Missing Via

Highlights overlapping metal segments that have missing vias in the overlapping area. For this check, the result browser displays only layer and area information as shown below.

Layer	Area (u^2)
METAL_4 - METAL_3	0.6174
METAL_4 - METAL_3	0.6174
METAL_4 - METAL_3	0.6174

Number of Top Via <

Highlights metal segments that have less than the specified number of top vias.

Number of Bottom Via <

Highlights metal segments that have less than the specified number of bottom vias.

Top / Bottom via Ratio >

Highlights metal segments that have more than the specified top to bottom via ratio.

Bottom / Top Via Ratio >

Highlights metal segments that have more than the specified bottom to top via ratio.

IR value >

Highlights segments that have less than the specified IR drop values.

Note: This check is performed at the segment level. For segments with multiple shapes, Voltus-Fi picks the worst-case IR values.

Overlap area (u^2) <

Highlights segments that have an overlapping area less than the specified overlapping area.

© 2020 All Rights Reserved.

Via Coverage Ratio (via area / overlap area) <	Highlights segments that have less than the specified via coverage ratio in the overlapping area. This check is used to highlight segments that have a via in the overlapping area but the via coverage is less as compared to the size of the overlapping area. For example, in the image below, the overlapping area of metal layers, M2 and M3, has a via, Via23. However, the ratio of via area to overlapping area is less than the user-specified ratio of 0.3. So, the overlapping area is highlighted.
Layer	Specifies the layer name for which information is provided.
Area (u^2)	Specifies the area of the segment or layer.
IR(V)	Specifies the worst IR drop in the node.
#TopVias	Specifies the number of top vias.
#BotVias	Specifies the number of bottom vias.
Ratio	Specifies the ratio of top-to-bottom or bottom-to-top vias.
Number of Violations	Specifies the number of violations for the selected check type.
Auto Zoom	Zooms into the violations for the specified check.
Cancel	Closes the form without executing the command.
Load	Loads the configuration file.
Save	Saves the configuration file.
Help	Opens help information in Cadence Help.

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

13

What-If EMIR Analysis

- Overview on page 446
- ECO Commands Supported in Voltus-Fi-XL on page 447
- Performing What-If (ECO) Analysis in Voltus-Fi-XL on page 448
- Pre-Filling the Run Simulation Form on page 456
- Output on page 457
- Options for Performing What-If (ECO) Analysis on page 458
- Layout Modification Options on page 466
  - Adding Pins and Modifying the Layout on page 466
  - Deleting Pins and Modifying the Layout on page 468
  - Adding Resistors and Modifying the Layout on page 469
  - Deleting Resistors and Modifying the Layout on page 470
  - Adding Vias and Modifying the Layout on page 470
  - Deleting Vias and Modifying the Layout on page 471
  - Design Rule Checking on page 472

© 2020 All Rights Reserved.

Overview

The What-If analysis feature lets you analyze the impact of potential layout changes, without implementing these changes in the layout, and re-extracting the xDSPF file. These changes are also called the ECO changes.

The What-If flow is supported in the regular “iterated” method EMIR flow, and in an incremental “itereco” method EMIR flow, which reuses the circuit simulation results from a previous iterated mode EMIR analysis. This is shown in the figure below.

Figure 13-1 The What-If Flow

The What-If changes are defined in an ECO file, which is included in the EMIR control file (`emir.config`). All ECO changes are relative to the original xDSPF content. In one What-If simulation, all changes defined in the ECO file are applied. These are then analyzed together in the second stage of the EMIR analysis.

Multiple incremental ECO runs are supported. For each run, the changes defined in the ECO file are applied to the original xDSPF content. Changes from previous ECO runs are not considered. In the incremental What-If analysis, only the nets with ECO changes are analyzed.

Other EMIR methods, such as “`method=direct`”, are not supported when performing What-If analysis.

All ECO changes for a specific What-If run are defined in one ECO file.

© 2020 All Rights Reserved.

ECO Commands Supported in Voltus-Fi-XL

The following ECO commands are supported in Voltus-Fi-XL:

- `add_pin net=<net_name> name=[node1 node2 …]`
- `delete_pin net=<net_name> name=[node1 node2 …]`
- `add_cap net=<net_name> cap=[node1 cvalue1 node2 cvalue2 …]`
- `add_res net=<net_name> node=<node1 node2> r=<rvalue>`
- `delete_res net=<net_name> name=<resistor1 resistor2 …>`
- `scale_res net=<net_name> name=<resistor1 resistor2 …. > scale = <scalefactor>`
- `add_via` when executed creates an `add_res` entry in the ECO file
- `delete_via` when executed creates a `delete_res` entry in the ECO file

The parameters listed above are defined as follows:

- `net_name`: name of the net as defined in the `xDSPF |*net` statement. There is no support for wildcards, same as that in the xDSPF file.
- `node1, node2`: name of pin, sub, or tap node. There is no support for wild cards, same as that in the xDSPF file.
- `cvalue1, cvalue2`: the value of capacitance to be added. Negative value is supported for reducing net capacitance. It is a floating number.
- `rvalue … value`: value of resistance to be added. This is a floating number.
- `resistor1, resistor2`: the instance name of resistors to be deleted, or scaled.
- `scalefactor`: the scaling factor. This is a floating number.

An example ECO file (`emir.eco`) is as follows:

```
add_pin net=[VDD] name=[VDD#9]
delete_pin net=[bias1] name=[Rj3652]
scale_res net=[VDD] name=[Rg1 Rg2 Rg3 Rg4] scale=0.5
delete_res net=[VDD] name=[Rg5]
add_res net=[VDD] node=[VDD#80 VDD#81] r=10
add_cap net=[VDD] cap=[VDD#80 10e-15 VDD#81 5e-15]
```

Alternatively, all ECO commands can also be used with the prefix, `eco_` in the `emir.conf` file directly. For example,

© 2020 All Rights Reserved.

```
eco_add_pin net=[VDD] name=[VDD#9]
```

When performing a regular iterated method EMIR analyses with What-If, the only change required in the `emir.conf` file is the reference to the ECO file.

```
solver method=iterated
eco file=-"emir.eco"
```

The following lines need to be defined in the `emir.conf` file to enable an incremental What-If EMIR analysis.

```
solver method=itereco inputwf="./input.emirtap.pwl"
eco file="emir.eco"
```

Performing What-If (ECO) Analysis in Voltus-Fi-XL

The What-If flow in Voltus-Fi-XL GUI is as follows:

- ECO commands are specified in the Voltus-Fi-XL GUI and the ECO file is generated.
- Voltus-Fi-XL calls Spectre to perform the EMIR analysis. Fully accurate EMIR simulation is performed.
- Only nets that are changed in the ECO statement are changed.

To perform What-If analysis, first load the IR drop analysis results, select the net and click *Show Plot*. In the IR/EM Results form, the *What-If (ECO) Analysis* tab gets enabled. The software provides support in the form of tips about how to enable the *What-If (ECO)* tab when you hover over the tab name. This is shown below. It also provides help to fill out information in the various fields in the tab. This tooltip appears when you hover over the fields in the form.

Figure 13-2 Tooltips for What-If (ECO) Analysis Tab of the IR/EM Results Form

© 2020
All Rights Reserved.

Figure 13-3 the IR/EM Results Form – What-If (ECO) Tab

In this form, provide the following information:

- Select the nets on which you want to perform What-If (ECO) analysis. In the text field provided with the *Select Nets* field, enter the full net name. The text field also accepts regular expression. For example, if you enter `V*`, the table will list all nets whose names start with `V`, for example `VDD` and `VSS`.

© 2020
All Rights Reserved.

- From the *whatif changes* list, select the ECO command you want to run. The following commands are available:
  - `add_pin`
  - `delete_pin`
  - `add_cap`
  - `scale_res`
  - `add_res`
  - `delete_res`
  - `add_via`
  - `delete_via`

  Note: For each of the above options, there are some changes in the fields enabled in the form. For details, see Options for Performing What-If (ECO) Analysis.

In the *Node name* group box, provide the information for the node on which the What-If (ECO) analysis is to be performed.

- Select the *Layer* on which you want to perform the analysis from the list. By default, this field is populated with the name of the layer selected in the Display form. If multiple layers are selected in the Display form, all the layer names will appear in the list.
- Click *Get Node* and then select any point on the specified layer in the layout. The nearest node on the specified layer is identified for the whatif changes. The node name and its x- and y- coordinates are populated in the three text boxes; *Name*, *X*, and *Y*, respectively.

In the *Recurrence* group box, provide the following information:

- Click *Recurrence* to add pins in recurrence in both horizontal and vertical directions. The starting point is the node that was specified using the *Get Node* button in the previous step. This option is applicable only when `add_pin` ECO command is selected. For this, perform the following tasks:
  - Specify dX, or delta-X, which is the horizontal spacing between the nodes that will be used for adding pins.
  - Specify dY, or delta-Y, which is the vertical spacing between the nodes that will be used for adding pins.
  - Specify *num*, which is the number of nodes with the specified spacing, both horizontally and vertically.

© 2020 All Rights Reserved.

When you click *Recurrence*, the Filters pop-up window opens. It displays a table with names of nodes that will be used for adding pins along with their x- and y-coordinates. When you click *OK*, the selected nodes will be used to add pins. This information is then populated in the table provided in the What-If (ECO) form. This is shown below.

Figure 13-4 Adding Multiple Pins for What-If (ECO) Analysis

- In the *Node-2* group box, provide the same information as above for the second node. This field is only enabled when `add_res` is specified in the *whatif changes* list.

  Note: You can choose a node either on the same layer or on a different layer.

© 2020
All Rights Reserved.

- Click *Add data* to add the location of the nodes in the table provided in the form. You can add multiple pairs of co-ordinates in the table. The table provides information about the type of What-If (ECO) analysis, the net name, information about the nodes, and the analysis data.
- Click *layout-modify* to modify the layout of the ECO commands. The button lets you modify the layout for the specified ECO command. For details, see Layout Modification Options.
- Specify the *Layer* on which you want to create the bounding box.
- Click *Bounding box* and then select an area on the Virtuoso layout. Depending on the what-if analysis type, nodes and resistors will be selected from the layout. For example, if `delete_res` ECO command is selected in the *whatif changes* list, resistors will be selected for deletion. The nodes and resistors are selected on the basis of the specified layer name. These nodes will be listed in the Filters pop-up window. When you click *OK*, all the selected nodes will be used for deleting resistors.

  If `add_pin` command is selected in the *whatif changes* list, then all nodes inside the rectangle, for the specified layer, will be selected. These nodes will be listed in the Filters pop-up window. When you click *OK*, all the selected nodes will be used for adding pins. Their names and the corresponding net names will be added in the table in the What-If (ECO) form.

  Note: The Bounding Box can only be used when `add_pin`, `add_cap`, `delete_res`, and `scale_res` ECO commands are selected in the whatif changes list. An example of using the Bounding Box for adding pins is shown below.

© 2020 All Rights Reserved.

Figure 13-5 Using the Bounding Box Option to Add Pins

- Specify the path and name of the ECO file to be generated in the *ECO file* field.
- Click *Delete row* to delete the data added in the table. For this, first select the row for which you want to delete the data and then click *Delete row.*
- Click *Generate ECO* to generate the ECO file with the specified filename in the specified directory. The file opens in a pop-up window. A sample ECO file is shown below.

© 2020 All Rights Reserved.

Figure 13-6 Sample ECO File

- Click *Run ECO to* run the simulation with the specified What-If ECO changes. When you click *Run ECO*, the Run Simulation form opens. The fields in this form are filled in automatically with information from the Spectre run. For details of how the form is pre-filled by the software, see Pre-Filling the Run Simulation Form.

  In this form, check all the pre-filled information to ensure that it is correct. This form is shown below.

© 2020 All Rights Reserved.

Figure 13-7 The Run Simulation Form

In this form, the following information is pre-filled:

- The name of the simulation *Run directory*
- The name of the input *Ckt file*
- The name of the simulation configuration file in the *Conf file* field

  Note: You can edit the *Ckt file* and the *Conf file* by using the edit buttons provided against these options.

- The path of the output directory for the What-If Spectre run in the *Output dir* field, and
- The Spectre command line parameters in the *Other cmd line options* field, for example, if Spectre XPS is used, then `+spice +xps` and `+cktpreset=sram` is specified and if Spectre APS is used, then `+spice +aps` and `+cktpreset=sram` is specified

  Note: For details of command line options, see “Postlayout Simulation” chapter in

© 2020 All Rights Reserved.

the *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

- Click *Run Simulation*. Once the simulation run is complete, you will get a pop-up telling you that the run is successful. The details of the run are provided in the Run log output.

  Errors, if any, are written in the log. You can fix the errors and run the simulation again.

- Once the simulation run is complete, the *Load ECO Results* button is enabled. This button is used to load the simulation results file that includes the What-if ECO changes. When you click this button, the Run Simulation form closes and the *IR* tab of the IR/EM Results form opens. In this tab, the new results file is loaded, which includes information about only those nets for which what-if ECO changes have been made.

  You can now generate IR/EM plots as per the changes made in ECO simulation.This is shown below.

Figure 13-8 IR/EM Results Form – Results File with What-If ECO Changes

Pre-Filling the Run Simulation Form

The Run Simulation form that opens when you click the *Run ECO* button in the *What-If (ECO)* analysis tab, is pre-filled with information that is already available from the Spectre run. To ensure that the form is pre-filled, the following information must be available:

- The emir bin file that contains the simulation results from the original Spectre run
- The configuration file generated during the original Spectre run
- The log file generated during the original Spectre run

© 2020 All Rights Reserved.

The software performs below steps to extract the data from the original Spectre run for pre-filling the Run Simulation form.

1. The software locates the log file generated by Spectre. The “`xps.log`” or the “`aps.log`” file is located either in the same directory as the emir bin file or in the parent directory of the emir bin file.
2. The software parses the "`.log`" file and searches for the "command line" used in the original Spectre run. The command line contains the "ckt file" and "command line options" used in the original Spectre run.
3. It then creates a directory parallel to the directory containing the "ckt file" for the What-If Spectre run. The new directory is created with the name, "`whatIfRunDir`". This will contain the output of the What-If Spectre run.
4. The software searches for the “conf” file generated after the original Spectre run in the emir bin file directory and then creates a new "conf" file for the What-If spectre run, in the directory "`whatIfRunDir`" and performs the following tasks in the newly created "conf" file:
   - Copies the data from the "conf" file from the original Spectre run
   - Changes the solver method from "`iteronly`" to "`itereco`"
   - Appends the ECO file, "`eco file:ECO_filename`" to the conf file

Now, when you click *Run Simulation*, the simulation runs using the above newly-generated conf file that includes information about the ECO options.

Video

To view the demonstration on performing what-if analysis in Voltus-Fi, see Performing What-If Analysis using Voltus-Fi video.

Access to this video will depend on the availability of a web browser and a Cadence Online Support account.

Output

The output of the above simulation run with ECO changes is a simulation database (`*.emir#_bin)`. You can use this simulation database for running the EMIR analysis.

For this, load this simulation database—with ECO changes—when loading the IR drop analysis results on the IR tab of the IR/EM Results form.

The subsequent EMIR analysis will be performed using this simulation database.

© 2020 All Rights Reserved.

Options for Performing What-If (ECO) Analysis

When you select the ECO command that you want to run in the *whatif changes* list, there are some changes in the other options on the What-If (ECO) tab. These are detailed below.

- For `add_pin`, the following flows are supported for What-If (ECO) analysis:
  - You can select a node by retrieving the x- and y- coordinates information in the layout. This node will be used for adding pins.
  - You can add multiple nodes at specified distances from an identified node at once by using the *Recurrence* option provided in the What-If (ECO) Analysis form. These nodes are used to add pins.
  - You can create a bounding box in the layout for a specified layer. All nodes within the bounding box or the rectangle will be used to add pins.

  The fields on the What-If (ECO) tab are the same as described above and shown in Figure 13-3 on page 449. For details, see Performing What-If (ECO) Analysis in Voltus-Fi-XL.

- For `delete_pin`, the following options are available in the What-If (ECO) Analysis form:
  - The *Node name* group box is replaced with the *Pin name* group box. This is shown below.
  - The *show pins* button is enabled. Click this button to view the list of pins on the selected layer. The Filters pop-up opens. The pin name and the x- and y-coordinates of the pin are provided in the pop-up. Select the row and click *OK*. The pin name and its coordinates appear in the table. This is shown below.

© 2020 All Rights Reserved.

Figure 13-9 Options for delete_pin What-If (ECO) Analysis

© 2020
All Rights Reserved.

Figure 13-10 The Filters Pop-Up Window

- For `add_cap`, the following options are available in the What-If (ECO) form:
  - Specify the *Cap value(F)*, which is the capacitance value for the analysis in femtofarads (`fF`).
  - Provide the node information in the same manner as that detailed for `add_pin` option.

Figure 13-11 Options for add_cap What-If (ECO) Analysis

© 2020
All Rights Reserved.

- For `scale_res`, the following options are available in the What-If (ECO) form:
  - Specify the *Scale factor(Ohm)* for the analysis.
  - The *Node name* group box is replaced with the *Resistor* group box. Provide the *Resistor* information in the same manner as the node information detailed for the `add_pin` option.

Figure 13-12 Options for scale_res What-If (ECO) Analysis

- For `add_res`, the following options are available in the What-If (ECO) form:
  - Specify the *Res value (Ohm),* which is the resistance value for the analysis.
  - Both the Node group boxes, *Node1* and *Node2* are enabled for this option. You can provide the resistor information for multiple nodes.

© 2020 All Rights Reserved.

Figure 13-13 Options for add_res What-If (ECO) Analysis

- For `delete_res`, the following options are available in the What-If (ECO) form:
  - The *Node name* group box is replaced with the *Resistor* group box. Provide the *Resistor* information in the same manner as the node information detailed for the `add_pin` option.

© 2020 All Rights Reserved.

Figure 13-14 Options for delete_res What-If (ECO) Analysis

- For `add_via`, the following options are available in the What-If (ECO) form:
  - The *Node name* group boxes are replaced with the *Via-node- and Via-node-2* group boxes. Provide the *Res value (Ohm)* information.

  The two via nodes that are specified must be on different layers.

© 2020 All Rights Reserved.

Figure 13-15 Options for add_via What-If (ECO) Analysis

- For `delete_via`, the following options are available in the What-If (ECO) form:
  - The *Node name* group box is replaced with the *Via* group box.

  Via layer must be selected in the *Layer* box.

© 2020
All Rights Reserved.

Figure 13-16 Options for delete_via What-If (ECO) Analysis

© 2020
All Rights Reserved.

Layout Modification Options

Voltus-Fi-XL provides options to modify the layout for the specified ECO command in the GUI mode. For this, *layout-modify* button is provided on the *What-If (ECO)* tab of the IR/EM Results form. This feature lets you view the layout modifications after adding and deleting pins, resistors, and vias.You can also perform design-rule checks (DRCs) and view the design-rule violations (DRVs) in the layout after performing these actions.

Note: This feature is only available when the design is opened in the Virtuoso Layout Suite (VLS) XL.

The following topics are covered in this section:

- Adding Pins and Modifying the Layout
- Deleting Pins and Modifying the Layout
- Adding Resistors and Modifying the Layout
- Deleting Resistors and Modifying the Layout
- Adding Vias and Modifying the Layout
- Deleting Vias and Modifying the Layout
- Design Rule Checking

Adding Pins and Modifying the Layout

On the *What-If (ECO)* tab of the IR/EM Results form, perform these steps to add a pin and modify the layout:

1. Select *add_pin* ECO command from the *Whatif changes* list.
2. Click *Get Node* and then click anywhere on the Virtuoso layout to get a sub-node on the net.
3. Click *Add data* to add the ECO command, `add_pin` in the table.
4. Click the *layout-modify* button provided in the table to add the pin in the layout. The default pin shape is `0.1uX0.1u`.

© 2020 All Rights Reserved.

Figure 13-17 Adding a Pin and Modifying the Layout – Step 1

© 2020
All Rights Reserved.

Figure 13-18 Adding a Pin and Modifying the Layout – Step 2

Deleting Pins and Modifying the Layout

Follow these steps to delete a pin and modify the layout:

1. Select *delete_pin* ECO command from the *Whatif changes* list.
2. Click *show pins* to view the list of pins. Select the pin you want to delete. The `delete_pin` ECO command is added in the table.
3. Click *layout-modify* to delete the pin in the layout.

© 2020
All Rights Reserved.

Adding Resistors and Modifying the Layout

Follow these steps to add a resistor and modify the layout:

1. Select *add_res* ECO command from the *Whatif changes* list.
2. Select a metal layer from the *Node layer* group box and get the *Node 1* and *Node 2* information from the layout. Both nodes should be on the same layer.
3. Specify the *Res value (ohm).*
4. Click *Add data* to add the ECO command, `add_res` in the table.
5. Click *layout-modify* to add the resistor in the layout. The new resistor created in the layout will have a default wire width of `0.1u`.

Figure 13-19 Adding a Resistor and Modifying the Layout

© 2020
All Rights Reserved.

Deleting Resistors and Modifying the Layout

Follow these steps to delete an MMSIM/Spectre resistor and modify the layout:

1. Select *delete_res* ECO command from the *Whatif changes* list.
2. Select a metal layer and get the resistor information from the layout.
3. Click *Add data* to add the ECO command, `delete_res` in the table.
4. Click *layout-modify* to delete the resistor from the layout. Design rule violations (DRVs) will be checked after modifying the shape.

Figure 13-20 Deleting a Resistor and Checking the DRVs

Adding Vias and Modifying the Layout

Follow these steps to add a via and modify the layout:

1. Select *add_via* ECO command from the *Whatif changes* list.

© 2020
All Rights Reserved.

2. Select a metal layer and select *Node 1* from the layout.
3. Select another layer and select *Node 2* from the layout.
4. Click *Add data* to add the via in the layout. The software searches for existing via definition (viaDef) in the technology file based on the via layers selected by you. The following scenarios are possible.
   a. If the via definition is available, a new via is created.
   b. If the via definition is not available, the tool errors out.
5. The generated via has the following default values:
   c. Number of cut rows=3
   d. Number of cut columns=3

Figure 13-21 Adding a Via and Modifying the Layout

Deleting Vias and Modifying the Layout

Follow these steps to delete a via and modify the layout:

© 2020
All Rights Reserved.

1. Select *delete_via* ECO command from the *Whatif changes* list.
2. Select a via layer and select a via from the layout.
3. Click *Add data* to add the ECO command, `delete_via` in the table.
4. Click *layout-modify* to delete the via from the layout.

Design Rule Checking

Design rules are checked during layout modifications. Virtuoso XL DRD (Design-Rule-Driven) editing is automatically turned ON by Voltus-Fi while performing the layout modifications. The violations can be viewed in Annotation Browser and the layout window.

The figure below shows the DRVs post layout modifications after deleting a via.

Note: This feature is only available when the design is opened in the Virtuoso Layout Suite XL.

Figure 13-22 Checking DRVs after Deleting a Via

© 2020
All Rights Reserved.

14

Power-Grid View Generation

- What is Power-Grid View?
- PGV Generation in Voltus-Fi-XL
- Data Requirements and Flow for PGV Creation
- Open Access (OA)-Based Library Generation
- Automatic Generation of Layer Map Files
- Generating Power-Grid Views in Voltus-Fi-XL
- Troubleshooting Information for PGV Generation in Voltus-Fi-XL

© 2020 All Rights Reserved.

What is Power-Grid View?

A power-grid view (PGV) is a model view of the design that contains information about the power ports of the cell or block. It also contains information about the internal power grid, intrinsic capacitance, and the tap current distribution within the cell or block.

PGV Generation in Voltus-Fi-XL

Increasing operating frequency and density of IC designs increases the need for IR drop and Electromigration (EM) power-grid analysis. Power-grid signoff analysis is provided by Voltus in the digital domain, and Voltus-Fi-XL in the analog domain. Voltus-Fi-XL provides the ability to create power-grid views (PGVs) of an analog design block that can be used in Voltus for mixed-signal analysis. PGVs can be generated both in the GUI and batch command mode in Voltus-Fi-XL.

PGVs provide an accurate characterization of capacitance and currents, and power-grid extraction. Voltus uses these views to model power rail and power distribution information for each instance of the cell in the design for dynamic or static full-chip power-grid analysis.

Note: Both Voltus and Voltus-Fi-XL support PGV generation only for power pins.

For more information about PGVs, refer to the “Power-Grid Library Generation” chapter in the *Voltus IC Power Integrity Solution User Guide*.

Related Topics

create_pgv, Generating Power-Grid Views in Voltus-Fi-XL, Create Power Grid Views Form

Data Requirements and Flow for PGV Creation

Power-grid view generation in Voltus-Fi-XL depends upon the simulator-generated database and current files. For generating PGVs in Voltus-Fi-XL, ensure the following while performing simulation using Spectre APS/XPS.

- Enable the `emir` mode. The EMIR analysis is run like a regular APS/XPS simulation with the additional `+emir` option. For details, see Simulation Requirements and Setup in the “Data Preparation” chapter.
- Generate the current files for all the power nets for which PGVs are being generated. This is done by specifying UTI commands in the Spectre input file. The syntax of the command is as follows:

© 2020 All Rights Reserved.

```
uti0 uti signal=net_name start=start_time
cycles=number_of_cycles_for_this_measure intervals=number_of_points
clockcycle=clock_period method=[0|1|2][0|1|2][0|1|2]
filename=filename_prefix_for_this_measure termflag=0 Namemangling=0
```

The syntax contains the following parameters:

- `signal=signal_name`

  Specifies the net name for which current is to be calculated.

  Note: Ensure that the net name is specified along with the instance hierarchy name. For example, if your instance hierarchy name is `IO`, then specify the signal name as `IO.VDD` for the `VDD` net.

- `start=start_time`

  Specifies the start time from which the current will be written in the current file.

- `cycles=number_of_cycles_for_this_measure`

  Specifies the number of clock cycles to be captured in the current file. By default, one cycle is captured.

- `intervals=number_of_points`

  Specifies the number of time points or measurement intervals within a clock cycle.

- `clockcycle=clock_period`

  Specifies the length of the clock cycle or the total duration for which the current information is being written in the current file. You must specify the unit.

- `method=[0|1|2][0|1|2][0|1|2]`

  Specifies the method that the simulator uses in post-processing clock analysis data.

  `[012]` The number in the first position, `(0)`, specifies the method of peak current calculation.

  `[012]` The number in the second position, `(1)`, specifies the method of average current calculation.

  `[012]` The number in the third position, `(2)`, specifies the method of RMS current calculation.

  `0`: Reports the maximum out of each interval over all periods.

  `1`: Reports the average out of each interval over all periods.

  `2`: Does not calculate peak, average, or RMS current.

© 2020 All Rights Reserved.

The default is `212`, which is average current calculation.

- ❑ `filename=`*`filename_prefix_for_this_measure`*

  Specifies the prefix for the current files that are being generated.

  Note: For Voltus-Fi-XL, ensure that the prefix provided is `"vps"`. For example, your filename could be `vps_VDD.ptiavg`.

- ❑ `termflag [0 | 1]`

  Specifies the terminals which will be output. Specify `0` to output the first three terminals. Specify `1` to output all the terminals.

- ❑ `Namemangling`

  Specifies namemangling. When set to `0`, specifies that names should not be mangled.

  Here is an example of this command.

```
uti0 uti signal="IO.VDD" start=0n cycles=1 intervals=133 clockcycle=2ns
method="212" filename="vps" termflag=0 Namemangling=0
```

- In the Spectre input emir conf file, specify `eisopt save_gnd_caps=1` to include the grid capacitances in the EMIR database so that they can be used in power-grid flow. For example:

```
net name=[I0.*] analysis=[iavg irms vmax vavg]
eisopt save_gnd_caps=1
emirutil autorun=[true]
```

The following table shows the required inputs for generating power-grid views in Voltus-Fi-XL.

Table 14-1 Inputs Required for PGV Generation

Type of Inputs	Details
`Simulation database`	It is required as input for performing EMIR analysis in Voltus-Fi-XL.
`Quantus Techfile`	It is required for reading the technology data for PGV generation.
`LEF DEF file`	It includes the LEF information for the technology data. It is required for port connectivities.

© 2020 All Rights Reserved.

Type of Inputs	Details
`Macro LEF file`	It specifies the LEF information for the design. **Note**: In case the technology data and design data are in the same LEF file, you do not need to specify the Macro LEF file separately.
`OA Library`	It specifies the OpenAccess (OA) database library name for PGV generation.
`PGDB layer map file`	It provides mapping between the layer names in the xDSPF (simulation EMIR database) to the layer names in the technology file.
`LEF DEF layer map file`	It provides mapping between the layer names in the LEF file to the layer names in the technology file.
`Net names`	It is required for pin information.

The following diagram illustrates the flow of PGV generation in Voltus-Fi-XL:

Figure 14-1 Power-Grid View Generation Flow in Voltus-Fi-XL

© 2020
All Rights Reserved.

Open Access (OA)-Based Library Generation

Voltus-Fi-XL lets you specify an OpenAccess (OA) database for PGV Generation. You can either specify the LEF technology file/Macro LEF file or the OA libraries. The OA library files contain information from both the LEF technology file and the Macro LEF file.The layer map file is automatically created from the specified OA file. Alternatively, you can specify the layer map file.

In the GUI, the OA libraries can be specified in the *OA Libs* field of the Create Power Grid Views form. This is shown in the subsequent section.

In batch mode, the OA library can be specified by using the `-oa_lib` parameter of the create_pgv command.

Important

The OA library should have either the abstract view or the layout view to read the required LEF pin information. The tool reads the abstract view to get the pin information and if this view is not present, it can read this information from the layout view.

© 2020 All Rights Reserved.

Automatic Generation of Layer Map Files

The PGV flow in Voltus-Fi requires the following layer map information:

- LEF layer map file: Provides the mapping between the LEF layers and layers in the technology file.
- PGDB layer map file: Provides the mapping between the xDSPF layers and layers in the technology file.

These layer map files can either be specified in batch mode and the GUI. However, if not specified, the software generates these files for PGV generation.

Automatic Generation of the LEF layer map File

The following parameter of the `create_pgv` command is used to specify the LEF layer map file:

```
create_pgv -lef_layer_map filename
```

The above parameter is optional. If specified, the software uses the specified layer map file for running the PGV flow. If not specified, Voltus-Fi automatically generates the LEF layer map file.

Note: Voltus-Fi generates the LEF layer map only for metal layers. The layers below the metal layers are not included in the mapping. This means, if a macro LEF file has ports in lower layers, they will not be included in the mapping.

The PGDB layer map file provides a mapping between the DSPF layer names that are written in the simulation database and the technology file layer names. Multiple DSPF layers can be mapped to one technology file layer.

The following parameter of the `create_pgv` command is used to specify the PGDB layer map file:

```
create_pgv -pgdb_layer_map pgdb_layermap_file_name
```

To generate the layer map file automatically, do not specify the above option. In this case, the software generates the PGDB layer map file with the name, `"VFI_generated_pgdb_layer_map.txt"`.

Note: The DSPF file should include the "`*LAYER_MAP`" section information.

An example of the `*LAYER_MAP` section for a third-party DSPF file is shown below:

© 2020 All Rights Reserved.

```
*LAYER_MAP
*0 SUBSTRATE
*1 AP        ITF=AP
*2 M0_A      ITF=M0
*3 M0_B      ITF=M0
*4 M1_A      ITF=M1
*5 M1_B      ITF=M1
*6 M0_A_SRM_SP    ITF=M0
*7 M0_B_SRM_SP    ITF=M0
*8 M0_A_SRM_2P    ITF=M0
*9 M0_B_SRM_2P    ITF=M0
*10 M1_A_SRM_SP    ITF=M1
```

An example of the `*LAYER_MAP` section for a Quantus DSPF file is shown below:

```
*LAYER_MAP
*0 none
*1 M4
*2 M3
*3 M2
```

Note: This feature is available only with the Quantus versions, EXT 16.1.x and later.

Note: If the DSPF file does not contain the "`*LAYER_MAP`" section, an error message is displayed asking the user to specify the PGDB layer map by using the `-pgdb_layer_map` parameter of the `create_pgv` command.

Workaround for Missing Layer Information in the Auto-Generated PGDB layer Map File

If the software issues a warning message during the auto-generation of the layer map file, such as the one shown below, it means these layers are not mapped in the VFI-generated PGDB layer map file. In this case, add these layers manually in the file.

```
Warning: The following layers from dspf are missing in the pgdbLayerMap file
specified. Please add these layers in pgdblayermap file. Missing layers may cause
connectivity problems in PGV creation for the net VDD.
mwires bwires
```

If there are no warnings, the VFI-generated PGDB layer map file can be used for running the PGV flow without making any changes.

© 2020 All Rights Reserved.

Generating Power-Grid Views in Voltus-Fi-XL

Voltus-Fi-XL creates PGVs for both static and dynamic rail analysis. The software creates PGVs for static rail analysis along with the PGVs for dynamic rail analysis by using the average values of the current waveforms. However, it does not create PGVs only for static rail analysis.

To create power-grid views in Voltus-Fi-XL, perform the following steps.

1. In the Voltus-Fi-XL console, choose *IR/EM Analysis – Create Power Grid View*.

© 2020 All Rights Reserved.

The Create Power Grid Views form is displayed.

2. In the *LEF Files and Layermap Files* group box, specify the various filenames.

- Specify the *OA Libs*, which is the name of the OA database library.

  Note: If this option is specified, you do not need to specify the *Tech Lef File/Macro*

© 2020 All Rights Reserved.

Lef File options.

- Specify the *Macro Name,* which is the top cell name.
- Specify the *Library Name,* which is the name of the library. This option is used to create a library with a name different than that of the macro. By default, a library is created with the same name as that of the macro.
- Specify the *Extraction Tech File*, which provides the name of the technology file that will be used for reading the technology data.
- Specify the *Tech Lef File*, which includes the LEF information for the technology data.
- Specify the *Macro Lef File, which includes the* LEF information for the design.

  Note: In case the technology data and design data are in the same LEF file, use any of the two options, *Tech Lef File* or *Macro Lef File* to specify the filename.
- Specify the Lef *Def Layer Map File*, which provides the mapping between LEF layer names and technology layer names.
- Specify the *PGDB Layer Map File,* which provides the mapping between xDSPF layer names that are written in the simulation database, and the technology layer names.

  Note: If an xDSPF layer name is missing in this file, the PGVs will not contain data for that layer.

  Note: For more information about the LEF layer map file and PGDB layer map file formats, see LEF layer Map File and PGDB layer Map File in the “File Formats” chapter.

  The Tech Lef file, the Macro Lef file, the Lef Def Layer Map file, and the PGDB Layer Map file can be created and edited using the edit buttons provided against each of these fields.
- Specify the *Simulation Directory*, which contains the current (`*.ptiavg`) files.

  Note: The PGV generation fails if this directory does not contain the current files for all the nets for which the PGVs are being generated.
- Specify the *Simulation Prefix*, which is the prefix used for creating an instance while performing the simulation. The prefix is specified along with the hierarchy separator. This input is optional.

  For example, you can specify “`XO.`” or “`IO.`”. In these examples, `“XO”` or `“IO”` are prefixes `and “.”` is the hierarchy separator. You can check the prefix used during simulation in your simulator input file.

© 2020 All Rights Reserved.

- Specify the *UTI Ron File* that is generated by Spectre. This file contains multiple on-resistance (Ron) values, that is minimum, maximum, and average, for powergates. A sample file is shown below:

```
D-Term-Net  S-Term-Net  Ron_avg(Ohm) Ron_max(Ohm) Ron_min(Ohm) Power-Gate
outp        VDD         1268.34      1302.38      1230.52      MPM3
outp        VDD         1268.03      1302.06      1230.2       MPM3@39
outp        VDD         1268         1302.04      1230.18      MPM3@40
outp        VDD         1268.03      1302.07      1230.21      MPM3@41
```

  If this file is not specified, Voltus-Fi-XL uses the `Ron` values from the MMSIM simulation database (`*.emir#_bin` file).

- Specify the *Trigger File*, which is a text file that is used during dynamic current simulation to run a detailed analysis for determining current distribution. For detailed information about the trigger file format and content, see Format and Contents of Trigger File in the “File Formats” chapter.

- Specify the *Net Name* and the corresponding *Voltage*.

- Specify the *State Directory / Results File*, which is the simulation result file. The naming convention of the simulation result file is, `*.emir#_bin`. For example, your result file could be named, `xps.emir0_bin`.

- Click *Add* to add the net, the corresponding voltage, and the EMIR Results DB file location to the list.

- Click *Remove* to remove a net from the list.

3. In the *Output group* box, specify the *Results Directory*, which is the output directory for the power-grid views. By default, the power-grid views are stored in the work directory.

When you specify the directory in which you want to save the PGV generation results, the software gives a warning if there are any issues with the specified directory. For example, if you provide a file path instead of a directory path or a directory path that is read-only, the software gives a warning. An example of a warning message for the latter scenario is shown below.

Note: The *Results Directory* field lets you save different outputs without overwriting the

© 2020 All Rights Reserved.

existing ones. However, if the output is stored in the default directory, then the new output file created while generating the power-grid views in the current run overwrites the existing one.

4. Click *OK*.

The following output is obtained after generating PGVs in Voltus-Fi-XL:

- A binary database that contains geometric views of the cell, port information, and the power-grid views.
- Text report and summary file with detailed information about the power-grid views.

The output of PGV generation in Voltus-Fi-XL is used in Voltus for dynamic power calculation and static and dynamic IR drop analysis. For more information, see the “Power-Grid Library Generation” chapter in *Voltus IC Power Integrity Solution User Guide*.

Troubleshooting Information for PGV Generation in Voltus-Fi-XL

After generating the PGVs in Voltus-Fi-XL, you can check whether the PGVs are correctly generated or not. Listed below are a couple of issues that you may encounter while generating PGVs, and the information to help resolve them.

- **No current information in the PGVs**

  If there is no current information in the PGVs, check your log file for a warning titled, VOLTUS_LGEN-3065. This warning is issued when there is a mismatch between taps in the current files and the simulation database. If all taps fail, the current value in the PGV will be zero. To check the names and number of taps in the current file, you can use Itaputil. Itaputil is a Cadence utility that either reports characteristics of tap current files or extracts portions of the data and places them into a new tap current file.

  For example, if your current file is `vps_VDD.ptiavg`, you can use the following command to list the names of all the taps in this file:

  ```
  itaputil list vps_VDD.ptiavg > names
  ```

  Note: To get help on the complete list of available commands in Itaputil, type the following in the xterm or shell:

  ```
  itaputil -h
  ```

- **The current file does not have any waveforms**

© 2020 All Rights Reserved.

If the current file does not have any waveforms, it could be because of an error in the UTI command used for generating the current file. For resolving this issue, use Itaputil to check the tap names in the current file and to write the current data in the file.

© 2020 All Rights Reserved.

15

Hierarchical EMIR Analysis

- Hierarchical EMIR Analysis in Voltus-Fi-XL
- The High-Level Flow on page 490
- Compatible Software Versions on page 492
- Data Requirements for the Hierarchical EMIR Flow on page 492
- Types of Block-Level xDSPF – Detailed, Port, and Via Snapped
- PWL Current Files for Top-Level Simulation
- Hierarchical EMIR Analysis Flow in GUI Mode
  - Opening the Hierarchical EMIR Forms
  - Creating the Block PGVs and xDSPF on page 496
  - Generating the Top-Level xDSPF File on page 500
  - Running the Top-Level Simulation on page 508
- The Hierarchical EMIR Analysis Batch Mode Flow on page 512
  - set_mmsim_pgv_nets on page 513
  - create_pgv on page 514
  - extract_xdspf on page 517
  - set_block_data on page 519
  - run_top_simulation on page 521
- The Hierarchical EMIR Analysis Batch Mode Automation Flow
  - Block PGV Generation
  - Block PGV Inputs for Top-Level Extraction and Simulation
  - The Top-Level Extraction and Simulation Run

© 2020 All Rights Reserved.

- Multi-Mode Support
- A Sample Script for Automated Batch Mode Flow

■ Analyzing the EMIR Results for Hierarchical Analysis on page 532
- Loading the IR/EM Results for the Hierarchical Flow in the GUI on page 532
- Viewing Plots for Top and Block-Level Results on page 534
- Analyzing the EMIR Results on page 536

■ Sample File Formats for Hierarchical EMIR analysis on page 538
- Sample Block-Level xDSPF Files on page 538
- A Sample Top-Level xDSPF file on page 541
- A Macro LEF File on page 543

© 2020 All Rights Reserved.

Hierarchical EMIR Analysis in Voltus-Fi-XL

Voltus-Fi is used for transistor-level power-integrity analysis that includes simulation and EM and IR drop (EMIR) analysis for transistor designs captured in the xDSPF netlist extracted from the layout. The target designs include analog, analog/mixed-signal (AMS), and custom digital designs of large sizes that are created using Virtuoso.

In the current scenario, shrinking geometries and increasing design complexities have increased the probability of chip failures if the designs are not accurately analyzed for either EM or IR problems.

The increase in the size of the designs, to include several million instances and an even larger number of nodes, might result in exceeding the simulation capacity required to solve the complete design. In such scenarios, it is imperative to apply an hierarchical approach to simulate these designs.

Hierarchical analysis is useful for simulating large top-level designs that have multiple levels. Flat simulation and analysis of such designs uses a lot of space and increases the run time tremendously.

The Voltus-Fi Hierarchical EMIR flow enables the hierarchical signoff for AoT (Analog on Top) designs with a cockpit that is fully integrated within Virtuoso, similar to the flat Voltus-Fi EMIR flow. The Hierarchical analysis results are displayed on the Virtuoso layout with all the features and capabilities of flat Voltus-Fi EMIR analysis.

Important

This feature is only available when the VTS-Fi-AA license is checked out. This is an advanced license that is not available as part of the Voltus-Fi-XL license.

The Voltus-Fi hierarchical EMIR flow has the following advantages:

- Lets you analyze and fix EMIR violations for the block and top-level of the design, together and separately.
- Provides a single interface for the complete hierarchical analysis.
- Provides flexibility to choose either port, detailed, or reduced models of blocks to be used for top-level signoff.
- Provides high performance and high simulation capacity as compared to the flat flow with comparable accuracy – EMIR signoff for large designs.
- Provides support for multiple ports as blocks.

© 2020 All Rights Reserved.

The High-Level Flow

The integrated hierarchical EMIR flow involves Quantus, MMSIM, and Voltus-Fi working in an integrated manner.

Following are the key features of this flow:

- The PGVs (power grid-views) are leveraged to create a model of the block to enable the hierarchical EMIR flow within Voltus-Fi.
- The PGVs and a block-level xDSPF file is written out.
- Extraction is performed using Quantus and the block PGVs are used to connect from the top level to the block level. The output is a top-level xDSPF file.
- Multi-mode simulation (MMSIM) is performed using Spectre® Accelerated Parallel Simulator (APS)/eXtensive Partitioned Simulator (XPS) to process the top-level xDSPF with the block-level xDSPF.
- The top-level and block-level EMIR analysis results are analyzed in Voltus-Fi and the plots are displayed on the Virtuoso layout.

This following diagram illustrates the above flow.

© 2020 All Rights Reserved.

Figure 15-1 The High-Level Flow

The Voltus-Fi Hierarchical EMIR flow uses the extraction and simulation tools in the flow and the Voltus-Fi cockpit drives the flow. The sequence of steps in the Hierarchical flow within Voltus-Fi are listed below.

- Generating block-level PGVs and block-level xDSPF
- Running Quantus for generating the top-level xDSPF
- Running top-level simulation using both block-level and top-level xDSPF files
- Displaying the EMIR plots for top and block levels and analyzing the results

The above steps are detailed in subsequent sections of this document. The following diagram illustrates the above flow.

© 2020 All Rights Reserved.

Figure 15-2 The Hierarchical EMIR Flow in Voltus-Fi

Compatible Software Versions

The following software versions are recommended for the hierarchical flow:

- Voltus-Fi-XL 6.1.7 ISR or ICADV123 ISR latest version
- MMSIM 15.1.0 latest version
- EXT 16.1.1 latest version

Data Requirements for the Hierarchical EMIR Flow

To run the hierarchical EMIR flow, ensure that the following inputs are available:

- All inputs for generating PGVs for the blocks. For details, see Data Requirements and Flow for PGV Creation in the “Power-Grid View Generation” chapter. In addition, for the hierarchical flow, provide the following:

© 2020 All Rights Reserved.

  - LibGen command file for any additional commands needed for creating block PGVs.
  - Path to the LibGen binary. This is used only to specify a binary outside the IC hierarchy. By default, the software uses the binary available in the IC hierarchy.
- All inputs required for embedded extraction flow in Voltus-Fi. Ensure the following:
  - The Quantus hierarchy is available because the software calls the Quantus binary from this path.
  - The Pegasus, PVS, or Calibre® run is performed with the “lvs black box” option to generate the Pegasus, PVS, or Calibre database, which is the input database for generating the xDSPF file using Quantus.

    Note: For the hierarchical flow, Quantus only supports the input database `svdb` that is generated by Pegasus, PVS, and Calibre LVS.
  - The PGV binary database is available.
- Inputs required for top-level simulation.
  - Current/PWL file from the block simulation.
  - The input circuit file, which contains information about models/vectors and other parameters required for top-level simulation.

Note: Pegasus, the new physical verification system, will be available in the 18.2 physical verification release.

Types of Block-Level xDSPF – Detailed, Port, and Via Snapped

In the hierarchical flow, block-level xDSPF is generated from the PGVs.

The PGVs contain the complete RC grid for all the power nets and their node locations and current taps, which can be written in the xDSPF format. However, the complete RC grid may not be required for all the blocks, so you can choose to create the xDSPF for a block to include only a specific type of information. The following options are available:

- **Detailed xDSPF**: Includes the complete RC grid information for the blocks. The current taps are on the device locations of the original block. The xDSPF subcircuit name is `<BlockName>_Detailed`.
- **Port xDSPF**: Includes the port model information for all the interface nodes but does not include the RC grid information. The current taps are at the interface locations, and the xDSPF subcircuit name is `<BlockName>_Port`.

  Note: This xDSPF file is not recommended for hierarchical analysis because of low

© 2020 All Rights Reserved.

accuracy results.

- **ViaLayer/MetalLayer/<*LayerName*>**: Includes the RC grid information only for the layers above the specified layer. However, the capacitance and currents of all the layers below the specified layer will be distributed on the specified layer. The xDSPF subcircuit name is `<BlockName>_<LayerName>`.

The name of the layer above which the RC information should be included in the via layer xDSPF that is generated is specified in the GUI by using the *Stop Layer* field in the Create Power Grid Views form. In batch mode, it is provided by specifying the `-hier_flow_stop_layer layer_name` parameter of the `create_pgv` command.

For example, if `via1` is specified as the stop layer, then the power grids of all the layers below `via1` will be removed from the xDSPF file and their capacitance and the currents will be distributed either on `via1` or the layers above it.

Important

If the type of xDSPF file is not specified while performing the top-level simulation, then the software generates Detailed and Port xDSPF files. If the stop layer is specified, all three types of xDSPF files are generated by default.

PWL Current Files for Top-Level Simulation

For the top-level simulation in the hierarchical flow, block currents are specified using PWL files. These files are generated while running the block simulation.

Figure 15-3 Block-Level Simulation PWL Generation

© 2020
All Rights Reserved.

The PWL files include the following information:

- Tap device and terminal name
- Subckt port PWL current
- Tap device PWL current
- Average resistance and capacitance for tap device
- Time-varying always-on resistance (Ron) for power-gated designs

The PWL files are written using the following keyword in the EMIR configuration file:

```
eisopt save_tapi=<designName.tapi>
```

Note: The PWL file is only created in the Iterated mode of simulation. It is not supported in the direct mode of simulation. For details of the two modes, see Spectre APS/XPS EMIR Flow Overview in the “Data Preparation” chapter.

© 2020 All Rights Reserved.

Hierarchical EMIR Analysis Flow in GUI Mode

The hierarchical EMIR flow in Voltus-Fi-XL enables the hierarchical signoff for the Analog on Top (AoT) designs with a cockpit that is fully integrated within Virtuoso. Voltus-Fi-XL supports the hierarchical EMIR flow in both GUI and batch modes. The flow in the GUI mode comprises the following steps:

1. Opening the Hierarchical EMIR Forms
2. Creating the Block PGVs and xDSPF
3. Generating the Top-Level xDSPF File
4. Running the Top-Level Simulation

Opening the Hierarchical EMIR Forms

To start the hierarchical EMIR flow, you need to open the three related forms: Create Power Grid Views, Extract xDSPF, and Hierarchical Top Simulation. To open these forms, perform the following step.

- In the Voltus-Fi-XL interface, choose *Hierarchical EMIR.*

Creating the Block PGVs and xDSPF

As a first step in the hierarchical flow within Voltus-Fi, block-level PGVs and xDSPF files are generated. In the GUI, the Create Power Grid Views form of Voltus-Fi is enhanced to specify the options for the hierarchical flow.

In batch mode, the set_mmsim_pgv_nets command is used to specify the nets for which you want to create PGVs, and the create_pgv command is enhanced to specify the settings for the hierarchical flow. The GUI use model is detailed below.

In the *Hierarchical EMIR* menu, click the *Block Power Grid View Generation* submenu. The Create Power Grid Views form opens.

© 2020 All Rights Reserved.

Figure 15-4 The Create Power Grid Views Form

In this form, you can provide the information detailed below.

- Specify the *OA Libs*, which is the name of the open access (OA) database library.

© 2020 All Rights Reserved.

Note: If this option is specified, the *Tech Lef File/Macro Lef File* options do not need to be specified.

- Specify the *Macro Name,* which is the top cell name.
- Specify the *Library Name,* which is the name of the library. This option is used to create a library with a name different than that of the macro. By default, a library is created with the same name as that of the macro.
- Specify the *Extraction Tech File*, which provides the name of the technology file used for reading the technology data.
- Specify the *Tech Lef File*, which includes the library exchange format (LEF) information for the technology data.

  Note: The *Tech Lef File and OA Libs* options are mutually exclusive.
- Specify the *Macro Lef File, which includes the* LEF information for the design.

  Note:

  - In case the technology data and design data are in the same LEF file, use any of the two options, *Tech Lef File* or *Macro Lef File* to specify the filename.
  - The *Macro Lef File and OA Libs* options are mutually exclusive.
- Specify the *Lef Def Layer Map File*, which provides the mapping between LEF layer names and technology layer names.

  Note: If the lef def layer map file is not specified, the software automatically generates this file. For details, see Automatic Generation of Layer Map Files in the “Power-Grid View Generation” chapter.
- Specify the power-grid database or the *PGDB Layer Map File,* which provides the mapping between the xDSPF layer names that are written in the simulation database, and the technology layer names.

  Note:

  - If an xDSPF layer name is missing in this file, the PGVs will not contain data for that layer.
  - If the PGDB layer map file is not specified, the software automatically generates this file.

  For details, see Automatic Generation of Layer Map Files in the “Power-Grid View Generation” chapter.

© 2020 All Rights Reserved.

For more information about the LEF layer map file and PGDB layer map file formats, see LEF layer Map File and PGDB layer Map File in the "File Formats" chapter.

The *Tech Lef File*, the *Macro Lef File*, the *Lef Def Layer Map File*, and the *PGDB Layer Map File* can be created and edited using the edit buttons provided against each of these fields.

- Specify the *Simulation Directory*. For the hierarchical flow, the simulation directory may or may not contain the current (`*.ptiavg`) files.
- Specify the *Simulation Prefix,* which is the prefix used for creating an instance while performing the simulation. The prefix is specified along with the hierarchy separator. This input is optional.

  For example, you can specify "`XO.`" or "`IO.`". In these examples, `"XO"` or `"IO"` are prefixes `and "."` is the hierarchy separator. You can check the prefix used during simulation in your simulator input file.

- Specify the *UTI Ron File* that is generated by Spectre. This file contains multiple on-resistance (Ron) values – minimum, maximum, and average – for powergates. A sample file is shown below:

```
D-Term-Net   S-Term-Net   Ron_avg(Ohm) Ron_max(Ohm) Ron_min(Ohm) Power-Gate
outp       VDD         1268.34      1302.38      1230.52     MPM3
outp       VDD         1268.03      1302.06      1230.2      MPM3@39
outp       VDD         1268         1302.04      1230.18     MPM3@40
outp       VDD         1268.03      1302.07      1230.21     MPM3@41
```

  If this file is not specified, Voltus-Fi uses the `Ron` values from the Spectre simulation database (`*.emir#_bin` file).

- Specify the *Trigger File*, which is a text file that is used during a dynamic current simulation to run a detailed analysis for determining the distribution of current. For detailed information about the trigger file format and contents, see Format and Contents of Trigger File in the "File Formats" chapter.
- The *Hierarchical Flow xDSPF generation* option is selected by default. This option enables the hierarchical EMIR analysis.
- Specify the *Stop Layer* to specify the layer for creating a snapped via layer xDSPF. When the stop layer name is specified, the RC grid for all the layers below the specified layer will not be included in the xDSPF file that is generated for the block.
- Specify the *Sub Node Char* that will be used as the subnode identification. By default, `#` is specified.
- Specify the *Net Name* and the corresponding *Voltage*.

© 2020 All Rights Reserved.

- Specify the *State Directory/Results File*, which is the simulation result file. The naming convention of the simulation result file is, `*.emir#_bin`.
- The table below is populated with the information about the specified *Net Name*, *Voltage,* and *EMIR Results DB File*.
  - Click *Add* to add a net to the list.
  - Click *Remove* to remove a net from the list.
- In the *Output* group box, specify the *Results Directory*, which is the output directory for the PGVs. By default, the PGVs are stored in the work directory.

Generating the Top-Level xDSPF File

Once the block PGVs and xDSPF files are generated, the next step involves performing extraction using the embedded Quantus flow of Voltus-Fi and generating the top-level xDSPF file.

In the GUI, the Extract xDSPF form of Voltus-Fi is enhanced to specify the options for the hierarchical flow.

In batch mode, the extract_xdspf command is enhanced to include the settings for the hierarchical flow. The GUI use model is detailed below.

In the *Hierarchical EMIR* menu, select *Extract Top DSPF*. The Extract xDSPF form opens.

© 2020 All Rights Reserved.

Figure 15-5 The Extract xDSPF Form

© 2020
All Rights Reserved.

In this form, you can provide the information detailed below.

In the *Input* section:

- For *Data Type*, select either *PVS/Pegasus* or *Calibre*. This specifies whether the input data will come from *Pegasus* or *Calibre*. By default, *PVS/Pegasus* is selected.

  Note: In case PVS was used to generate data, Pegasus should be selected.

- Specify the *Data Directory*, which is the svdb directory created with lvs black box option, that provides the path to the directory in which the input data is stored.
- Specify the *Run Name*, which is the name of the Pegasus or Calibre run.
- Specify the *Post Processing Script*, which is the name of the post-processing script file.

In the *Process* section:

- Specify the *Foundry / Process Node*. The following options are available:
  - *N5/N3:* for N5 and N3 process nodes
  - *N7:* for N7 process nodes
  - *N10-PG*: for N10 process nodes with PG rules
  - *N10*: for N10 process nodes
  - *N20/N16/N12*: for N20, N16, and N12 process nodes
  - *N14*: for N14 process nodes
  - *Others > 20*: for process nodes greater than N20
- Select *Signal EM* to specify that signal nets should be included in the xDSPF file. When *Signal EM* is selected, the following command is added to the Quantus command file or the common command language (CCL) file, to perform decoupled RC extraction `(rc_decoupled)` for all nets.

  Note: In `rc_decoupled` extraction, an RC network for the specified net(s) with both parasitic resistance and capacitance decoupled to ground is extracted.

  ```
  extract \
     -selection "all" \
     -type "rc_decoupled"
  ```

  When *Signal EM* is not selected, you need to provide the names of the power or ground nets on which `rc_decoupled` extraction will be performed. For all other nets, `c_only_decoupled` extraction is performed.

© 2020 All Rights Reserved.

Note: In `c_only_decoupled` extraction, only parasitic capacitance between the specified nets and adjacent nets decoupled to the specified cap ground is extracted.

For example, when the following command is specified, `c_only_decoupled` extraction is performed on all nets:

```
extract \
   -selection "all" \
   -type "c_only_decoupled"
```

When the following command is specified, `rc_decoupled` extraction is performed on nets specified in the `vfiQrc_nets` file. For all other nets, `c_only_decoupled` extraction is performed.

```
extract \
   -selection "nets_file ./vfiQrc_nets"
   -type "rc_decoupled"
```

- Specify *PG Net(s)* or the power and ground nets that should be included in the xDSPF file. This option is specified only when the *Signal EM* option is not selected. If the *Signal EM* option is not selected and *PG Nets* are not provided, a pop-up window opens asking for the list of PG Nets.
- Specify the *Nets File* that contains the names of the nets to be included in extraction. When the *Nets File* option is checked, the *Signal EM* and *PG Net(s)* options will be ignored.
- The *Hierarchical EMIR flow* check box is selected by default. This option enables hierarchical EMIR analysis.
- Specify the *PGV Library Name*, which is the block PGV library name that is generated in the first step and used for the block-level extraction performed for the hierarchical flow.

  Note: If there are multiple block PGVs to be used, create a merged library and specify the name of that library in the *PGV Library Name* field. This is required to be done only in the GUI mode. In batch mode, the merged library is created automatically.
- In the *Extraction Type* cyclic field, specify one of the following:
  - *RC*: specifies that both parasitic resistors and capacitors will be extracted
  - *R only*: specifies that only parasitic resistors will be extracted

  Note: When *Extraction Type* selected is *R only*, the simulation time is short. While selecting the extraction type, consider the type of simulation being run. By default, *RC* extraction type is selected.
- In the *Cap Coupling Model* cyclic field, select one of the following:
  - *De-coupled*: specifies that decoupled capacitance extraction will be performed

© 2020 All Rights Reserved.

- *Coupled*: specifies that coupled capacitance extraction will be performed

  By default, the *De-coupled* mode is selected.

Note: While specifying the coupling mode, ensure that there is no `extract` command defined in the additional CCL file. The `extract` command in the additional CCL file will override the above options. For more information about the above options, see the *Quantus Extraction Users Manual*.

- Select *Parasitic RES Color* to enable color-aware EM analysis.
- Select *Compression* to generate the xDSPF in a compressed format.
- Select *Reduce I-Cards* to enable the `reduce_i_cards` option in the CCL file that is used for generating the xDSPF file. If a net is capacitance only, it is qualified for reduced I card or instance card treatment. This means the I cards and the corresponding dummy resistors are not printed in the results.
- Select *Comment Sub Ckt* to comment out the top subcircuit statements from the xDSPF file.
- Select *Sub Conductor Model* to specify the parasitic resistor models for the subconductor layers that should be included during extraction. This is used to support different EM rules for poly layers during EM analysis.
- Select *Filter RES Center* to include sub-nodes after eliminating resistors with small resistance values. When this option is selected, the survivor sub-nodes, among a group of sub-nodes that are formed by the filtered resistors, are chosen based on their proximity to the geometric center of the group. This field is specific to process nodes 5nm and below.
- Select *Filter RES By Layer* to specify the LVS layer name for which the minimum resistance value for the extracted parasitic resistors is to be applied. If the layer name is not specified, the default value, `RH_TN_6 0`, is used. This field is specific to process nodes 5nm and below.
- Specify the *Ref Node* or the reference node on which the extraction is to be performed. By default, `0` is specified.
- Specify the *Sub-Node Char* that will be used as subnode identification. By default, `#` is specified.
- Specify the *Fracture Via Count* that will be used to divide pseudo vias into segments during extraction.
- Specify the *Via Array Spacing,* which is the distance, a floating point micron value that will be used as the maximum distance criteria for grouping vias within the same array. This means that if the distance between vias is less than the specified maximum

© 2020 All Rights Reserved.

distance, they are grouped within the same array. The default value of the via array spacing is 0.35 times the via size for N10, N16, N12, and N20 processes. For other processes, the default is *auto*, in which case the software sets the via spacing automatically.

Use this option to specify a value other than the default value of 0.35.

- Specify the *Via Array Count* number, without the per via layer count option, which is the number of vias per side allowed in an array. The default value is 4. For example, if the via array count is set to 2, the number of vias allowed per array side will be 2.

  Note: For more information about the above, see the *Quantus Extraction Users Manual*.

In the *Technology* section:

- From the list, either specify the *Technology Library File* that provides the path of the technology file to be used for reading the technology data, or the name of the *Technology Directory* that contains the technology file.
- Specify the *Technology Name*, which is the technology filename to be used for extraction. This field is optional if the technology directory has been specified.
- Specify the *Corner*, which is the name of the technology corner to be picked from the library.
- Specify the *Temperature* at which the extraction will be performed. By default, a temperature of 25 degree Celsius is used. The temperature specified for extraction is optional. The options, temperature and `temp_coeff`, specified in the CCL file are mutually exclusive. The following scenarios hold true.
  - If you specify a temperature for extraction in the Extract xDSPF form, or by using the `-temperature` parameter of the `extract_xdspf` command in batch mode, the following option in the CCL file will be set to `false`:

    ```
    -include_parasitic_res_temp_coeff "false" \
    ```

    The software displays the following message:

    ```
    User specified temperature will be used for extraction. The option -
    include_parasitic_res_temp_coeff will be set to false.
    ```

  - If the temperature is not specified for extraction, the following option in the CCL file will be set to `true`:

    ```
    -include_parasitic_res_temp_coeff "true" \
    ```

    The software displays the following message:

    ```
    No user specified temperature value so the option -
    include_parasitic_res_temp_coeff will be set to true.
    ```


© 2020 All Rights Reserved.

- Specify the number of CPUs to be used during the Quantus extraction run in the *No of CPUs* field.
- Specify the *SubCkt Pin Order File* to provide the order of pins to be used during extraction. The pin order is specified in the xDSPF file using the `.SUBCKT` and `.ENDS` statements. By default, for the hierarchical flow, the software uses the pin order file for blocks from the PGV directory. You can use this option to provide a file with a different pin order that will override the order specified using the default `.SUBCKT` statement in the xDSPF file.

  Below are examples of pin orders in the xDSPF file before and after specifying the pin order file.

  Without the sub-circuit pin order file

  ```
  .SUBCKT top In_001 VDD Out_001  VSS In_002 Out_002 In_003 Out_004 ...
  .ENDS
  ```

  With the sub-circuit pin order file

  ```
  .SUBCKT top VDD VSS In_001 Out_001 In_002 Out_002 In_003 Out_004 ...
  .ENDS
  ```

- Specify the *Customized Extract Commands File* to specify the file with user-customized extract commands. When this file is specified, the Voltus-Fi-generated CCL will not have any extract commands, and the commands from the customized extract commands file will be added in the Voltus-Fi-generated CCL.

  A sample customized extract commands file is as follows:

  ```
  extract \
     -selection "all" \
     -type "rc_coupled"

  extract \
     -selection nets_file "file.txt" \
     -type "none"
  ```

  Note: You can create a `customextract` file by using either single or multiple extract settings. Ensure that the commands are in the Quantus CCL file syntax because this file is added as is in the Voltus-Fi-generated CCL.

- Select *Add Extract Command* if you want Voltus-Fi to add the extract commands to the CCL file.

  By default, when a `customextract` file is specified, all commands, including the extract commands, from that file are added to the Voltus-Fi-generated CCL file. However, when you select *Add Extract Command*, Voltus-Fi adds the extract commands to the CCL

© 2020 All Rights Reserved.

file, and all other commands specified in the `customextract` file are added to the CCL file.

- Specify the *Parasitic blocking device cells file* to specify the file containing parasitic blocking device cell names, if any. Once a cell is declared as a blocking cell, all the parasitic RCs inside the cell are not extracted. If the cell content overlaps top cell routing, then the top cell routing's RCs in the overlapping regions are also not extracted.

In the *Output* section:

- Specify the *Directory*, which is the output directory in which the xDSPF file will be saved. If this file is not specified, a pop-up window opens asking for the name of the directory.
- Specify the name of the *xDSPF* file that will be generated. If the filename is not specified, the software generates a file with the name, `<runName>.dspf`, by default.
- Select *Autorun Spfchecker* to automatically run the SPF Checker utility after the extraction run. SPF Checker checks the syntax of the DSPF/SPEF followed by opening the output log. It also compares the terminal names in the DSPF and the device models and populates the summary section with the recommended mapping statements or the `spf aliasterm` statements.

  This check box is selected by default.

  Note:

  - To use this feature, the SPF Checker utility must be located at the MMSIM/Spectre installation directory.
  - SPF Checker works only when the top-level Quantus run is successful.
- Click *Generate CCL* to create the CCL or the Quantus command file that will be used to run Quantus. By default, `vfiQuantus-<cellname>.ccl` is created and the file opens in the Voltus-Fi console. You can view the file and modify it, if required.
- Click *Generate and Run* to create the CCL file and run Quantus to create the xDSPF file in a single step. This is useful when you do not want to view the CCL file before running Quantus.

  Note:

  - When you click *Generate and Run*, Voltus-Fi creates a file, `.vfiExtractDSPFGuiLast` to save the form settings. This file gets loaded automatically when the next session starts. If you do not want to load these settings, remove this file before starting the session. This file is created along with `cds.lib`.
  - The values specified in the `.cdsinit` file override the values of the `.vfiExtractDSPFGuiLast` file of the previous session.

© 2020 All Rights Reserved.

- Click *Run* to select the CCL file to use for running Quantus. This is useful if you have modified the auto-generated CCL file in the previous step. The log is saved both in the `vfiQuantus.log` and on the Voltus-Fi console.

  The `vfiQuantus-<cellname>.log` file stores all the CCL commands and the info, warning, and error messages issued by Voltus-Fi during the extraction run. It is saved in the specified output directory.

  With every successive extraction run, the name of the log file is updated to `vfiQuantus-<cellname>.log.(n-1)`. Here, n-1 is the run number. Therefore, the greater the numeric value in the filename, the more recent or latest that file is.

  For example, after three extraction runs, the following log files are created:

  - `vfiQuantus-<cellname>.log` (log file for the first run)
  - `vfiQuantus-<cellname>.log.1` (log file for the second run)
  - `vfiQuantus-<cellname>.log.2` (log file for the latest run)

  For information about the format of the Voltus-Fi-generated CCL file and its example, see Common Command Language (CCL) File in the “File Formats” chapter.

- Click *Reset* if you want to revert to the default values.

Running the Top-Level Simulation

Once the block-level PGVs are generated and top-level xDSPF is created, the next step is performing the top-level simulation. In this step, the top-level xDSPF generated by Quantus is processed along with the block-level xDSPF generated by VFI-LibGen to perform simulation.

In the GUI, the Hierarchical Top Simulation form is provided to specify the options for the hierarchical flow.

In batch mode, two commands, set_block_data and run_top_simulation are provided to specify the settings for the hierarchical flow. The GUI use model is detailed below.

In the *Hierarchical EMIR* menu, click the *Top Simulation* submenu. The Hierarchical Top Simulation form opens.

© 2020 All Rights Reserved.

Figure 15-6 The Hierarchical Top Simulation Form

In the *Block* group box, specify the following:

- In the *xDSPF Type* box, specify the type of xDSPF to be used from the PGV directory. You can choose from the following options:
  - *Detailed*: to select the detailed block xDSPF file

© 2020 All Rights Reserved.

- *Port*: to select the port block xDSPF file
- *Via Snapped*: to select the via layer xDSPF generated using via layer snapping. For details, see Types of Block-Level xDSPF – Detailed, Port, and Via Snapped

- Specify the *PGV dir for the block xDSPF file*.
- In the *Instance(s)* field, specify the instance name of the block. For multiple instances of the same block that use the same current file, specify the instances separated by a space.
- In the *Current Type* list box, specify whether the current analysis will be *Static* or *Dynamic*. By default, *Dynamic* is selected.
- Specify the path to the *Current File*. This is a PWL file. For details of the file contents and how to generate the PWL files, see PWL Current Files for Top-Level Simulation.
- Click *Add Block* to add the block data specified above to the table.
- Click *Add Block Instance* to add another instance to the already added block with a different current file.
- Click *Remove* to remove any block data from the table.

In the *Top* group box, specify the following:

- Specify the top-level *xDSPF File* generated in the previous step.
- Specify the input *Ckt File* with models/vectors and other parameters for top-level simulation.
- Select the *Bus Bit Char* from the options, `<>`, `[]`, and `{}` available in the cyclic field. The default character is `<>`.

In the *EMIR Config file* group box, specify the following:

- Specify an existing EMIR configuration file or generate an EMIR config file. To specify an existing file, click the Browse button and select the file. For generating the EMIR config file, specify the following options:
  - The name of the *EM Techfile*
  - The *Instance Hierarchy*

© 2020 All Rights Reserved.

Figure 15-7 Creating an EMIR Config File

- Click *Create New* to generate a default EMIR config file, `emir_vfi.conf`. A sample file is shown below.

- Specify the *Results Directory* to store the results of the top-level simulation. The default directory is `vfitop`.
- Click *Run* to run the top-level simulation.

© 2020
All Rights Reserved.

The Hierarchical EMIR Analysis Batch Mode Flow

The following commands are provided for performing the Hierarchical EMIR analysis flow.

- The following commands are used to create block PGVs in batch mode:
  - set_mmsim_pgv_nets
  - create_pgv
- The following command is used to generate top-level xDSPF in batch mode:
  - extract_xdspf
- The following commands are used to run top-level simulation in batch mode:
  - set_block_data
  - run_top_simulation

© 2020 All Rights Reserved.

set_mmsim_pgv_nets

```
set_mmsim_pgv_nets
     -net net_name
     -voltage voltage_value
     -voltage_source_nets voltage_source_only_nets
     -results_file result_file_name
     [-prefix prefix_name]
     [-feedthru_nets {net1 net2 net#}]
```

Specifies the nets for which you want to create PGVs in the Spectre flow. You can use this command to specify multiple PGV nets at a time.

Example:

```
set_mmsim_pgv_nets \
-net { AVDD AVSS Ibias inm inp outm outp } \
-voltage { 1.3 0 0 0 0 0 0 } \
-results_file psf/input.emir0_bin \
-prefix I0
```

For detailed information about the parameters of this command, see set_mmsim_pgv_nets in the “Batch Mode Execution” chapter.

© 2020 All Rights Reserved.

create_pgv

```
create_pgv
     -macro_name macro_name
     [-lib_name name_of_library]
     [-hier_flow {true | false}]
     [-hier_flow_stop_layer layer_name]
     [-disable_signal_pgdb {true | false}]
     [-stop_on_missing_lef_pins {true | false}]
     [-libgen_cmd_file cmd_file_name]
     [-libgen_path libgen_binary_path]
     -tech_file tech_file_name
     [-tech_lef_file tech_lef_file_name]
     -type {static | dynamic}
     [-macro_lef_file macro_lef_file_name]
     -sim_dir simulation_dir
     [-lef_layer_map lef_layer_map_file_name]
     [-oa_lib list_of_OA_libraries]
     -pgdb_layer_map pgdb_layermap_file_name
     [-pgdb_version version_number]
     [-trigger_file trigger_file_name]
     [-output_directory directory]
     [-uti_Ron_file Ron_report_file_dumped_by_Spectre]
     [-create_no_device_nets true | false]
     [-powergate_state_flow true | false]
     [-use_pwl_files true | false]
     [-map_connectivity_layers true | false]
     [-units value]
```

Creates PGVs. The PGVs contain information about the power ports of a transistor or block, the internal power grid, and the tap current distribution within a transistor or block.

Note: The options in blue in the above syntax are relevant only for the hierarchical EMIR analysis flow and are detailed below.

Parameters

`-disable_signal_pgdb true | false`	
	Checks the nets or pins with port shapes in the LEF file and confirms that the input results database (`emir_bin`) contains these nets data. When set to `false`, generates an error message if the results database does not contain the data of the LEF pins or nets. Default: `True`

© 2020 All Rights Reserved.

`-hier_flow true | false`	
	When set to `true`, enables the hierarchical analysis flow. **Note:** This parameter is optional for the regular PGV generation flow but it is required for the hierarchical flow. It is used to enable via snapping to a higher level in the PGV.
`-hier_flow_stop_layer` *`layer_name`*	
	Specifies the name of the stop layer for generating the via layer xDSPF file. For example, if layer, via1 is specified, then the power grids of all the layers below via1 will not be included in the xDSPF file, and their capacitance and currents will be distributed either on via1 or layers above it. **Note:** This parameter is optional for the regular PGV generation flow but it is required for the hierarchical flow. It is used to enable via snapping to a higher level in the PGV.
`-sim_dir` *`simulation_dir`*	
	Specifies the simulation directory for the current data. This is a required parameter. **Note:** For the hierarchical flow, the simulation directory may or may not contain the current (`*.ptiavg`) files.
`-stop_on_missing_lef_pins true | false`	
	When set to `false`, generates a warning message (instead of an error message) if the `emir_bin` database does not contain data about LEF pins or nets and continues to generate the PGVs. Default: `True`

Example

The following command is used to create power-grid views for hierarchical EMIR analysis:

```
create_pgv  \
-macro_name DiffOpAmp \
-tech_file typical_unified/qrcTechFile \
-tech_lef_file PGV_gen_setup/LEF/gsclib045_tech.lef \
-type dynamic \
-macro_lef_file ./DiffOpAmp_multiAVDD.lef \
-sim_dir ./ \
-lef_layer_map PGV_gen_setup/lefdef.map \
-pgdb_layer_map PGV_gen_setup/pgdb.map \
-output_directory vfi_pgv_hier \
```


© 2020 All Rights Reserved.

```
-hier_flow true \
-hier_flow_stop_layer Via1 \
-libgen_cmd_file mylibgen.cmd \
-libgen_path ../voltus_libgen
```


© 2020 All Rights Reserved.

extract_xdspf

```
extract_xdspf
     -signalEM {true | false}
     [-PGNets list_of_pg_nets]
     -process {n7 | n10pg | n10 | n20 | n16 | n12 | others}
     -techDir tech_dir_name | -techLib tech_lib_file_name
     [-techName technology_name]
     [-techCorner technology_corner_name]
     [-temperature value]
     -inputType {pegasus | pvs | calibre}
     -inputDir input_dir_name
     -RunName input_run_name
     -outputDir output_dir_name
     [-hier_flow {true | false}]
     [-hier_flow_pgv {1.cl 2.cl}]
     [-outputDspf output_dspf_filename]
     [-output_xy_generic {true | false}]
     [-coloring {true | false}]
     [-subcktPinOrder pin_order_file]
     [-multiCpu number_of_cpus]
     [-customExtract extract_Commands_File]
     [-groundNet ground_net_name]
     [-blockingCellFile parasitic_blocking_device_cell_names_file]
     [-subNodeChar sub_node_char]
     [-compression {true | false}]
     [-reduction {true | false}]
     [-commentSubCkt {true | false}]
     [-subConductorModel {true | false}]
     [-fractureViaCnt via_count_number]
     [-arrayViaSpacing {auto | value}]
     [-lic_queue timeout_in_seconds]
     [-generate_ccl_only {true | false}]
     [-user_ccl_file ccl_file_name]
     [-bus_bit_char string]
     [-tech_cmd_file filename]
     [-tech_layer_setup_file filename]
     [-icell_map_file filename]
     [-keep_pin_shorting_res {true | false}]
     [-keep_pin_shorting_res_by_layer {"layer1" "layer2" .... }]
```

Generates the xDSPF file using Quantus from within Voltus-Fi.

Note: Before running this command, ensure that the Quantus hierarchy is available because Voltus-Fi calls the Quantus binary from this path and also ensure that the Pegasus, PVS, or Calibre® run is performed successfully.

Note: The options in blue in the above syntax are specified only for the hierarchical EMIR analysis flow and are detailed below. For the description of the other options of this command, see extract_xdspf in the "Batch Mode Execution" chapter.

© 2020 All Rights Reserved.

Parameters

`-hier_flow true | false`	When set to `true`, enables the hierarchical EMIR analysis flow. **Note:** This parameter is optional for the regular extraction flow but it is required for the hierarchical flow.
`-hier_flow_pgv 1.cl 2.cl`	Specifies the names of the PGV files (`.cl`). You can specify multiple filenames separated by a space. Voltus-Fi creates a merged PGV. **Note:** This parameter is optional for the regular extraction flow but it is required for the hierarchical flow.

Examples

- The following command is used to run the hierarchical flow using the `DiffOpAmp.cl` PGV file:

```
extract_xdspf
-signalEM true \
-process n20 \
-techLib ../pvtech.lib \
-techName gpdk045  \
-techCorner "rcx_typical" \
-inputType pegasus \
-inputDir ./svdb \
-RunName DiffOpAmp_top_2subblock_VSSport \
-outputdir vfi_quantus_top \
-temperature 25 \
-hier_flow true \
-hier_flow_pgv DiffOpAmp.cl \
-subcktPinOrder pin_order_file \
-groundNet  AVSS \
-reduction true
```


© 2020 All Rights Reserved.

set_block_data

```
set_block_data
     -dspf block_dspf_filename
     -instance {XI1 XI2 ...}
     [-current_type {static | dynamic}]
     -current_file current_filename
```

Specifies the block data information to be used in the top-level simulation.

Parameters

`-dspf` *`block_dspf_filename`*	
	Specifies the name of the block xDSPF file to be used. This is a required parameter.
`instance {XI1 XI2 ...}`	Specifies the instance names of the child cells. For multiple instances of the same block that use the same current file, specify the names separated by a space. This is a required parameter.
`-current_type {static | dynamic}`	
	Specifies whether the current analysis type will be static or dynamic. This is an optional parameter. Default: `dynamic`
`-current_file` *`current_filename`*	
	Specifies the current file to be used. This is a PWL file. for details of the file and how to create it, see PWL Current Files for Top-Level Simulation. This is a required parameter.

Examples

- Use the following command to specify the same PWL file for all instances:

```
set_block_data
-dspf DiffOpAmp_detailed.dspf
-instance {XI1 XI2}
-current_type dynamic
-current_file DiffOpAmp.pwl
```


© 2020 All Rights Reserved.

- Use the following command to specify a different pwl file for each instance:

```
set_block_data
-dspf DiffOpAmp_detailed.dspf
-instance {XI1 XI2}
-current_type dynamic
-current_file {DiffOpAmp.pwl DiffOpAmp.pwl}
```

- Use the following command to specify the static current for a block:

```
set_block_data
-dspf DiffOpAmp_detailed.dspf
-instance {XI1 XI2}
-current_type static
-current_file block_i.txt
```

- Use the following command to use the via snapped xDSPF:

```
set_block_data
-dspf DiffOpAmp_vialayer.dspf
-instance {XI1 XI2}
-current_type dynamic
-current_file DiffOpAmp.pwl
```


© 2020 All Rights Reserved.

run_top_simulation

```
run_top_simulation
     -cktfile input_ckt_file
     [-dspf top_dspf_file]
     [-bus_bit_char bus_bit_char]
     [-emir_conf_file emir_config_file]
     [-generate_cmd_only {true | false}]
     [-output_directory output dir]
     [-overwrite_cmd_files {true | false}]
     [-sim_switches "sim_cmd_line_options"]
     [-use_config_file_flow {true | false}]
```

Specifies the information to be used for performing the top-level simulation.

Parameters

`-bus_bit_char` *`bus_bit_char`*	
	Specifies the bus delimiter character. The following options are available: `<>`, `[]`, `{}`. This is an optional parameter. Default: `<>`
`-cktfile` *`input_ckt_file`*	Specifies the input circuit file for the top simulation. This file contains information about models/vectors and other parameters required for the top-level simulation. This is a required parameter.
`-dspf` *`top_dspf_file`*	Specifies the top-level xDSPF file to be used for the top-level simulation. This is an optional parameter if the `extract_xdspf` command is specified. In that case, Quantus is run and the top-level xDSPF file that is generated is used by the tool for the top-level simulation.
`-emir_conf_file` *`emir_config_file`*	
	Specifies the top-level EMIR configuration file. This is an optional parameter. If the file is not specified, the tool generates a default `emir.conf` file for all nets.
`-generate_cmd_only {true | false}`	

© 2020 All Rights Reserved.

	When set to `true`, the Spectre run is skipped. The input command files, `vfi_input.scs` and `vfi_runsimulation`, are created. You can make changes in these files and then run Spectre. This is an optional parameter. Default: `false`
`-output_directory` *`output dir`*	Specifies the name of the directory in which the results of the top-level simulation will be saved. This is an optional parameter. By default, the tool uses the `topSimulation` directory.
`-overwrite_cmd_files {true | false}`	
	When set to `false`, the tool generates new input command files and does not overwrite the input command files generated in the previous run. For example, if `vfi_input.scs` and `vfi_runsimulation` files were generated in the previous run, and if this option is set to `false` in the new run, then new files, `vfi_input1.scs` and `vfi_runsimulation1`, are created. By default, this option is set to `true`, which means the input command files generated in the previous run are overwritten. This is an optional parameter. Default: `true`
`-sim_switches "`*`sim_cmd_line_options`*`"`	
	This option is used only to add additional Spectre command-line options for running the top-level simulation. The names of the switches are specified within quotes, as shown in the example below. This is an optional parameter. Default: none Example: `-sim_switches "+top +mt=8 +ms +fastdc"`
`-use_config_file_flow {true | false}`	

© 2020 All Rights Reserved.

	Specifies whether to store `tapi` commands in the `vfi_emir.conf` file instead of the `vfi_input.scs` file. Default: `false`

Example

Use the following command to run the top-level simulation using the top-level xDSPF file, `VSSport.dspf`:

```
run_top_simulation
     -dspf DiffOpAmp_top_2subblock_VSSport.dspf
     -cktfile input.ckt
     -bus_bit_char <>
     -emir_conf_file emir.conf
     -output_directory topsimvfi
```


© 2020 All Rights Reserved.

The Hierarchical EMIR Analysis Batch Mode Automation Flow

The hierarchical EMIR analysis flow in Voltus-Fi detailed in the above document is automated for the convenience of the user. The subsequent sections provide details of batch mode automation flow.

- Block PGV Generation
- Block PGV Inputs for Top-Level Extraction and Simulation
- The Top-Level Extraction and Simulation Run
- Multi-Mode Support

Block PGV Generation

In this step, block PGVs are generated using either batch commands or the GUI. Once a block PGV is created, the PGV run directory stores all the data needed for both the top-level Quantus and the top-level simulation runs.

In addition to the `.cl` files, the DSPF files and port_order files, and the PWL files from block simulation, are copied in the PGV run directory.

The PWL or current file for a block PGV is picked automatically from parsing the block `emir.conf` file, which has the name of the PWL created using the following option:

```
eisopt save_tapi=<pwl file name>
```

The PGV run directory includes the following data:

- The `cell.cl` file, which is the PGV binary database
- Block cell DSPF files for all modes (detailed, port, and snapped).
- The port_order file
- The block current/PWL file

Block PGV Inputs for Top-Level Extraction and Simulation

Once all the block PGVs are created, create a list of all the PGV directories and provide that as input for next step. This is done in a simple text file that contains the paths for each block PGV.

© 2020 All Rights Reserved.

The `set_pgv` command is used to provide all the details of the block-level data to be used for the top-level extraction and simulation runs.

set_pgv

```
set_pgv
     -module block_name
     -pgv_dir pgv_run_dir
     [-dspf dspf_file]
     [-delay value]
     [-repeat value]
     [-instances instance_list]
     [-view {snapped | detailed | port}]
     [-pwl {pwl_file1 pwl_file2...}]
```

Specifies the information to be used from the block PGV and xDSPF generation step for the top-level Quantus and simulation runs.

Note: This command is only used in batch mode automation flow for hierarchical EMIR analysis.

Parameters

`-delay value`	Specifies the value by which to delay the PWL file content. This value can be either positive or negative. This is an optional parameter. The default delay value is 0. Example: `set_pgv -delay 5n`
`-dspf dspf_file`	Specifies the block-level xDSPF file to be used for top-level extraction and simulation. This is an optional parameter. By default the tool uses the xDSPF file stored in the PGV run directory.
`-instances instance_list`	Specifies the list of instances for which you want to specify the PWL file. This is an optional parameter. By default, all the instances of the module use the PWL file stored in PGV run directory.
`-module block_name`	Specifies the name of the block. This is a required parameter.

© 2020 All Rights Reserved.

`-pgv_dir` *`pgv_run_dir`*	Specifies the name of the PGV run directory that contains the `.cl` file and the xDSPF files. This is a required parameter.
`-pwl {`*`pwl_file1 pwl_file2....`*`}`	
	Specifies the PWL files to be used for top-level simulation. You can specify a single PWL file to be used for all instances or for the instances specified using the `-instances` parameter. You can also specify multiple PWL files for different instances of a block. For details, see Multi-Mode Support. This is an optional parameter. By default, the software uses the PWL file stored in the PGV run directory.
`-repeat` *`value`*	Specifies the number of times the PWL content should be repeated. This value is always positive. This is an optional parameter. The default value of this parameter is 1, which means the PWL content is used only once.
`-view {snapped | detailed | port}`	
	Specifies the type of block-level xDSPF file to be used for top-level simulation. The following options are available: ■ `snapped`: a via layer xDSPF generated using via layer snapping ■ `detailed`: a detailed xDSPF file with complete RC grid information for the blocks ■ `port`: a port xDSPF file, which only includes the port order information for all interface nodes and does not include any RC grid information This is an optional parameter. *Default*: `detailed`

Examples

- The following command picks up the detailed DSPF and PWL from the `pgv_dir`, `vfi_pgv_hier` and applies it to all the instances.

```
set_pgv -module DiffOpAmp -pgv_dir ./blockPGV/vfi_pgv_hier
```


© 2020 All Rights Reserved.

- The following command picks up the detailed DSPF and PWL from the `pgv_dir`, `vfi_pgv_hier`, applies it to all the instances and delays the PWL content by `5n`.

```
set_pgv -module DiffOpAmp -pgv_dir ./blockPGV/vfi_pgv_hier -delay 5n
```

- The following command picks up the via layer snapped DSPF and PWL from the `pgv_dir`, `vfi_pgv_hier.via` and applies it to all the instances.

```
set_pgv -module DiffOpAmp -pgv_dir ./blockPGV/vfi_pgv_hier.via -view snapped
```

- The following command specifies the PWL, `DiffOpAmp.pwl`, for instance, `X1`. For all other instances for the same block, the PWL stored in pgv_dir, `vfi_pgv_hier.via`, is used.

```
set_pgv -module DiffOpAmp -pgv_dir ./blockPGV/vfi_pgv_hier.via -instances {XI1
} -pwl {blockSim/psf/input.emirtap/DiffOpAmp.pwl}
```

- The following commands provide the block-level data for different blocks, block 1 and block 2:

```
set_pgv -module block1 -pgv_dir ./block1PGV/vfi_pgv_hier1
set_pgv -module block2 -pgv_dir ./block2PGV/vfi_pgv_hier2
```

The Top-Level Extraction and Simulation Run

The top-level extraction run and the simulation run is a single-run process that takes the following as input:

- A list of block PGVs specified using the `set_pgv` command as detailed in the previous section.
- The Voltus-Fi extract_xdspf command inputs such as Pegasus/PVS/Calibre directory and technology files and so on. This is similar to block embedded Quantus flow with no additional input.
- Simulation inputs specified using the run_top_simulation command. These inputs include the input circuit file containing models, input vectors, and so on for the top-level simulation run.

Voltus-Fi uses the PGV list to create a merged `.cl` file for the top-level Quantus run. Once the top-level Quantus run is successful, the xDSPF is parsed to map the block instances with the xDSPF and PWL files.

While running the above, the tool uses the automation commands to create the following internal command files for extraction and simulation.

- `Hier_Flow_Automated_Top_Extraction.cmd`
- `Hier_Flow_Automated_Top_Simulation.cmd`

© 2020 All Rights Reserved.

These command files can be used for debugging, to verify that data is picked correctly. If needed, the command files can be used to run a specific step.

© 2020 All Rights Reserved.

Multi-Mode Support

The multi-mode support means that the hierarchical flow supports the use of different PWL files for different instances of a block.

Use the `-pwl` parameter of the `set_pgv` command to override the default PWL file for any instance with a different PWL file. This feature eliminates the need to specify or keep data for multiple modes in the PGV because the PGV and block xDSPF do not have any dependency on the current data stored in the PWL.

Example

```
set_pgv
-module DiffOpAmp
-pgv_dir ./blockPGV/vfi_pgv_hier.via
-instances {XI1 XI2}
-pwl {blockSimMode1/psf/input.emirtap/DiffOpAmp.pwl
blockSimMode2/psf/input.emirtap/DiffOpAmp.pwl}
```

A Sample Script for Automated Batch Mode Flow

Following are sample scripts for three different scenarios of the automated hierarchical EMIR analysis flow.

Sample 1: Using a detailed xDSPF file

In the script sample below, a `vpsbatch_automate.cmd` is run and the block-level PGV data is picked from the `vfi_pgv_hier` directory.

```
vpsbatch -cmd vpsbatch_automate.cmd
```

Where,

the `vpsbatch_automate.cmd` includes the following information:

```
include blockpgv.txt

extract_xdspf -signalEM true \
   -process n20 \
   -techLib ./pvtech.lib -techName gpdk045  -techCorner "rcx_typical" \
   -inputType pegasus -inputDir topQRC/svdb -RunName
DiffOpAmp_top_2subblock_VSSport \
   -temperature 25 \
   -groundNet  AVSS

run_top_simulation  -cktfile topSim/input.ckt  -emir_conf_file topSim/emir.conf
```

The `blockpgv.txt` file contains the following information:

```
set_pgv -module DiffOpAmp -pgv_dir ./blockPGV/vfi_pgv_hier
```


© 2020 All Rights Reserved.

Sample 2: Using a snapped xDSPF file

In the script sample below, a `vpsbatch_automate_vialayer.cmd` is run and the block-level PGV data is picked from the `vfi_pgv_hier.via` directory.

```
vpsbatch -cmd vpsbatch_automate_vialayer.cmd
```

Where,

the `vpsbatch_automate_vialayer.cmd` includes the following information:

```
include blockpgv_vialayer.txt
extract_xdspf -signalEM true \
   -process n20 \
   -techLib ./pvtech.lib -techName gpdk045  -techCorner "rcx_typical"  \
   -inputType pegasus -inputDir topQRC/svdb -RunName
DiffOpAmp_top_2subblock_VSSport \
   -temperature 25 \
   -groundNet  AVSS
run_top_simulation  -cktfile topSim/input.ckt  -emir_conf_file topSim/emir.conf
```

The `blockpgv_vialayer.txt` file contains the following information:

```
set_pgv -module DiffOpAmp -pgv_dir ./blockPGV/vfi_pgv_hier.via -view snapped
```

Sample 3: Specifying the PWL file

In the script sample below, user-specified PWL file, `DiffOpAmp.pwl` is used instead of the default PWL file in the pgv_dir. The `vpsbatch_automate_pwl.cmd` is run and the block-level PGV data is picked from the `vfi_pgv_hier.via` directory.

```
vpsbatch -cmd vpsbatch_automate_pwl.cmd
```

Where,

the `vpsbatch_automate_pwl.cmd` includes the following information:

```
include blockpgv_pwl.txt
extract_xdspf -signalEM true \
   -process n20 \
   -techLib ./pvtech.lib -techName gpdk045  -techCorner "rcx_typical"  \
   -inputType pegasus -inputDir topQRC/svdb  -RunName
DiffOpAmp_top_2subblock_VSSport \
   -temperature 25 \
   -groundNet  AVSS
run_top_simulation  -cktfile topSim/input.ckt  -emir_conf_file topSim/emir.conf
```


© 2020 All Rights Reserved.

The `blockpgv_pwl.txt` file contains the following information:

```
set_pgv -module DiffOpAmp -pgv_dir ./blockPGV/vfi_pgv_hier.via -instances {XI1 } -
pwl {blockSim/psf/input.emirtap/DiffOpAmp.pwl}
```


© 2020 All Rights Reserved.

Analyzing the EMIR Results for Hierarchical Analysis

This section covers how to load the EMIR results, view the plots, and analyze the results. The following topics are covered:

- Loading the IR/EM Results for the Hierarchical Flow in the GUI
- Viewing Plots for Top and Block-Level Results
- Analyzing the EMIR Results

Loading the IR/EM Results for the Hierarchical Flow in the GUI

In the IR/EM Results form, do the following:

1. Specify the *State Directory/Results File*, which is the `emir_bin` file generated after the hierarchical simulation.
2. Specify the *DFII layermap* file.
3. Click *Load Results*.

© 2020 All Rights Reserved.

Figure 15-8 Loading the IR/EM Results for the Hierarchical Flow

The *Net Type* and *Net Name* information are populated, as shown above. In addition, the table displays the block-level information. For example, the first two rows in the table display the type of net and the corresponding net name (`AVDD` and `AVSS`). The subsequent rows

© 2020 All Rights Reserved.

display the names of nets for particular blocks (`X11.AVDD`, `X11.AVSS`, and so on). You can plot the results for both the top level and block level.

Viewing Plots for Top and Block-Level Results

After loading the results, do the following:

1. Select *Combine Hier Nets*. This check box is selected by default. It plots top nets and the associated block nets together. The option is available only when the hierarchical EMIR results are loaded and is ON by default.
2. Select a top net and click *Show Plot*.

   The associated block nets are selected automatically. The plots of the top net and the block nets (only the interface nodes) are displayed on the Virtuoso layout. When the *Combine Hier Nets* check box is not selected, the nets are treated individually and are plotted with their full data.

The first image shows the IR drop analysis plot for the selected net, `AVDD`, at the top level. The second image shows the IR drop analysis results for the net for a specific block, `X11.AVDD`.

© 2020 All Rights Reserved.

Figure 15-9 Viewing the Top-Level Results for the Selected Net, AVDD

© 2020
All Rights Reserved.

Figure 15-10 Viewing the Block-Level Results for the Block X11 of the AVDD Net

Analyzing the EMIR Results

You can analyze and debug the various IR and EM plots for the block and top levels, either together or separately. This is depicted in the diagrams below.

© 2020 All Rights Reserved.

Figure 15-11 Viewing the Top- and Block-Level Results Together

Figure 15-12 Viewing the Top- and Block-Level Results Separately

© 2020
All Rights Reserved.

Sample File Formats for Hierarchical EMIR analysis

This section covers the following sample file formats:

- Sample Block-Level xDSPF Files
- A Sample Top-Level xDSPF file
- A Macro LEF File

Sample Block-Level xDSPF Files

The following sample files are provided below:

- A Sample Detailed xDSPF File
- A Sample Port xDSPF File
- A Sample Via Layer xDSPF

A Sample Detailed xDSPF File

A part of a sample top-level xDSPF file is shown below.

```
------------------------------------------------------------
*|DELIMITER #
*|BUSBIT []
*|GLOBAL_TEMPERATURE 110

.SUBCKT DiffOpAmp AVDD AVSS Ibias inm inp outm outp

*|GROUND_NET 0

*|NET AVDD 0
** Interface nodes: 3
** Regular nodes: 1703
** Tap nodes: 200
** Resistors: 3178
*|P (AVDD%2 I 0 43.17 47.708)
*|S (AVDD 0.041 47.708)
*|S (AVDD%1 25.13 49.308)
*|I (MM5@49#s MM5@49 s I 0 27.75 36.17)
*|I (MM5@48#s MM5@48 s I 0 27.75 29.02)
```


© 2020 All Rights Reserved.

```
*|I (MM0@49#s MM0@49 s I 0 25.931 36.17)
*|I (MM0@48#s MM0@48 s I 0 25.931 29.02)
*|I (MM5@44#s MM5@44 s I 0 38.711 29.02)
*|I (MM5@45#s MM5@45 s I 0 38.711 36.17)
*|S (AVDD#3 2.655 44.265)
*|S (AVDD#4 2.655 43.965)
*|S (AVDD#5 2.655 43.485)
R1_AVDD MM13@7#s MM4@6#s 0.001 $pdiff_conn $W=10000 $L=0 $X=31.4005 $Y=42.985
R2_AVDD MM0@43#s MM0@47#s 0.001 $pdiff_conn $W=10000 $L=0 $X=14.8105 $Y=36.17
R3_AVDD MM0@42#s MM0@46#s 0.001 $pdiff_conn $W=10000 $L=0 $X=14.8105 $Y=29.02
-------------------------------------------------------------------------------
```

where,

`*|P` - interface nodes for the block

`*|I` - current taps

`*|S` - sub nodes of the block

`R` - resistor information

A Sample Port xDSPF File

A part of a sample port xDSPF file is shown below.

```
-----------------------------------------------------------------
Port DSPF:
*|DSPF 1.0
*|DESIGN "DiffOpAmp"
*|VENDOR "Cadence Design Systems, Inc."
*|PROGRAM "Voltus_libgen"
*|DIVIDER /
*|DELIMITER #
*|BUSBIT []
*|GLOBAL_TEMPERATURE 110

.SUBCKT DiffOpAmp AVDD AVSS Ibias inm inp outm outp
*|GROUND_NET 0
*|NET AVDD 0
** Interface nodes: 3
** Regular nodes: 0
** Tap nodes: 3
```


© 2020 All Rights Reserved.

```
** Resistors: 0
*|P (AVDD%2 I 0 43.17 47.708)
*|P (AVDD 0.041 47.708)
*|P (AVDD%1 25.13 49.308)

*|NET AVSS 0
** Interface nodes: 1
** Regular nodes: 0
** Tap nodes: 1
** Resistors: 0
*|P (AVSS I 0 0.041 2.15)
*|NET Ibias 0
** Interface nodes: 1
** Regular nodes: 0
** Tap nodes: 1
** Resistors: 0
*|P (Ibias I 0 0.04 38.86)
ENDS
-----------------------------------------------------------------
```

A Sample Via Layer xDSPF

A part of a sample via layer xDSPF file is shown below.

```
-----------------------------------------------------------------
*|DSPF 1.0
*|DESIGN "DiffOpAmp"
*|VENDOR "Cadence Design Systems, Inc."
*|PROGRAM "Voltus_libgen"
*|DIVIDER /
*|DELIMITER #
*|BUSBIT []
*|GLOBAL_TEMPERATURE 110

.SUBCKT DiffOpAmp AVDD AVSS Ibias inm inp outm outp
*|GROUND_NET 0
*|NET   AVDD    0
** Interface nodes:     3
** Regular nodes:       1580
** Resistors:   1670
*|P (AVDD%2 I 0 43.17 47.708)
```


© 2020 All Rights Reserved.

```
*|S (AVDD 0.041 47.708)
*|S (AVDD%1 25.13 49.308)
*|I (AVDD#1144 MM5@49 s I 0 26.09 33.245)
*|I (AVDD#1065 MM5@48 s I 0 24.21 33.245)
*|I (AVDD#1019 MM0@49 s I 0 23.27 33.245)
*|I (AVDD#990 MM0@48 s I 0 22.33 33.245)
*|I (AVDD#619 MM0@47 s I 0 14.81 33.245)
*|I (AVDD#806 MM0@46 s I 0 18.57 33.245)
*|I (AVDD#619 MM0@45 s I 0 14.81 33.245)
*|S (AVDD#4 2.655 43.965)
*|S (AVDD#5 2.655 43.485)
*|S (AVDD#6 2.655 43.005)
*|S (AVDD#7 2.655 42.525)
R1_AVDD AVDD#3 AVDD#4 0.196186 $metal1_conn $W=600 $L=3000 $X=2.655 $Y=44.115
R2_AVDD AVDD#4 AVDD#5 0.313897 $metal1_conn $W=600 $L=4800 $X=2.655 $Y=43.725
R3_AVDD AVDD#5 AVDD#6 0.313897 $metal1_conn $W=600 $L=4800 $X=2.655 $Y=43.245
R4_AVDD AVDD#6 AVDD#7 0.313897 $metal1_conn $W=600 $L=4800 $X=2.655 $Y=42.765
R5_AVDD AVDD#7 AVDD#8 0.313897 $metal1_conn $W=600 $L=4800 $X=2.655 $Y=42.285
.ENDS
-------------------------------------------------------------------------------
```

A Sample Top-Level xDSPF file

A part of a sample top-level xDSPF file is shown below.

```
-------------------------------------------------------------------------------
*|DSPF 1.5
*
*|DATE "Fri Jan  6 09:00:59 2017"
*|VENDOR "Cadence Design Systems"
*|PROGRAM "Cadence Quantus Extraction"
*|VERSION "16.1.1 Linux 64 bit - (Wed Dec  7 10:12:15 PST 2016)"
*|DESIGN "DiffOpAmp_top_2subblock"
*|DIVIDER /
*|DELIMITER #
*|DeviceFingerDelim "@"
*|BUSBIT []
*|GLOBAL_TEMPERATURE 25
*|OPERATING_TEMPERATURE 25
** caps2d version: 10
*
```


© 2020 All Rights Reserved.

```
.SUBCKT  DiffOpAmp_top_2subblock inm outm Ibias inp AVDD AVSS outp
*
*Net Section
*
*|GROUND_NET AVSS
*
*LAYER_MAP
*0 none
*1 metal11_conn
*2 metal10_conn
*6 Via1
*7 Via2
*15 via10_cap

*|NET inm 2.06705e-15
*|P (inm X 0 13.9350 2.2885) // $llx=13.9350 $lly=2.2885 $urx=13.9350 $ury=2.2885
$lvl=5
*|I (XI1#inm XI1 inm X 0 13.9350 2.8700) // $llx=13.9350 $lly=2.8700 $urx=13.9350
$ury=2.8700 $lvl=5
*|S (inm#1 13.9350 3.1465) // $llx=13.9350 $lly=3.1465 $urx=13.9350 $ury=3.1465
$lvl=5
*|S (inm#2 13.9350 2.2485) // $llx=13.9350 $lly=2.2485 $urx=13.9350 $ury=2.2485
$lvl=5
C1 inm#2 AVSS 1.36516e-17
C2 XI1#inm AVSS 5.35868e-17
C3 inm AVSS 1.99981e-15
Rk1 inm#1 XI1#inm 0.158887 $metal1_conn  $L=0.277 $W=0.08 $X=13.935 $Y=3.008 $lvl=5
Rk2 XI1#inm inm 0.334151 $metal1_conn  $L=0.5815 $W=0.08 $X=13.935 $Y=2.579 $lvl=5
Rk3 inm inm#2 0.022985 $metal1_conn  $L=0.04 $W=0.08 $X=13.935 $Y=2.2685 $lvl=5
*
*|NET outm 5.00591e-15
*|P (outm X 0 61.3645 2.6400) // $llx=61.3645 $lly=2.6400 $urx=61.3645 $ury=2.6400
$lvl=5
*|I (XI2#outm XI2 outm X 0 61.3650 3.0300) // $llx=61.3650 $lly=3.0300 $urx=61.3650
$ury=3.0300 $lvl=5
*|S (outm#1 70.1200 27.5500) // $llx=70.1200 $lly=27.5500 $urx=70.1200 $ury=27.5500
$lvl=15
C4 outm#65 AVSS 3.96163e-17
C5 outm#64 AVSS 1.32595e-17
C6 outm#63 AVSS 8.5239e-17
Ra4 outm#2 outm#5 0.007130 $metal11_conn  $L=1.64 $W=4.8 $X=68.475 $Y=26.73 $lvl=1
Ra5 outm#5 outm#8 0.007152 $metal11_conn  $L=1.645 $W=4.8 $X=68.475 $Y=25.0875
$lvl=1
```


© 2020 All Rights Reserved.

```
Ra6 outm#8 outm#11 0.020521 $metal11_conn  $L=0.7595 $W=4.8 $X=68.475 $Y=23.885
$lvl=1

--------------------------------------------------------------------------------
```

Where,

R/C is the resistor/capacitor information

A Macro LEF File

Sample Macro LEF file that contains the LEF information for the design is shown below.

```
VERSION 5.8 ;
BUSBITCHARS "[]" ;
DIVIDERCHAR "/" ;

PROPERTYDEFINITIONS
  LIBRARY artistTranslator STRING "4.2.1" ;
  LIBRARY artistVersion STRING "4.2.1" ;
  MACRO oaTaper STRING ;
  MACRO lastSavedExtractCounter INTEGER ;
  LAYER LEF58_TYPE STRING ;
  LAYER LEF58_ENCLOSURE STRING ;
  LAYER LEF58_SPACING STRING ;
  LAYER LEF58_WIDTH STRING ;
END PROPERTYDEFINITIONS

UNITS
  CAPACITANCE PICOFARADS 1 ;
  DATABASE MICRONS 2000 ;
END UNITS
MANUFACTURINGGRID 0.005 ;
LAYER OVERLAP
  TYPE OVERLAP ;
END OVERLAP
LAYER PWdummy
  TYPE MASTERSLICE ;
  PROPERTY LEF58_TYPE "TYPE PWELL ;" ;
  PROPERTY LEF58_SPACING "SPACING 0.3 ;" ;
  PROPERTY LEF58_WIDTH "WIDTH 0.3 ;" ;
END PWdummy
```


© 2020 All Rights Reserved.

```
LAYER Nwell
  TYPE MASTERSLICE ;
  PROPERTY LEF58_TYPE "TYPE NWELL ;" ;
  PROPERTY LEF58_SPACING "SPACING 0.6 ;" ;
  PROPERTY LEF58_WIDTH "WIDTH 0.3 ;" ;
END Nwell
---------------------------------------------------------------------------
```


© 2020 All Rights Reserved.

16

Batch Mode Execution

- Batch Mode Support in Voltus-Fi-XL
- extract_xdspf
- ground_pins
- power_pin
- run_static_simulation
- setup_simulation
- load_ir_results
- print_ir_report
- print_rlrp_report
- pin_2_pin_res
- print_subnode_to_tap_path_report
- load_em_results
- print_em_report
- load_view
- print_thermal_report
- analyze_esd_generic
- analyze_esd_current_density
- analyze_esd_network
- esd_optimizer
- setMultiCpuUsage
- setDistributeHost

© 2020 All Rights Reserved.

- create_pgv
- update_pgv
- set_mmsim_pgv_nets
- report_vfi_XL_ron
- search_res_id
- res_id_info
- search_node_id
- node_id_info

© 2020 All Rights Reserved.

Batch Mode Support in Voltus-Fi-XL

Voltus-Fi-XL can be run in batch mode and in the GUI. The batch mode flow of Voltus-Fi-XL supports both the configuration file or the `conf` file and the command file, `cmd_file`.

Command File Support

Use the following command to run Voltus-Fi-XL from the UNIX window, shell, or xterm:

```
vfibatch -cmd <cmd_file>
```

Where:

<cmd_file> is a file with a list of batch commands that define the input data, control the execution of the analysis, and specify the output location.

Note: You can also use the `emirreport` command, from the simulation hierarchy, to generate reports for IR and EM analysis in text, HTML, or CSV format.

Configuration File Support

Use the following command to run Voltus-Fi-XL by specifying the EMIR configuration file:

```
vfibatch -c <emir conf> -db <simulation directory path> -outdir

<output directory>
```

You can specify the control file options in the configuration file using the `emirutil` command.

Related Topics

Generating EMIR Analysis Reports in Voltus-Fi-XL, EMIR Control File Options Supported in Voltus-Fi-XL

© 2020 All Rights Reserved.

extract_xdspf

```
extract_xdspf
     -inputType {pegasus | pvs | calibre}
     -inputDir input_dir_name
     -RunName input_run_name
     -outputDir output_dir_name
     -process {n3 | n5 | n7prf | n7 | n10pg | n10 | n20 | n16 | n12 | others}
     -techDir tech_dir_name | -techLib tech_lib_file_name
     -signalEM {true | false}
     [-arrayViaSpacing {auto | value}]
     [-blockingCellFile parasitic_blocking_device_cell_names_file]
     [-bus_bit_char string]
     [-cclFileName vfiQuantus-<filename>.ccl]
     [-coloring {true | false}]
     [-commentSubCkt {true | false}]
     [-compression {true | false}]
     [-customExtract extract_Commands_File]
     [-device_properties_file file_name]
     [-fractureViaCnt via_count_number]
     [-groundNet ground_net_name]
     [-generate_ccl_only {true | false}]
     [-icell_map_file filename]
     [-keep_pin_shorting_res {true | false}]
     [-keep_pin_shorting_res_by_layer {"layer1" "layer2" .... }]
     [-lic_queue timeout_in_seconds]
     [-LVS_BOX_hierarchical_flow {true | false}]
     [-min_res_by_layer {true | false}]
     [-multiCpu number_of_cpus]
     [-outputDspf output_dspf_filename]
     [-output_xy_generic {true | false}]
     [-PGNets list_of_pg_nets]
     [-reduction {true | false}]
     [-stacked_via_effect {true | false}]
     [-subcktPinOrder pin_order_file]
     [-subNodeChar sub_node_char]
     [-subConductorModel {true | false}]
     [-techName technology_name]
     [-techCorner technology_corner_name]
     [-temperature value]
     [-tech_cmd_file filename]
     [-tech_layer_setup_file filename]
     [-tech_lpe_config_file ./lpe_confile]
     [-user_ccl_file ccl_file_name]
```

Generates the xDSPF file using Quantus from within Voltus-Fi-XL.

Note: Before running this command, ensure that the Quantus hierarchy is available because the tool will call the Quantus binary from this path, and also ensure that the Pegasus, PVS, or Calibre® run is performed successfully.

© 2020 All Rights Reserved.

Quantus has a plug-and-play flow that performs Techgen compilation automatically while doing extraction. This is intended to make it easier to maintain the technology files. When this feature is enabled, you can supply the raw techfiles from the foundry and Quantus takes care of the rest.

The following parameters are provided to enable this plug-and-play feature:

- `-tech_cmd_file` *`name of file`*
- `-tech_layer_setup_file` *`name of file`*
- `-icell_map_file` *`name of file`*
- `-tech_lpe_config_file` *`./lpe_confile`*

Note: The `extract_xdspf` command has the following parameters that are only used for hierarchical EMIR analysis.

```
-hier_flow {true | false}
-hier_flow_pgv {1.cl 2.cl}
```


© 2020 All Rights Reserved.

Parameters

`-arrayViaSpacing {auto | value}`	Specifies the via array spacing, which is the distance, a floating point micron value, that will be used as the maximum distance criteria for grouping vias within the same array. The default value of the via array spacing is 0.35 times the via size for `N7`, `N10`, `N16`, `N12`, and `N20` processes. For other processes, the default is `auto`, in which case the software sets the via spacing automatically. You can specify either `auto` or a floating value for this option. *Default*: `auto` This is an optional parameter.
`-blockingCellFile` *`parasitic_blocking_device_cell_names_file`*	Specifies the file containing the parasitic blocking device cell names, if any. Parasitic extraction for these devices will be skipped. This is an optional parameter.
`-bus_bit_char` *`string`*	Specifies the type of character to use as the bus notation in the xDSPF output file. If the argument specified for this option is `"[]"` (square brackets), then you must prefix the brackets using the backslash (`\`) character. For example: `-bus_bit_char "\[\]"` **Note:** No other argument except `[]` requires the (\) character. *Default*: `none` This is an optional parameter.
`-cclFileName vfiQuantus-<`*`filename`*`>.ccl`	Allows you to specify the name of the CCL file to be created. If specified, the CCL file is created as `vfiQuantus-<`*`filename`*`>.ccl` and if not specified, the file is created as `vfiQuantus-<`*`cellname`*`>.ccl`. *Default*: `none`
`-coloring {true | false}`	When set to `true`, this parameter enables color-aware EM analysis. This is an optional parameter. *Default*: `false`

© 2020 All Rights Reserved.

`-commentSubCkt {true | false}`	When set to `true`, this parameter will comment out the top sub-circuit statement in the xDSPF file. This is an optional parameter.
`-compression {true | false}`	When set to `true`, specifies that the xDSPF file should be generated in a compressed format. This is an optional parameter.

`-customExtract` *`extract_Commands_File`*	Specifies the file with user-customized `extract` commands. When this file is specified, the Voltus-Fi-generated CCL will not have any extract commands and the commands from the customized extract commands file will be added as is in the Voltus-Fi-generated CCL.
	
	A sample customized `extract` command fie is as follows:
	
	`extract \`
	`  -selection "all" \`
	`  -type "rc_coupled"`
	
	Note: You can create `customextract` file by using either single or multiple `extract` settings. Ensure that the commands are in the Quantus CCL file syntax as this file is added as is in the Voltus-Fi-generated CCL.
	
	This is an optional parameter.
	
	For information about the format of the Voltus-Fi-generated CCL file and its example, see Common Command Language (CCL) File.

`-device_properties_file` *`file_name`*	Specifies the name of the device properties file when the input type is `Calibre`. This is an optional parameter.
`-fractureViaCnt` *`via_count_number`*	Specifies the fracture via count to be used for dividing pseudo vias into segments during extraction. *Default*: `1`

© 2020 All Rights Reserved.

`-generate_ccl_only {true | false}`	When set to `true`, generates a CCL file for Quantus but does not execute the Quantus binary. This allows you to modify the CCL file that is generated to create your own customized CCL file. This customized file can then be specified for running Quantus by using the `-user_ccl_file` option. *Default*: `false`
`-groundNet` *`ground_net_name`*	Specifies the name of the ground net on which extraction is to be performed. *Default*: `0` This is an optional parameter.
`-inputDir` *`input_dir_name`*	Specifies the directory in which the Pegasus, PVS, or Calibre data is stored. This is a required parameter.
`-inputType {pegasus | pvs | calibre}`	Specifies whether the input data is from the Pegasus or Calibre tool. This is a required parameter.
`-keep_pin_shorting_res {true | false}`	Preserves the device pin shorting resistors on the `*wires` layers. The device pin shorting resistors are identified by the layer names of "`wires`". For example, the layers "`mwires`", "`qwires`", "`rwires`", "`cwires`", "`bwires`", and "`dwires`", consist of shorting resistors for the MOS, BJT, R, C, generic, and D devices respectively. When set to `true`, the shorting resistors will not be removed. *Default*: `false` This is an optional parameter.
`-keep_pin_shorting_res_by_layer {"`*`layer1`*`" "`*`layer2`*`" .... }`	Specifies the shorting layer names to be preserved. The valid layer names are "`mwires`", "`qwires`", "`rwires`", "`cwires`", "`bwires`", and "`dwires`". This is an optional parameter.
`-lic_queue` *`timeout_in_seconds`*	Specifies the license timeout wait time in seconds. This is an optional parameter.

© 2020 All Rights Reserved.

`-LVS_BOX_hierarchical_flow {true | false}`	When set to `true`, enables the LVS box hierarchical flow of Quantus. This flow requires some additional options to be included in the Quantus CCL file. When the `-LVS_BOX_hierarchical_flow` parameter is set to `true`, these additional options are printed in the Quantus CCL file and the flow is enabled. *Default*: `false` This is an optional parameter.

`-min_res_by_layer {true | false}`	When this parameter is set to true, the following information is added in the CCL file:
	
	`filter_res \`
	`  -min_res_by_layer "RH_TN_6 0" \`
	`  -min_res 0.001 \`
	`  -remove_dangling_res false`
	
	This means that all RH_TN_6 layers are reserved without threshold resistor value.
	
	When this parameter is set to false (default) or not set, the following information is added in the CCL file:
	
	`filter_res \`
	`  -min_res 0.001 \`
	`  -remove_dangling_res false`
	
	This is an optional parameter.
	
	Default: false

`-multiCpu` *`number_of_cpus`*	Specifies the number of CPUs to be used for distributed processing during the Quantus extraction run. This is an optional parameter. By default, single CPU is used.
`-outputdir` *`output_dir_name`*	Specifies the directory in which the xDSPF file that is generated will be saved. This is a required parameter.
`-outputDspf -` *`output_dspf_filename`*	Specifies the name of the xDSPF file that is generated. This is an optional parameter. If the filename is not specified, the software generates the file, `runName.dspf` by default.

© 2020 All Rights Reserved.

`-output_xy_generic {true | false}`	When set to `true`, provides the x-axis and y-axis coordinates for a high resistance design resistor. This parameter introduces the GENERIC option in the CCL file. By default, this parameter is set to `true`. When this parameter is set to `false`, there will be no GENERIC option in the CCL file.
`-PGNets` *`list_of_pg_nets`*	Provides the list of PG nets, separated by spaces, for which the xDSPF file is to be generated. This is an optional parameter. It is only specified when the `-signalEM` parameter is set to `false`.
`-process {n3 | n5 | n7prf | n7 | n10pg | n10 | n20 | n16 | n12 | others}`	
	Specifies the process node. `n3`: for the N3 process node `n5`: for the N5 process node `n7prf`: for the N7PRF process node `n7`: for the N7 process node `n10pg`: for the N10 process node with PG rules `n10`: for the N10 process node `n20`: for the N20 process node `n16`: for the N16 process node `n12`: for the N12 process node `others`: for the N20 and above process nodes
`-reduction {true | false}`	When set to true, this parameter enables the `reduce_i_cards` option while generating the xDSPF file. This is an optional parameter. **Note:** It is recommended that this option should be set to `false` while performing extraction for the ESD analysis flow.
`-RunName` *`input_run_name`*	Specifies the name of the Pegasus, PVS, or Calibre run. This is a required parameter.

© 2020 All Rights Reserved.

`-signalEM {true | false}`	Specifies whether the names of signal nets should be included in the xDSPF file or not. If signal nets are not to be included in the EM analysis, set this parameter to `false`.
`-stacked_via_effect {true | false}`	When set to `true`, reports the `W` parameter as 1,000,000 microns. When set to `false`, reports the actual wire width for the short horizontal metal segment. If a wire segment between the inner layers of adjacent stacked vias is short, the current flowing in the short horizontal segment is expected to be mostly vertical, whereas the horizontal currents may be non-real. By setting this option to true, you can modify the W parameter to be set to 1,000,000 microns to prevent false EM violations. This is an optional parameter. *Default*: `false`
`-subConductorModel {true | false}`	When set to true, specifies that parasitic resistor models for subconductor layers should be included during extraction. This is used to support different EM rules for poly layers in EM analysis. This is an optional parameter.
`-subNodeChar` *`sub_node_char`*	Specifies the character that will be used for subnode identification. *Default*: `#` This is an optional parameter.
`-subcktPinOrder` *`pin_order_file`*	Specifies the name of the sub-circuit pin order file. The pin order is specified in the xDSPF file using the `.SUBCKT` and `.ENDS` statements. You can use this option to provide a file with a different pin order that will override the order specified using the default `.SUBCKT` statement in the xDSPF file. This is an optional parameter.
`-techDir` *`tech_dir_name`* `| -techLib` *`tech_lib_file_name`*	

© 2020 All Rights Reserved.

	Use either the `-techDir` parameter to specify the name of the directory containing the tech file or use the `-techLib` parameter to specify the path of the technology file that Quantus will use for generating the xDSPF file. One of the two parameters must be specified.
`-techCorner` *`technology_corner_name`*	Specifies the technology corner name to be picked from the library. This parameter is optional.
`-techName` *`technology_name`*	Specifies the technology filename to be used for extraction. This parameter is optional if the `-techDir` parameter has been specified.

`-temperature` *`value`*	Specifies the temperature at which the extraction will be performed. It is specified in degree Celsius.
	
	This is an optional parameter. If not specified, the software will use 25 degree Celsius as the default temperature.
	
	The options, `-temperature` and `temp_coeff` specified in the CCL file are mutually exclusive. The following scenarios hold true.
	
	■ If you specify a temperature by using this parameter, the following option in the CCL file will be set to `false`:
	`-include_parasitic_res_temp_coeff "false" \`
	The software displays the following message:
	`User specified temperature will be used for extraction. The option include_parasitic_res_temp_coeff will be set as false.`
	
	■ If you do not specify a temperature, the following option in the CCL file will be set to `true`:
	`-include_parasitic_res_temp_coeff "true" \`
	The software displays the following message:
	`No user specified temperature value so the option include_parasitic_res_temp_coeff will be set as true.`

© 2020 All Rights Reserved.

`-user_ccl_file` *`ccl_file_name`*	Specifies the name of the user-customized CCL file to be used for running Quantus.
`-tech_cmd_file` *`filename`*	Specifies the name of the command file to enable the plug and play feature of Quantus.
`-tech_layer_setup_file` *`filename`*	Specifies the name of the technology layer setup file to be used for Techgen compilation. This parameter is only relevant for the plug and play feature of Quantus.
`-tech_lpe_config_file` *`./lpe_confile`*	Specifies the name of the technology LPE configuration file that is to be passed to Quantus. This parameter is only relevant for the plug and play feature of Quantus.
`-icell_map_file` *`filename`*	Specifies the foundry provided `iCellmap` file that contains simulation model to DFII cell view mappings and transfer property information. This file is specified only for enabling the plug and play feature of Quantus.

Examples

- The following command generates the xDSPF file, `test_lib.dspf` in the `vfi_quantus1` directory:

```
extract_xdspf \
    -signalEM true \
    -process n20 \
    -techLib pegasus_tech.lib \
    -techName n20tech \
    -temperature 27 \
    -inputType pegasus \
    -inputDir test_lib/pegasus \
    -RunName test_lib \
    -outputdir vfi_quantus1 \
    -subcktPinOrder ./your_subckt_pin_order_file
    -customExtract myextract.txt
    -groundNet "vss" \
    -subNodeChar ":" \
    -compression true \
    -lic_queue 200
```

- The following set of commands are used to generate the xDSPF file, `test_lib.dspf` in the `vfi_quantus1` directory using the user-customized CCL file:

© 2020 All Rights Reserved.

Step 1:

```
extract_xdspf \
    -signalEM true \
    -process n20 \
    -techLib pegasus_tech.lib \
    -techName n20tech \
    -temperature 27 \
    -inputType pegasus \
    -inputDir test_lib/pegasus \
    -RunName test_lib \
    -outputdir vfi_quantus1 \
    -generate_ccl_only true
```

Step 2:

```
extract_xdspf \
-user_ccl_file vfiquantus-cell1.ccl \
```

Related Topic

The Hierarchical EMIR Analysis Batch Mode Flow

© 2020 All Rights Reserved.

ground_pins

```
ground_pins gnd_pin_list
```

Specifies the names of the virtual ground pins to which the SPICE subcircuit terminal will be connected. This command is used to set up static current analysis in Voltus-Fi-XL.

Parameters

gnd_pin_list	Specifies the names of the virtual ground pins.

Example

The following command is used to specify the names of two ground pins, `VSS` and `VSSA`:

```
ground_pins VSS VSSA
```

Related Topics

power_pin, run_static_simulation, setup_simulation

power_pin

```
power_pin
     -pin_name pin_name
     -voltage voltage_value
```

Specifies the power pin and its corresponding voltage value.To specify multiple power pins, use this command multiple times. This command is used to set up static current analysis in Voltus-Fi-XL.

© 2020 All Rights Reserved.

Parameters

`-pin_name` *`pin_name`*	Specifies the name of the virtual power pin to which the SPICE subcircuit terminal should be connected.
`-voltage` *`voltage_value`*	Specifies the voltage value of the power pin.

Example

The following commands are used to specify the names of two power pins, `VDD` and `VDDA`, and their corresponding voltage values:

```
power_pin \
    -pin_name VDD \
    -voltage 0.9
power_pin \
    -pin_name VDDA \
    -voltage 1.1
```

Related Topics

ground_pins, run_static_simulation, setup_simulation

run_static_simulation

```
run_static_simulation
     -method {Ipeak | Iavg}
     -nets net_name_list
     [-freq freq_value]
     [-global_activity number]
     [-activity_file file_name]
     [-vcd vcd_file_name]
     [-clock_file clock_file_name]
     [-cmd_include_file command_file_name]
     [-average_power power_value_in_W]
     [-power_file power_file_name]
```

Runs static current analysis in Voltus-Fi-XL.

© 2020 All Rights Reserved.

Parameters

`-activity_file` *`file_name`*	Specifies the path of the activity file, which contains the activity factor on any node of the circuit. This is an optional parameter. It is only used for the average static current analysis.
`-average_power` *`power_value_in_W`*	Specifies the average power of the design. The default unit is `W` (Watts). If the value is in any other unit, it must be specified. For example, `20mW`. You can specify a single value or multiple values depending upon whether the same or a different power is to be used for the specified nets. This is shown in the examples below. This is an optional parameter. This is only used for the average static current analysis.
`-clock_file` *`clock_file_name`*	Specifies the name of the clock file, which contains information about the clock nodes, subcircuit definitions, and subcircuit instance names and their frequencies. This is an optional parameter. This is only used for the average static current analysis.
`-cmd_include_file` *`command_file_name`*	
	Specifies the name of the file that passes additional user commands to Thunder during static current analysis. This is an optional parameter.
`-freq` *`freq_value`*	Specifies the value of the dominant frequency of the design. The unit is in `mHz` (Mega Hertz). This is an optional parameter. This is only used for the average static current analysis.
`-global_activity` *`number`*	Specifies the average number of times all the unset nodes switch in a clock cycle. This is an optional parameter. This is only used for average static current analysis.
`-method {Ipeak | Iavg}`	Specifies whether the peak or average static analysis will be performed. This is a required parameter.

© 2020 All Rights Reserved.

`-nets` *`net_name_list`*	Specifies the list of nets for which current analysis is to be performed. This is a required parameter.
`-power_file` *`power_file_name`*	Specifies the name of the average power file, which contains the average power for specific sub circuits. This is an optional parameter. This is only used for the average static current analysis.
`-vcd` *`vcd_file_name`*	Specifies the vcd file, which contains information about the number of transitions for each net. This is an optional parameter. This is only used for the average static current analysis.

Examples

- The following command is used to run the average static current analysis.
  - In this example, the same power value is assigned to the two nets, `VDD` and `VSS`:

```
run_static_simulation \
   -method Iavg \
   -nets {VDD VSS} \
   -freq 300 \
   -global_activity 0.25 \
   -average_power 20mW
```

  - In this example, different power values are assigned to the three nets, `VDD_1`, `VDD_2`, and `VSS_1`:

```
run_static_simulation  \
   -method Iavg \
   -nets {VDD_1 VDD_2 VSS_1} \
   -freq 300 \
   -global_activity 0.25 \
   -average_power {10mW 8mW 18mW}
```

    Note: The above command is only valid if you provide power values for all the specified nets.

- The following command is used to run the peak static current analysis:

```
run_static_simulation \
    -method Ipeak \
    -nets {VDD VSS}
```


© 2020 All Rights Reserved.

Related Topics

ground_pins, power_pin, setup_simulation, Static Current Analysis Files

setup_simulation

```
setup_simulation
     -spice_netlist spice_file_name
     [-simulation_temperature temp_value]
     [-hierarchy_separator char_name]
     [-topcell cell_name]
     [-output_directory dir_name]
```

Sets up the options for static current analysis in Voltus-Fi-XL.

Parameters

`-hierarchy_separator` *`char_name`*	Specifies the hierarchical separator in the circuit file. This is an optional parameter.This is specified only if the hierarchy separator is other than "`/`".
`-output_directory` *`directory_name`*	Specifies the name of the output directory where the current database will be saved. This is an optional parameter. If this parameter is not specified, the output is stored in the current working directory.
`-simulation_temperature` *`temp_value`*	Specifies the temperature, in degree Celsius, at which the simulation will be performed. This is an optional parameter. *Default*: 25 degree Celsius
`-spice_netlist` *`spice_file_name`*	Specifies the SPICE netlist file, which is the DSPF file. This is a required parameter.
`-topcell` *`cell_name`*	Specifies the name of the design. It is an optional parameter. By default, the tool populates the top cell name from that specified in the layout.

Example

The following command is used to set up simulation:

© 2020 All Rights Reserved.

```
setup_simulation \
    -spice_netlist  design.spice \
    -simulation_temperature 100 \
    -topcell designA \
    -output_directory ./vps_simulation_1
```

Related Topics

ground_pins, power_pin, run_static_simulation

load_ir_results

```
load_ir_results
     results_file_name
```

Loads the results of IR drop analysis in Voltus-Fi-XL.

Parameter

results_file_name	Specifies the directory in which the IR drop analysis results should be loaded.

Example

The following command loads the results for IR drop analysis that are stored in the `xps.emirtap.emir0_bin` file:

```
load_ir_results \
./abc/raw.out/xps.emirtap.emir0_bin
```

Related Topics

print_ir_report, print_rlrp_report, pin_2_pin_res, print_subnode_to_tap_path_report

print_ir_report

```
print_ir_report
     -net {all_power | netname}
     -type {ir |iv |rc |rcavg |rcrms | reffective}
     [-threshold threshold_value]
     -filename output_file_name
```


© 2020 All Rights Reserved.

Prints the results of IR drop analysis in Voltus-Fi-XL. You can generate IR reports for all the power nets or for specific nets.

Note: Before running this command, ensure that the IR drop results are loaded using the `load_ir_results` command.

© 2020 All Rights Reserved.

Parameters

`-filename` *`output_file_name`*	Specifies the name of the IR drop analysis report file. This is a required parameter.
`-net {all_power |` *`netname`*`}`	Specifies whether the IR drop analysis report is to be printed for all the power nets or for the specified net. Select `all_power` to print the IR report for all the power nets. Select *`netname`* to print the IR report for the specified net. This is a required parameter.
`-threshold` *`threshold_value`*	Specifies that resistors or nodes that have a threshold IR drop ratio value greater than the specified value will be reported. This is an optional parameter. *Default*: The report prints results for all resistors or nodes.
`-type {ir | iv | rc |rcavg |rcrms | reffective}`	
	Specifies the analysis type for which the report is being generated. `ir` – specifies that the report is generated for IR drop analysis. `iv` – specifies that the report is generated for transistor-based supply voltage data analysis. `rc` – specifies that the report is generated for peak resistor Current Density analysis. `rcavg` – specifies that the report is generated for average resistor Current Density analysis. `rcrms` – specifies that the report is generated for RMS resistor Current Density analysis. `reffective` – specifies that the report is generated for effective resistance analysis. This is a required parameter.

Examples

- The following command prints the IR drop analysis report for all the nets for `ir` analysis in the `all_ir.txt` file:

```
print_ir_report \
-net all_power \
```


© 2020 All Rights Reserved.

```
-type ir \
-filename all_ir.txt
```

- The following command prints the IR drop analysis report for the `VDD` net for `iv` analysis in the `vdd_iv.txt` file:

```
print_ir_report \
-net VDD \
-type iv \
-filename vdd_iv.txt
```

- The following command prints the IR drop analysis report for the `VSS` net for `rc` analysis in the `vss_rc.txt` file:

```
print_ir_report \
-net VSS \
-type rc \
-filename vss_rc.txt
```

- The following command prints the IR drop analysis report for all power nets for `rcrms` analysis in the `all_power_rcRMS.txt` file:

```
print_ir_report \
-net all_power \
-type rcrms \
-filename all_power_rcRMS.txt
```

Related Topics

load_ir_results, print_rlrp_report, pin_2_pin_res, print_subnode_to_tap_path_report

print_rlrp_report

```
print_rlrp_report
     -net netname
     -filename output_file_name
     -tap tapname
```

Prints the RLRP analysis report in Voltus-Fi-XL.

Note: Before running this command, ensure that the IR drop results are loaded using the `load_ir_results` command.

© 2020 All Rights Reserved.

Parameters

-filename *output_file_name*	
	Specifies the name of the RLRP analysis report file. This is a required parameter.
-net *netname*	Specifies the name of the net for which the RLRP report is to be printed. This is a required parameter.
-tap *tapname*	Specifies the instances or tap-nodes of the specified net for which the LRP values are to be printed. You can specify multiple tap nodes for the specified net as shown in the example below. This is a required parameter.

Example

Use the following commands to load the IR drop analysis results and print the RLRP analysis report for the `TVDD` net in the `RLRP.rpt` file:

```
load_ir_results ./abc/raw.out/xps.emirtap.emir0_bin
print_rlrp_report -net TVDD -filename RLRP.rpt -tap MavD7_1_unmatched#d
print_rlrp_report -net TVDD -filename RLRP.rpt -tap MavD7_2_unmatched#d
```

pin_2_pin_res

```
pin_2_pin_res
     -xdspf filename
     -x1 value
     -y1 value
     -x2 value
     -y2 value
     -layer1 layername
     -layer2 layername
     -net netname
```

Calculates the effective resistance between any two nodes (pins, tap nodes, or subnodes), either on the same layer or on different layers of the same net in Voltus-Fi-XL. Use this command after the `load_ir_results` command.

Note: All parameters of this command are required parameters.

© 2020 All Rights Reserved.

Parameters

`-xdspf` *`filename`*	Specifies the location of the xDSPF file to be used for the Static Power Grid Solver (SPGS) flow of Spectre.
`-x1` *`value`*	Specifies the x-coordinate for node 1 in microns.
`-y1` *`value`*	Specifies the y-coordinate for node 1 in microns.
`-x2` *`value`*	Specifies the x-coordinate for node 2 in microns.
`-y2` *`value`*	Specifies the y-coordinate for node 2 in microns.
`-layer1` *`layername`*	Specifies the name of the layer for node 1.
`-layer2` *`layername`*	Specifies the name of the layer for node 2.
`-net` *`netname`*	Specifies the name of the net on which the two nodes are located.

Example

Use the following commands to load the IR results and specify the nodes between which you want to calculate effective resistance.

```
load_ir_results \
./abc/raw.out/xps.emirtap.emir0_bin

pin_2_pin_res \
    -xdspf abc/raw.out/abc_sample.dspf \
    -x1 83.40 \
    -y1 34.205 \
    -x2 83.40 \
    -y2 35.205
    -layer1 mt1 \
    -layer2 mt2 \
    -net VDD
```

Related Topics

load_ir_results, print_ir_report, print_rlrp_report, print_subnode_to_tap_path_report

© 2020 All Rights Reserved.

print_subnode_to_tap_path_report

```
print_subnode_to_tap_path_report
     -net {netname}
     -subnode {subnode name}
     -filename {report_file_name}
     [-never_traverse_top {true | false}]
     [-report_complete_path {true | false}]
```

Prints the results of the nearest tap nodes for the specified subnodes in Voltus-Fi-XL.

Note: Before running this command, ensure that the IR drop results are loaded using the `load_ir_results` command.

© 2020 All Rights Reserved.

Parameters

`-filename` *`output_file_name`*	
	Specifies the name of the report file to be generated. This is a required parameter.
`-never_traverse_top {true | false`	
	Specifies whether or not the software should traverse the top layers to find a tap node. By default, the software will traverse the top layers to reach the nearest tap node. Set this parameter to `true` if you do not want to traverse the top layers. *Default*: `false` This is an optional parameter.
`-net` *`netname`*	Specifies the name of the net for which the report is to be generated. This is a required parameter.
`-report_complete_path {true | false}`	
	Specifies whether or not the complete path for each traced tap node should be reported. By default, the software will not report the complete path.
`-subnode` *`subnode_name`*	
	Specifies the name of the subnode for which the report is to be generated. This is a required parameter.

Example

The following command prints the report for the `VDD` net for tap nodes of the subnode, `VDD:36919` in the `subnode_tap_path.rpt` file:

```
print_subnode_to_path_report \
-net VDD \
-subnode VDD:36919 \
-filename subnode_tap_path.rpt
```

Related Topics

load_ir_results, print_ir_report, print_rlrp_report, pin_2_pin_res

© 2020 All Rights Reserved.

load_em_results

```
load_em_results
     -tech_file tech_file_name
     [-em_only_ict_file ict_file_name]
     [-layer_mapfile layer_map_filename]
     results_file_name
```

Loads the results of EM analysis in Voltus-Fi-XL.

Note: The `qrcTechFile` name and layer map file must be specified if the layer names in the `qrctechFile` are different from those in the simulation database.

Parameters

`-em_only_ict_file` *`ict_file_name`*	
	Specifies the name of the ICT file with process and EM model information that will be used for EM analysis. This is an optional parameter. For a sample EM only ICT file, see the “File Formats” chapter.
`-layer_mapfile` *`layer_map_filename`*	
	Specifies the name of the layer map file that provides the mapping between the layer names used in the technology file and the simulation database. This is an optional parameter. It is used only when the layer names in the `qrcTechFile` are different from those in the simulation database.
`results_file_name`	Specifies the name of the file which contains the EMIR simulation results data. This is a required parameter.
`-tech_file -`*`tech_file_name`*	
	Specifies the name of the `qrcTechFile` or the `emDataFile` used for EM analysis. This is a required parameter.

Example

The following command loads the results of EM analysis stored in the `xps.emirtap.emir0_bin` file, and when the layer map file is `contactmapfile`:

```
load_em_results \
-tech_file qrcTechFile \
```


© 2020 All Rights Reserved.

```
-layer_mapfile contactmapfile /xps.emirtap.emir0_bin
```

Related Topic

print_em_report

print_em_report

```
print_em_report
     -net {all_power | all_signal | all_nets | netname}
     -type {javg | jmax | jabsavg | jacpeak | jacrms | rc | rcavg | rcrms | tc |
     tcavg | tcrms | acrms_sh | avg_sh | peak_sh}
     [-threshold threshold_value]
     -filename output_file_name
     [-rule custom_em_rule_name]
```

Prints the EM analysis reports in Voltus-Fi-XL.

Note: Before running this command, ensure that the EM results are loaded using the `load_em_results` command.

© 2020 All Rights Reserved.

Parameters

`-filename` *`output_file_name`*	
	Specifies the name of the EM analysis report file. This is a required parameter.
`-net {all_power | all_signal | all_nets |` *`netname`*`}`	
	Specifies that the EM analysis report is to be generated for one of the following: `all_power`: prints the report for all power nets `all_signal`: prints the report for all signal nets `all nets`: prints the report for all nets *`netname`*: prints the report for the specified net This is a required parameter.
`-rule` *`custom_em_rule_name`*	Specifies the custom EM rule for which the report will be created. This is an optional parameter.
`-threshold` *`threshold_value`*	
	Specifies that resistors or nodes that have a threshold value of EM ratio above the specified value will be reported. This is an optional parameter. *Default*: The report prints the results for all resistors or nodes.
`-type {javg |jmax | jabsavg | jacpeak | jacrms | rc | rcavg | rcrms | tc | tcavg | tcrms | acrms_sh | avg_sh | peak_sh}`	

© 2020 All Rights Reserved.

	Specifies the analysis type for which the report is being generated. `javg` – specifies that the report is generated for the average Current Density analysis data. `jmax` – specifies that the report is generated for the peak Current Density analysis data. `jabsavg` – specifies that the report is generated for the average absolute Current Density analysis data. `jacpeak` – specifies that the report is generated for the AC peak Current Density analysis data. `jacrms` – specifies that the report is generated for the RMS Current Density analysis data. `rc` – specifies that the report is generated for the peak resistor Current Density analysis data. `rcavg` – specifies that the report is generated for the average resistor Current Density analysis data. `rcrms` – specifies that the report is generated for the RMS resistor Current Density analysis data. `tc` – specifies that the report is generated for the peak tap Current Density analysis data. `tcavg` – specifies that the report is generated for the average tap Current Density analysis data. `tcrms` – specifies that the report is generated for the RMS tap Current Density analysis data. `acrms_sh` – specifies that the report is generated for AC RMS current analysis data with self-heating effect. `avg_sh` – specifies that the report is generated for average current analysis data with self-heating effect. `peak_sh` – specifies that the report is generated for peak current analysis data with self-heating effect. **Note:** The above three types of analyses are applicable only in the SHE flow. This is a required parameter.

© 2020 All Rights Reserved.

Examples

- The following command prints the EM analysis report for all the power nets for `javg` analysis in the `all_power_javg.txt` file:

```
print_em_report \
-net all_power \
-type javg \
-filename all_power_javg.txt
```

- The following command prints the EM analysis report for all the signal nets for `jmax` analysis in the `all_signal_jmax.txt` file:

```
print_em_report \
-net all_signal \
-type jmax \
-filename all_signal_jmax.txt
```

- The following command prints the EM drop analysis report for the VDD net for `jabsavg` analysis in the `vdd_jabsavg.txt` file:

```
print_em_report \
-net VDD \
-type jabsavg \
-filename vdd_jabsavg.txt
```

- The following command prints the EM drop analysis report for the VDD net for `tcrms` analysis in the `vdd_tcrms.txt` file:

```
print_em_report \
-net VDD \
-type tcrms \
-filename vdd_tcrms.txt
```

Related Topic

load_em_results

load_view

```
load_view
     -libname lib
     -cellname cell_name
     -viewname view_name
```

Specifies the layout to be used for self-heating effect (SHE) analysis in Voltus-Fi-XL.

© 2020 All Rights Reserved.

Parameters

`-cellname` *`cell`*	Specifies the name of the cell.This is a required parameter.
`-libname` *`lib`*	Specifies the name of the library. This is a required parameter.
`-viewname` *`layout`*	Specifies the layout view for which SHE analysis is to be performed. This is a required parameter.

Example

The following command specifies the `layout` view for `cell1` of library, `test_lib`:

```
load_view
     -libname test_lib \
     -cellname cell1 \
     -viewname layout
```

Related Topic

print_thermal_report

print_thermal_report

```
print_thermal_report
     -type {mos-region | metal}
     [-tiles tile_matrix]
     -filename output_file_name
```

Prints thermal reports for hierarchical self-heating effect (SHE) analysis in Voltus-Fi-XL.

Note: To generate hierarchical SHE analysis reports using this command, the VTS-Fi-AA license is required to be checked out. This is an advanced license that is not available as part of the Voltus-Fi-XL license.

If you do not want to perform hierarchical SHE analysis, ensure that this command is not specified in the command file.

The SHE EM analysis reports can be generated by specifying the `print_em_report` command in the command file with the required `-type` option. No special license is required for generating these reports.

© 2020 All Rights Reserved.

Parameters

`-filename` *`output_file_name`*	Specifies the name of the file in which the hierarchical SHE analysis results will be printed. This is a required parameter.
`-tiles` *`tile_matrix`*	Specifies the OD regions or metal resistors and their temperature change after hierarchical self-heating effect in tile format. The tile matrix is specified as follows: The design in the layout is divided into sections based on the tile matrix provided, which is the number of divisions in the x-direction multiplied by the number of divisions in the y-direction. For example: `-tiles 10x15` This is an optional parameter.
`-`*`type`* `{mos_region | metal}`	
	When `mos-region` is specified, the shape of OD regions and their temperature change after hierarchical self-heating effect are printed in the report. When metal is specified, the shape of metal resistors and their temperature change after hierarchical self-heating effect are printed in the report. This is a required parameter.

Example

The following sequence of commands will print the results of hierarchical SHE analysis for OD regions in a tile format in the `mos-region-tiles.txt` file:

```
load_em_results \
-tech_file qrcTechFile \
-layer_mapfile contactmapfile /input.emir0_bin

print_em_report \
-net all_nets \
-type acrms_sh \
-filename ./acrms_sh.txt

print_thermal_report \
-type mos-region \
```


© 2020 All Rights Reserved.

```
-tiles 15x10 \
-filename ./mos-region-tiles.txt
```

Related Topic

load_view

analyze_esd_generic

```
analyze_esd_generic
     -power_nets {pwr_net_names}
     -ground_nets {gnd_net_names}
     -signal_nets {signal_net_names}
     [-pgpin2clamp_threshold threshold_value]
     [-loop_threshold threshold_value]
     -target_models {model_name_1 ron_1 model_name_2 ron_2…}
     -dspf_file filename
     -tech_file file_name
     -output_dir dir_name
     [-discard_pin_to_first_ESD_for_signal_net {true | false}]
     [-esd_instance_name_string {inst_name1 inst_name2 ...}]
     [-merge_fingers {true | false}]
     [-cdl_file file_name]
```

Reports the effective resistance from the power or ground pin (bump) or the signal net to the nearest ESD cell in Voltus-Fi-XL. The ESD cell is a device used for ESD protection and is of two types: clamp and transistor. The command checks the effective resistance and generates reports for the following:

- **Pin-to-clamp:** Effective resistance from a pin to all the clamps or ESD devices.
- **Clamp-to-clamp:** Effective resistance is computed between all clamps for each net.
- **Pin-to-pin:** Effective resistance from a power pin to the ESD device to the ground pin within the specified threshold value.

This helps you identify bumps with effective resistance higher than the specified threshold value.

© 2020 All Rights Reserved.

Parameters

`-cdl_file` *`file_name`*	Specifies the name of the schematic CDL netlist. This is an optional parameter. If this netlist is specified, Voltus-Fi assumes that the extracted DSPF file is ESD-cells flat.
`-dspf_file` *`filename`*	Specifies the name of the DSPF file that is generated using Quantus. This file contains information about model names including the ESD model names required for ESD analysis. This is a required parameter. **Note:** Voltus-Fi supports reading of the DSPF file in the gzip format.
`-discard_pin_to_first_ESD_for_signal_net {true | false}`	
	Specifies whether or not the effective resistance from a pin to the first ESD device should be ignored for computing the pass/fail status of nets. This is an optional parameter. By default, this parameter is set to `false`, which means the effective resistance from a pin to the first ESD device is considered while generating the ESD analysis report.
`-esd_instance_name_string {`*`inst_name1 inst_name2 ...`*`}`	
	Specifies the names of the instances to be considered as ESD instances. For example, `-esd_instance_name_string "clamp ESD"` In the above example, only instances with names including the words, 'clamp' or 'ESD' are considered as ESD instances. This is an optional parameter. *Default*: `""` (empty string)
`-ground_nets {`*`gnd_net_names`*`}`	Specifies the names of the ground nets for which the ESD check is to be performed. This is a required parameter.

© 2020 All Rights Reserved.

`-loop_threshold` *`value`*	Specifies the threshold value for the effective resistance from the power pin to the ground pin that will be considered for generating a pass/fail report. If the computed effective resistance is less than the specified threshold value, a “pass” result is reported. Otherwise, a “fail” result is reported. This is an optional parameter. By default, the software considers `0` as the loop threshold value.
`-merge_fingers {true | false}`	
	Specifies whether to merge fingers for resistance checks. This is an optional parameter. *Default*: `true`
`-output_dir` *`output_dir_name`*	
	Specifies the name of the output directory in which the reports generated for the ESD analysis are written. This is a required parameter.
`-pgpin2clamp_threshold` *`value`*	Specifies the threshold value for effective resistance from the power pin to the clamp (ESD device) that will be considered for generating a pass/fail report. If the computed effective resistance is less than the specified threshold value, a “pass” result is reported. Otherwise, a “fail” result is reported. This is an optional parameter. By default, the software considers `0` as the pin-to-clamp threshold value.
`-power_nets {`*`pwr_net_names`*`}`	Specifies the names of the power nets for which the ESD check is to be performed. This is a required parameter.
`-signal_nets {`*`signal_net_names`*`}`	Specifies the names of the signal nets for which the ESD check is to be performed. This is a required parameter.
`-target_models {model_name_1 ron_1 model_name_2 ron_2…}`	

© 2020 All Rights Reserved.

	Specifies the ESD model names that are used in the DSPF netlist. Typically, the target models are of two types, diode and ESD MOS device. Each model is accompanied with the `RON` (always-on resistance) value for the model. This is a required parameter.
`-tech_file` *`file_name`*	
	Specifies the technology file that contains the EM limits for ESD calculations. This is a required parameter.

Example

The following command performs the effective resistance check for the specified power, ground, and signal nets for the specified ESD models, using the specified threshold value to generate the pass/fail reports for the ESD analysis:

```
analyze_esd_generic
    -power_nets {VDD}
    -ground_nets {VSS}
    -signal_nets {PAD_1 PAD_2}
    -tech_file qrcTechFile
    -pgpin2clamp_threshold 0.07
    -target_models {esd_mos 23.45 esd_diode 45.9}
    -dspf_file CD_TESTCASE.spice
    -output_dir outputdir/
```

Related Topics

analyze_esd_current_density, analyze_esd_network, esd_optimizer, setMultiCpuUsage, setDistributeHost

analyze_esd_current_density

```
analyze_esd_current_density
     -esd_conf_file conf_filename
     -run_async {true | false}
     -simulation_mode {DC CDM HBM}
     -zap_injection_current_polarity {positive | negative}
```


© 2020 All Rights Reserved.

```
-zap_injection_pins list_of_pins
-zap_injection_reference_pins list_of_pins
-use_remote_machines_as_many_zaps {true | false}
```

This command performs the ESD Current Density checks in Voltus-Fi-XL.

Note: To run this command, the configuration file is required to be specified by using the `-esd_conf_file` parameter. This configuration file includes information about all the other parameters needed to run this command. The configuration file is saved in the GUI.

© 2020 All Rights Reserved.

Parameters

`-esd_conf_file` *`conf_filename`*	
	Specifies the name of the configuration file that was saved in GUI. **Note**: This file includes information about the other parameters required for running this command. For details, see Running Current Density Checks. This is a required parameter.
`-run_async {true | false}`	When set to `true`, the tool runs distributed simulation (zaps) in an asynchronous mode. In the asynchronous mode, connections are established to remote machines. But Voltus-Fi does not wait for all connections to be established before starting the jobs on the connected machines. *Default*: `false` By default, Voltus-Fi runs ESD simulations in a synchronous mode. This means that all remote-host connections are established before running the jobs on the remote hosts.
`-simulation_mode {DC CDM HBM}`	Specifies the mode to be used for ESD event analysis for Current Density checks. The supported modes are: ■ *DC*: Current density check for static simulation mode ■ *CDM Mode*: Current density check for charged device model (CDM) stimulus mode ■ *HBM Mode*: Current density check for human body model (HBM) stimulus mode Multiple modes can be specified separated by a space. This is a required parameter.
`-zap_injection_current_polarity {positive | negative}`	
	Specifies the the polarity of the injection current, positive or negative in the zap combination to be simulated by Voltus-Fi-ESD. For details, see Using the Zap Parameters.
`-zap_injection_pins` *`list_of_pins`*	

© 2020 All Rights Reserved.

	Specifies the names of the static current injection pins in the zap combination to be simulated by Voltus-Fi-ESD. For details, see Using the Zap Parameters.
`-zap_injection_reference_pins` *`list_of_pins`*	
	Specifies the name of the reference ground pin in the zap combination to be simulated by Voltus-Fi-ESD. For details, see Using the Zap Parameters.
`-use_remote_machines_as_many_zaps {true | false}`	
	When this parameter is set to `true`, Voltus-Fi uses as many remote hosts as the number of zaps. This parameter overrides the `setMultiCpuUsage -remoteHost` parameter setting. For example, if there are ten zaps to be simulated, Voltus-Fi will try to make distributed connections in LSF to ten machines. The simulation will be run on the machines as and when they are acquired from the LSF. *Default*: `false` **Note**: Using this parameter does not guarantee that all zap simulations will run on exclusive machines. This is because if some simulations are completed before connections are established to other machines, then the pending jobs will start on the machines that are already connected.

Example

The following command performs the Current Density checks using the ESD configuration file, `ESD.conf`:

```
analyze_esd_current_density
    -esd_conf_file ESD.conf \
    -simulation_mode CDM \
    -zap_injection_pins {VDD} \
    -zap_injection_reference_pins {VSS} \
    -zap_injection_current_polarity {positive} \
    -run_async false \
    -use_remote_machines_as_many_zaps false
```


© 2020 All Rights Reserved.

Using the Zap Parameters

The `-zap_injection_current_polarity`, `-zap_injection_pins`, and the `-zap_reference_pins` parameters of the `analyze_esd_current_density` command can be used to control the zap combinations that are to be simulated by Voltus-Fi-ESD. If these parameters are not specified, all the zap combinations will be simulated and results will be generated.

Specific keywords can also be specified for the options, `-zap_injection_pins` and `-zap_reference_pins` so that all the power/ground/signal pins can be specified using one keyword. A description of the keywords is provided below:

- `ALL_POWER_PINS`: This keyword specifies that all power pins will be chosen.
- `ALL_GROUND_PINS`: This keyword specifies that all ground pins will be chosen.
- `ALL_SIGNAL_PINS`: This keyword specifies that all signal pins will be chosen.

An example of how these parameters can be used is provided below. In this example, the simulation for zaps will be run with the injection pin, `ALL_POWER_PINS` and the reference pin, `DVSS`, for all positive zaps.

```
analyze_esd_current_density
    -esd_conf_file ESD.conf
    -zap_injection_pins {ALL_POWER_PINS}
    -zap_injection_reference_pins {DVSS}
    -zap_injection_polarity {positive}
```

Related Topics

analyze_esd_generic, analyze_esd_network, esd_optimizer, setMultiCpuUsage, setDistributeHost

analyze_esd_network

```
analyze_esd_network
     -power_nets {pwr_net_names}
     -ground_nets {gnd_net_names}
     -signal_nets {signal_net_names}
     -target_models {model_name_1 ron_1 model_name_2 ron_2…}
     -dspf_file filename
     -output_dir dir_name
     -merge_fingers {true | false}]
     -tech_file file_name
     -discard_pin_to_first_ESD_for_signal_net {true | false}
```


© 2020 All Rights Reserved.

```
-rule_file file_name
-model_file file_name
-dfIIlayermap_file file_name
```

This command performs rule-based resistance and current density checks in Voltus-Fi-XL. The ESD checks are defined as a set of rules in a single rule file. Voltus-Fi-XL gets information of all the nets and clamps based on the inputs provided in the *ESD Common Setup* section of the ESD form, and runs resistance and current density checks in the background for the rules specified in the rule file. The tool generates a pass/fail summary report and a detailed report for each rule. In this flow, extraction is run only once for all methods, thereby saving time and memory.

This command performs the following types of rule-based ESD checks:

- **Pin-to-clamp:** Effective resistance is computed from a pin to all the clamps or ESD devices.
- **Clamp-to-clamp:** Effective resistance is computed between all clamps for each net.
- **EM:** Current density checks are performed for the DC Simulation model using current value as the input stimulus.

© 2020 All Rights Reserved.

Parameters

`-dfIIlayermap_file` *`filename`*	
	Specifies the name of the DFII layer map file that provides mapping between the layer names in the DSPF file and the technology file.
`-discard_pin_to_first_ESD_for_signal_net {true | false}`	
	Specifies whether or not the effective resistance from a pin to the first ESD device should be ignored for computing the pass/fail status of signal nets. This is an optional parameter. By default, this parameter is set to `false`, which means the effective resistance from a pin to the first ESD device is considered while generating the ESD analysis report.
`-dspf_file` *`filename`*	
	Specifies the name of the DSPF file that is generated using Quantus. This file contains information about model names including the ESD model names required for ESD analysis. This is a required parameter. **Note:** Voltus-Fi supports reading of the DSPF file in the gzip format.
`-ground_nets {`*`gnd_net_names`*`}`	
	Specifies the names of the ground nets for which the ESD check is to be performed. This is a required parameter.
`-merge_fingers {true | false}`	
	Specifies whether to merge fingers for resistance checks. This is an optional parameter. *Default*: `true`
`-model_file` *`filename`*	
	Specifies the name of the Spice model file that contains references to all the Spice models required to simulate the DSPF circuit.

© 2020 All Rights Reserved.

`-output_dir` *`output_dir_name`*	
	Specifies the name of the output directory in which the reports generated for the ESD analysis are written. This is a required parameter.
`-power_nets {`*`pwr_net_names`*`}`	
	Specifies the names of the power nets for which the ESD check is to be performed. This is a required parameter.
`-rule_file` *`filename`*	
	Specifies the name of the rule file that contains rules for computing effective resistance and performing current density analysis.
`-signal_nets {`*`signal_net_names`*`}`	
	Specifies the names of the signal nets for which the ESD check is to be performed. This is a required parameter.
`-target_models {`*`model_name_1 ron_1 model_name_2 ron_2…`*`}`	
	Specifies the ESD model names that are used in the DSPF netlist. Typically, the target models are of two types, diode and ESD MOS device. Each model is accompanied with the `RON` (always-on resistance) value for the model. This is a required parameter.
`-tech_file` *`file_name`*	
	Specifies the technology file that contains the EM limits for ESD calculations. This is a required parameter.

Example

The following command performs rule-based ESD checks for the specified power, ground, and signal nets for the specified ESD models, using the rules defined in the `rule.txt` file to generate the pass/fail reports for the ESD analysis:

© 2020 All Rights Reserved.

```
analyze_esd_network
    -power_nets {VDD}
    -ground_nets {VSS}
    -signal_nets {PAD_1 PAD_2}
    -tech_file qrcTechFile
    -rule_file rule.txt
    -target_models {esd_mos 23.45 esd_diode 45.9}
    -dspf_file CD_TESTCASE.spice
    -output_dir outputdir
```

Related Topics

analyze_esd_generic, analyze_esd_current_density, esd_optimizer, setMultiCpuUsage, setDistributeHost

esd_optimizer

```
esd_optimizer
     -esd_cells {cell_name1 cell_name2 ...}
     [-cdl_file file_name]
     [-cdl_cells {cell_name1 cell_name2 ...}]
     -dspf_file filename
     -output_dir dir_name
```

This command generates an optimized xDSPF file in Voltus-Fi-XL. It uses the specified full-chip flat xDSPF and optimizes it for faster Spectre simulation. The optimized xDSPF file consists of all ESD protection devices and nets, parasitics, and all other components that are related to the ESD discharge paths for all input net ports/pins.

When this command is run, ESD Analysis automatically runs the SPF Checker utility on the optimized xDSPF file to ensure that the DSPF file is compatible with the simulation requirements.

© 2020 All Rights Reserved.

Parameters

`-cdl_file` *`file_name`*	Specifies the name of the schematic CDL netlist. This is an optional parameter. If this netlist is specified, Voltus-Fi assumes that the extracted DSPF file is ESD-cells flat.
`-cdl_cells {`*`cell_name1 cell_name2 ...`*`}`	Specifies the ESD CDL cells. This is an optional parameter.
`-esd_cells {`*`cell_name1 cell_name2 ...`*`}`	Specifies the ESD protection clamp cells that are used in the DSPF netlist. This is a required parameter.
`-dspf_file` *`filename`*	Specifies the name of the full-chip flat DSPF file that is generated using Quantus. This file contains information about model names including the ESD model names required for ESD analysis. This is a required parameter. Voltus-Fi supports reading of the DSPF file in the gzip format.
`-output_dir` *`dir_name`*	Specifies the name of the output directory in which the optimized xDSPF file is written. This is a required parameter.

Example

The following command generates the optimized xDSPF file in the `CD_CDM_Simulation` directory:

```
esd_optimizer
    -esd_cells {g45pd2svt g45n2nvt} \
    -dspf_file pvs_qrc_run/vfi_quantus_block_esd/DiffOpAmp.dspf \
    -output_dir CD_CDM_Simulation
```

Related Topics

analyze_esd_generic, analyze_esd_current_density, analyze_esd_network, setMultiCpuUsage, setDistributeHost

© 2020 All Rights Reserved.

setMultiCpuUsage

```
setMultiCpuUsage
     [-acquireLicense integer]
     [-keepLicense {true | false}]
     [-licenseList {string}]
     [-localCpu { integer | max}]
     [-remoteHost integer]
     [-cpuPerRemoteHost integer]
     [-releaseLicense]
     [-threadInfo {0 | 1 | 2}]
     [-verbose]
     [-cpuAutoAdjust {true | false}]
     [-autoPageFaultMonitor {0 | 1 | 2 | 3}]
     [-reset]
     [-help]
```

Specifies the number of threads to use for multi-threading, or the maximum number of computers to use for distributed processing, or the maximum number of computers and the number of threads to use for super-threading. Optionally, this command reports the usage information.

© 2020 All Rights Reserved.

Parameters

`-help`	Outputs the command usage and a brief description about the command parameters.
`-acquireLicense` *`integer`*	
	Acquires licenses to enable the specified number of CPUs. For example, if you specify `-acquireLicense 5`, the software checks out enough licenses to enable five CPUs. After the licenses are checked out, they can be used for multi-threading, distributed processing, or super-threading.
`-autoPageFaultMonitor {0 | 1 | 2 | 3}`	
	Specifies to set a warning rank for performance issues caused by major page faults. The parameter supports the following values: 0 - specifies to disable the reporting of major page faults. 1 - specifies to set the warning rank as 1. This means PF/total PF > 0.8, b > 10, and wa > 85% 2 - specifies to set the warning rank as 2. This means PF/total PF > 0.5, b > 5, and wa > 65% 3 - specifies to set the warning rank as 3. This means PF/total PF > 0.2, b > 1, and wa > 40% where PF is the number of major page faults from the software, total PF is the total number of major page faults in the machine, b is the number of blocked processors, and wa is CPU waiting time.
`-cpuAutoAdjust {true | false}`	
	Specifies that the specified number of CPUs should be adjusted based on the available free memory. The software lowers the number of client processes on the local machine in case of insufficient memory. This is applicable to CPUs only on the local machine. The default value is `true`. If set to `false`, automatic adjustment of CPUs is disabled.
`-cpuPerRemoteHost` *`integer`*	

© 2020 All Rights Reserved.

	Specifies the number of CPUs used by clients for multi-threading. **Note**: For super-threading, you must use this parameter in conjunction with the `-remoteHost` parameter. *Default*: `1` **Note** : Save/restore is enabled for this option.
`-keepLicense {true | false}`	
	Specifies whether to keep the acquired multiple-CPU licenses until the current session ends. Specify this parameter before running any commands that require multiple-CPU applications. To release all multiple-CPU licenses immediately, use the `-releaseLicense` option. *Default*: `true`
`-licenseList {`*`tpsmp`*`}`	
	Specifies the list or order of licenses that the software should use for checking out licenses during multiple-cpu processing. The parameter does not support an empty license list, `{}`. This parameter allows the `tpsmp` option.
`-localCpu` *`string`*	Specifies the number of CPUs on the local machine. This parameter is required for multi-threading. *Default*: 1 **Note**: Save/restore is enabled for this option.
`-releaseLicense`	Releases all the multiple-CPU license(s) immediately. By default, the software holds multiple-CPU licenses until the end of the current session. To specify that the software release multiple-CPU licenses after every multiple-CPU command runs, use the `-keepLicense` parameter.
`-reset`	Resets the specified parameters back to the default values.
`-threadInfo {0 | 1 | 2}`	

© 2020 All Rights Reserved.

	Reports the usage information.
	
	Default: 0
	
	Specify one of the following values:
	
	0 : Does not write messages to the log file.
	
	1 : Writes the final message to the log file.
	
	For example,
	
	All threaded jobs finished (10 elapsed sec: 0 processor sec (parent), 0 system sec (parent), 0 processor sec (threads), 0 system sec (threads)).
	
	2 : Writes additional starting/ending information for each thread.
	
	For example,
	
	`Starting threaded job 1...`
	
	`Starting threaded job 2...`
	
	`Starting threaded job 3...`
	
	`Ending threaded job 2 (1 elapsed sec, 0 processor sec, 0 system sec, 1.160M).`
	
	`Ending threaded job 3 (2 elapsed sec, 0 processor sec, 0 system sec, 0.895M).`
	
	`Ending threaded job 1 (10 elapsed sec, 0 processor sec, 0 system sec, 1.172M).`
	
	`All threaded jobs finished (10 elapsed sec: 0 processor sec (parent), 0 system sec (parent), 0 processor sec (threads), 0 system sec (threads)).`

`-verbose`	Displays messages when changing the multiple-CPU settings.

Related Topics

analyze_esd_generic, analyze_esd_current_density, analyze_esd_network, esd_optimizer, setDistributeHost

© 2020 All Rights Reserved.

setDistributeHost

```
setDistributeHost
     [-help]
     [-add string]
     [-args string]
     [-custom]
     [-custom_script string]
     [-custom_script_list string]
     [-local]
     [-lsf]
     [-nc]
     [-noWaitTaskComeUp]
     [-queue string]
     [-remove string]
     [-reportLsfInfo]
     [-resource string]
     [-rsh]
     [-sge]
     [-shellTimeout integer]
     [-single_cpu_lsf_args string]
     [-ssh]
     [-timeOut integer]
     [-env_script "script"]
```

Specifies the multiple-CPU processing configuration for distributed processing or super-threading. Voltus-Fi-XL finds hosts from the last `setDistributeHost` command specified before a distributed processing or super-threading command starts.

© 2020 All Rights Reserved.

Parameters

`-help`	Outputs the command usage and a brief description about the command parameters.
`-add` *`string`*	Adds the specified hosts to the `rsh/ssh` configuration. If you specify more than one host, separate each host name with a space. Each time you specify a host, the software uses it to run one client process. To run more than one client process on a host, specify it more than once. For example, if you have two machines that each has two CPUs and you want to use both CPUs in both machines, type a command like the following: `setDistributeHost -rsh -add {machine1 machine2 \ machine1 machine2}`
`-args` *`string`*	Specifies any additional arguments you need to provide to the Sun Grid Engine (SGE) or Load Sharing Facility (LSF) job submission command.
`-custom`	Use a custom job submission script for distributed processing.
`-custom_script` *`string`*	
	Specifies a custom script to run distributed processing.
`-custom_script_list` *`string`*	
	Specifies that the launch of multiple jobs should be customized, based on the resources required for these jobs. This parameter lets you specify multiple launch strings in order of priority with a maximum allowed job count for each string. If you request for N jobs, the command starts submitting jobs starting from string1, until the corresponding count is reached, at which point it starts submitting the next string in the list. Therefore, the command submits the first "`count1`" jobs with string1 , the next "`count2`" jobs with string2 and so on. It stops when the total submitted jobs has reached the requested number (N). **Note** : This parameter can only be specified with the `-custom` parameter.

© 2020 All Rights Reserved.

`-env_script "script"`	Specifies a user-defined shell environment script that is run for each of the slave hosts. The parameter allows you to pass these environment settings to the slave hosts by including them in the shell environment script. The license variables used by the software are also automatically passed to the slave hosts to avoid license errors.
`-local`	Runs all the distributed processing jobs on the master machine.
`-lsf`	Specifies a LSF configuration. If you do not specify any parameters for `-lsf`, the software uses the default settings. **Note**: To use this parameter, LSF must already be set up (typically by specifying the `LSF_ENVDIR` and `LSF_SERVERDIR` environment variables) and `bsub` must be in your search path. Contact your LSF administrator for details.
`-nc`	Uses NC commands (Network Computer from RTDA) for distributed processing.
`-noWaitTaskComeUp`	When specified, does not allow the software to wait for tasks that are delayed or are not likely to occur.
`-queue string`	Specifies the queue for the LSF or Sun Grid Engine (SGE) configuration.
`-remove string`	Removes the specified hosts or all hosts from the `rsh/ssh` configuration. If you specify more than one host, separate each host name with a space.

© 2020 All Rights Reserved.

`-reportLsfInfo`	Specifies to report LSF information in a single report. This parameter allows you to collect the system statistics on the status of the distributed jobs. It has the following information: Host name for master and LSF tasks Number of CPUs on each machine that are being used by a process Job ID/PID for each task process Total memory used on each machine Total virtual memory used on each machine Total CPU time used on each machine You must specify this parameter along with the `-lsf` parameter. To get the LSF information, specify the `getDistributeHost -reportLsfInfo` command parameter. You must specify the `getDistributeHost -reportLsfInfo` only after the LSF tasks exit to obtain the task CPU time. If you specify this command before the task exits, the command gives a warning message.
`-resource` *`string`*	Specifies a resource string for the SGE or LSF queue. **Note**: The correct resource string is specific to your installation. Contact your LSF administrator for the appropriate parameters and values.
`-rsh`	Specifies a remote shell configuration.
`-shellTimeout` *`integer`*	Specifies the number of seconds the host machine waits for `rsh/ssh` machines to become available for multiple-CPU processing. *Default*: 5
`-sge`	Specifies an SGE configuration.
`-single_cpu_lsf_args` *`string`*	
	Specifies the `bsub` options. **Note**: You must use this parameter to adjust the LSF argument for a single CPU application. Alternatively, use the `-args` parameter to adjust the LSF argument for an application that runs in the super-threading mode.

© 2020 All Rights Reserved.

`-ssh`	Specifies a secure shell configuration.
`-timeout` *`integer`*	Specifies the default timeout for a submitted job. This functionality prevents the master machine from hanging when the clients do not come up for some reason. After timeout, a warning message appears informing you that the distributed CPUs did not come online within the expected time. The master will continue waiting for the clients and it is up to you to continue waiting or stop the run if there are problems with the Distributed Resource Management (DRM) system. A sample warning message is given below: Task id 2 did not come up within the specified timeout period (300 seconds). Use `setDistributeHost -timeout` to change the value. This may be a real error due to incorrect or incomplete platform/machine specification provided via setDistributeHost to your distributed resource management system (LSF/Sun Grid Engine etc.). If running locally (`setDistributeHost -local`), the current machine may be overloaded with too many jobs and swapping (check free memory and CPUs with top). For local runs, consider reducing the number of CPUs you are using via the `setMultiCpuUsage` command, or free up some memory by killing other memory intensive processes. *Default*: `30 (local/rsh/custom) and 3600 (lsf/sge)`

Related Topics

analyze_esd_generic, analyze_esd_current_density, analyze_esd_network, esd_optimizer, setMultiCpuUsage

create_pgv

```
create_pgv
     -macro_name macro_name
     [-lib_name name_of_library]
     [-libgen_cmd_file cmd_file_name]
     [-libgen_path libgen_binary_path]
     -tech_file tech_file_name
     [-tech_lef_file tech_lef_file_name]
     -type {static | dynamic}
```


© 2020 All Rights Reserved.

```
[-macro_lef_file macro_lef_file_name]
-sim_dir simulation_dir
[-lef_layer_map lef_layer_map_file_name]
[-oa_lib list_of_OA_libraries]
[-pgdb_layer_map pgdb_layermap_file_name]
[-pgdb_version version_number]
[-trigger_file trigger_file_name]
[-output_directory directory]
[-uti_Ron_file Ron_report_file_dumped_by_Spectre]
[-create_no_device_nets true | false]
[-powergate_state_flow true | false]
[-use_pwl_files true | false]
[-map_connectivity_layers true | false]
[-units value]
```

Creates power-grid views in Voltus-Fi-XL. The power-grid views contain information about the power ports of a transistor or block, the internal power grid, and the tap current distribution within a transistor or block.

Note: The `create_pgv` command has the following parameters that are used only for hierarchical EMIR analysis.

```
-hier_flow {true | false}
-hier_flow_stop_layer layer_name
-disable_signal_pgdb {true | false}
-stop_on_missing_lef_pins {true | false}
-sim_dir simulation_dir
```

Parameters

© 2020 All Rights Reserved.

`-macro_name` *`macro_name`*	
	Specifies the name of the macro. This is a required parameter.
`-lib_name` *`name_of_library`*	
	Specifies the name of the library. This is specified if you want to create a library with a name different than that of the macro. This is an optional parameter. By default, a library is created with the same name as that of the macro.
`-libgen_cmd_file` *`cmd_file_name`*	

	Specifies the name of the LibGen command file that includes LibGen commands, if any, required to be specified for creating PGVs.
	For example, the LibGen command file may contain the following:
	`setvar source_drain_equivalent false`
	`setvar dspf_vialayer_bounding_box 10`
	This is an optional parameter.

`-libgen_path` *`libgen_binary_path`*	
	Specifies the location of the LibGen library. This is used only to specify a library outside the IC hierarchy. By default, the software uses the library available in the IC hierarchy. This is an optional parameter. For example, `-libgen_path /tools.lnx86/ssvtools/bin/voltus_libgen \`
`-map_connectivity_layers true | false`	
	Controls the handling of `*wires` layers. When set to `true`, the `*wires` layer resistors are assigned to the layers of its connecting nodes. When set to `false`, the `*wires` layer resistors are assigned to the layers specified in the PGDB layer map file. *Default*: `true`
`-tech_file` *`tech_file_name`*	

© 2020 All Rights Reserved.

	Specifies the technology file to be used for reading the technology data. This is a required parameter.
`-tech_lef_file` *`tech_lef_file_name`*	
	Specifies the LEF information for the technology data. This is an optional parameter. This parameter is not specified when the OA libraries are specified.
`-type {static | dynamic}`	
	Specifies the type of analysis for which you want to generate power-grid views. This is a required parameter.
`-macro_lef_file` *`macro_lef_file_name`*	
	Specifies the LEF information for the design. **Note**: If the technology data and design data are in the same LEF file, use any of the two options, `-tech_lef_file` or `-macro_lef_file` to specify the filename. In this case, it is an optional parameter.
`-sim_dir` *`simulation_dir`*	
	Specifies the simulation directory for the current data. This is a required parameter.
`-lef_layer_map` *`lef_layer_map_file_name`*	

	Specifies the mapping between LEF layer names and technology layer names.
	
	The syntax of the layer map file is as follows:
	
	`#type  technology_layer_name  lefdef  lef_layer_name`
	
	Example:
	
	`# ----  ----------  -----  ----------`
	`metal  METAL_1  lefdef  M1`
	`via  VIA_1  lefdef  VIA1`
	`metal  METAL_2  lefdef  M2`
	`via  VIA_2  lefdef  VIA2`
	
	where “type” can be `poly/via/metal/diff`.
	
	Note: This is an optional parameter. If not specified, the software automatically generates the layer map file.

`-oa_lib` *`list_of_libraries`*	

© 2020 All Rights Reserved.

	Specifies the OpenAccess (OA) database library name for PGV generation. When this option is specified, the layer map file is automatically generated. However, you can also specify the layer map file by using the `-lef_layer_map` parameter. **Note**: The parameters, `-tech_lef_file` /-`macro_lef_file` and `-oa_lib` are mutually exclusive. This is an optional parameter.
`-pgdb_layer_map` *`pgdb_layermap_file_name`*	

	Specifies the mapping between the xDSPF file layer names, written in the simulation database, and the technology layer names.
	
	The syntax of the layer map file is as follows:
	
	`# type  technology_layer_name  pgdb  xDSPF_layer_name`
	
	Example:
	
	`# ----  ---------- ---- ----------`
	`  poly  POLYCIDE  pgdb  poly`
	`  via  CONT  pgdb  pl1co`
	`  via  CONT  pgdb  odCont1`
	`  via  CONT  pgdb  odCont2`
	`  metal  METAL_1  pgdb  metal1`
	`  via  VIA_1  pgdb  VIA1`
	`  metal  METAL_2  pgdb  metal2`
	`  via  VIA_2  pgdb  VIA2`
	`  diff  OD  pgdb  nwires`
	`  diff  OD  pgdb  mwires`
	
	where “type” can be `poly/via/metal/diff`.
	
	Note: Ensure that all the xDSPF file layer names are mapped to the technology layer names. You can map multiple xDSPF file layer names to a single technology layer name. This is shown in the above example.
	
	Note: This is an optional parameter. If not specified, the software automatically generates the layer map file.

`-pgdb_version` *`version_number`*	

© 2020 All Rights Reserved.

	Specifies the PGDB version. It is only used when you want to use an older PGDB version instead of the default version, which is the latest available version. This is an optional parameter. For example, `-pgdb_version 9.1` *Default*: The latest version
`-trigger_file` *`trigger_file_name`*	
	Specifies the trigger file that is used during dynamic current simulation to run a detailed analysis for determining current distribution. This is an optional parameter.
`-output_directory` *`directory`*	
	Specifies the name of the output directory for power-grid view generation. This is an optional parameter. By default, the power-grid views are stored in the work directory.
`-uti_Ron_file` *`Ron_report_file_dumped_by_Spectre`*	
	Specifies the `UTI-RON` file that contains multiple on-resistance (Ron) values, that is minimum, maximum, and average `(min/avg/max)` values, for powergates. This is an optional parameter. If this file is not specified, Voltus-Fi-XL uses the `RON` values from the MMSIM simulation database (`*.emir#_bin` file).
`-create_no_device_nets true | false`	
	When set to `true`, creates power-grid views even when the net has no current files. This is used to capture the nets with only physical data (RC grid) and no electrical data (taps). This is an optional parameter. *Default*: `false`
`-powergate_state_flow true | false`	

© 2020 All Rights Reserved.

	When set to `true`, searches for power gate state files (`*pwg_state)` and passes them to LibGen. The power gate state files are searched in the UTI Ron directory in which the UTI Ron file is specified. If the UTI Ron file or directory is not specified, then this option does not have any effect even when set to `true`.
	
	Examples:
	
	`create_pgv`
	`-uti_ron_dir simout/output.pg.state/xps.raw/emir_uti.raw/`
	`-powergate_state_flow true`
	
	In the above example, the power gate state files are searched in the `emir_uti.raw` directory.
	
	`create_pgv`
	`-uti_ron_file simout/output.pg.state/xps.raw/xps.emirtap.pwg_ron`
	`-powergate_state_flow true`
	
	In the above example, the power gate state files are searched in the `xps.raw` directory in which the UTI Ron file, `xps.emirtap.pwg_ron`, has been specified.
	
	Power gate state files are generated by Spectre only when the following option is set:
	
	`eisopt report_pwg_state=yes`

`-use_pwl_files true | false`	
	When set to `true`, searches for net-specific PWL files instead of current (`pti`) files in the simulation directory and passes them to LibGen through the `*.vhdr` files. The default value is `false`.

`-units` *`value`*	Specifies the value of a variable used in PGV generation in database units per micron.
	
	Default: `1000`
	
	Example:
	
	`create_pgv`
	`-units 10000`
	
	In the above example, the value of the variables used in PGV generation is set to 10000 database units per micron.

© 2020 All Rights Reserved.

Example

The following command is used to create power-grid views:

```
create_pgv \
     -macro_name design1 \
     -lib_name ether \
     -tech_file RCgen.tch \
     -type dynamic \
     -tech_lef_file tech.lef \
     -macro_lef_file mac.lef \
     -sim_dir pgv-sim-132
     -lef_layer_map lefdef.layermap.txt \
     -pgdb_layer_map pgdpmap.txt \
     -trigger_file trigger_vps.txt \
     -output_directory ./vps_pgv_1
```

Related Topics

- The Hierarchical EMIR Analysis Batch Mode Flow
- set_mmsim_pgv_nets

update_pgv

```
update_pgv
     -library library_name
     -trigger_file trigger_file_name
     [-library_prefix prefix]
     [-output_directory directory]
     [-libgen_path libgen_binary_path]
     [-libgen_cmd_file cmd_file_name]
     [-force]
```

Updates a power-grid view (PGV) incrementally.

Parameters

© 2020 All Rights Reserved.

`-library` *`library_name`*	
	Specifies the name of the library that contains the old PGV.
`-trigger_file` *`trigger_file_name`*	
	Specifies the trigger file to update the old PGV.
`-library_prefix` *`prefix`*	
	Specifies the prefix for the generated PGV library (*`prefix`*`.cl`).
`-output_directory` *`directory`*	
	Specifies the name of the output directory in which the merged PGV is created.
`-libgen_path` *`libgen_binary_path`*	
	Specifies the location of the LibGen library. This is used only to specify a library outside the IC hierarchy. By default, Voltus-Fi uses the library available in the IC hierarchy. For example, `-libgen_path /tools.lnx86/ssvtools/bin/voltus_libgen \`
`-libgen_cmd_file` *`cmd_file_name`*	

	Specifies the name of the LibGen command file that includes LibGen commands, if any, required to be specified for creating PGVs.
	For example, the LibGen command file may contain the following commands:
	`setvar source_drain_equivalent false`
	`setvar dspf_vialayer_bounding_box 10`

`-force`	Updates the report forcefully even if the taps or current sources of every net do not match the existing PGV.

Example

The following command updates the PGVs contained in the `ether` library. The merged PGV will be created in the `vps_pgv_2` directory:

```
update_pgv \
     -library ether \
     -trigger_file trigger_vps.txt \
     -output_directory ./vps_pgv_2
```


© 2020 All Rights Reserved.

Related Topics

- The Hierarchical EMIR Analysis Batch Mode Flow
- set_mmsim_pgv_nets

set_mmsim_pgv_nets

```
set_mmsim_pgv_nets
     -net net_name
     -voltage voltage_value
     -voltage_source_nets voltage_source_only_nets
     -results_file result_file_name
     [-prefix prefix_name]
     [-feedthru_nets {net1 net2 net#}]
```

Specifies the nets for which you want to create power-grid views (PGVs) in the MMSIM flow. You can use this command to specify multiple PGV nets at a time.

© 2020 All Rights Reserved.

Parameters

`-net` *`net_name`*	Specifies the names of nets for which you want to create PGVs.
`-voltage` *`voltage_value`*	
	Specifies the voltage of the net.
`-voltage_source_nets` *`voltage_source_only_nets`*	
	Specifies the names of nets that are connected only to voltage sources and for which a current file, with a `.pti` extension, is not available.
`-results_file` *`result_file_name`*	
	Specifies the simulation result file to be used. The naming convention of the simulation result file is, `*.emir#_bin`.
`-prefix` *`prefix_name`*	
	Specifies the name of the prefix used while performing the simulation.
`-feedthru_nets {net1 net2 net#}`	Specifies the feedthrough or dummy power nets for the PGV generation. When specified, generates EM views for the specified feedthrough/dummy nets in the PGV flow. **Note:** EM views are generated for the specified feedthrough/dummy nets even if there are no current (`pti`) files or power-grid data for these nets from Spectre simulation.

Example

The following command is used to create power-grid views for nets `VSS` and `VDD`, voltage source net, `CEN`, and feedthrough net, `WEN`:

```
set_mmsim_pgv_nets \
-net {VDD VSS CEN} \
-voltage {1.1 0 1.1} \
-results_file /emirtap.emir0_bin \
-voltage_source_nets {CEN}
-prefix XO.
-feedthru_nets {WEN}
```


© 2020 All Rights Reserved.

Related Topics

- The Hierarchical EMIR Analysis Batch Mode Flow
- create_pgv

report_vfi_XL_ron

```
report_vfi_XL_ron
     -pin_pair_file file_name
     -output_file file_name
```

Generates the layer-wise on-resistance (RON) report for IR drop and EM analysis for the specified pin-pairs in Voltus-Fi-XL.

Parameters

`-output_file` *`file_name`*	Specifies the name of the output file to be created with the RON report. This is a required parameter.
`-pin_pair_file` *`file_name`*	Specifies the name of the pin-pair file that contains information about all the pin pairs for which the RON report is to be generated. The pins in each pair are separated by a space. Multiple pin-pairs can be specified in different lines. The software reports the RON between each pin-pair listed in the file. This is a required parameter.

Example

- The following command creates a RON report in the `output.rpt` file for the pin-pairs specified in the `pin_pair.txt` file:

  ```
  report_vfi_XL_ron -pin_pair_file pin_pair.txt -output_file output.rpt
  ```

Related Topic

Reporting On-Resistance (RON) Values

© 2020 All Rights Reserved.

search_res_id

```
search_res_id
     -x x_location
     -y y_location
     -layer extracted_layer_name
     [-net net_name]
     -name resistor_name
```

Specifies the location and other information related to the resistor that is being queried in Voltus-Fi-XL. You can either search a specific resistor by providing its location, layer name, and net name; or query it by providing the name of the resistor.

You can also specify this command in a Tcl file (`conf.tcl`) and specify the file in the EMIR control file by using the following `emirutil` command as follows:

```
emirutil postTclCmdFile=[conf.tcl]
```


© 2020 All Rights Reserved.

Parameters

`-layer` *`extracted_layer_name`*	
	Specifies the name of the extracted layer on which the resistor is located. **Note:** This parameter is only required if the resistor name is not being specified using the `-name` parameter.
`-name` *`resistor_name`*	
	Specifies the name of the resistor to be queried. When this parameter is specified, no other parameter is required to be specified. If any other parameter is specified, it will be ignored.
`-net` *`net_name`*	Specifies the name of the net. This is an optional parameter. **Note:** This parameter is only specified if the resistor name is not being specified using the `-name` parameter.
`-x` *`x_location`*	Specifies the X-co-ordinate of the resistor to be queried. **Note:** This parameter is only required if the resistor name is not being specified using the `-name` parameter.
`-y` *`y_location`*	Specifies the Y-co-ordinate of the resistor to be queried. **Note:** This parameter is only required if the resistor name is not being specified using the `-name` parameter.

Examples

- The following command is used to query a resistor at the specified co-ordinates on metal layer, `m1`:

```
search_res_id
-x 4.78
-y 9.75
-layer m1
```

- The following command is used to query a specific resistor, `Ra12`:

```
search_res_id -name Ra12
```

Related Topics

res_id_info, search_node_id, node_id_info

© 2020 All Rights Reserved.

res_id_info

```
res_id_info
     -id res_id
     [-type
     {default/a/w/l/La/Lu/Lb/Wu/Wb/delta_T/Lv2v_rms/Tj/Tlife/Td/thickness/r/Iavg/
     Ipeak/Irms/Iabs_avg/Ipos_avg/Ineg_avg/Iavg_max/Ipeak_max/Irms_max/Iacpeak_ma
     x/T/N/name/layer/netName/node1/node2/type/x/y/current_direction/feolT/beolT/
     beta/alpha/alpha_connecting/alpha_overlapping}]
```

Specifies the information to be queried for the specified resistor (pRes) in Voltus-Fi-XL.

You can also specify this command in a Tcl file (`conf.tcl`) and specify the file in the EMIR control file by using the following `emirutil` command as follows:

```
emirutil postTclCmdFile=[conf.tcl]
```


© 2020 All Rights Reserved.

Parameters

`-id` *`res_id`*	Specifies the pRes ID for which the information is to be queried. This is a required parameter.

© 2020 All Rights Reserved.

`-type`	Specifies the type of information to be queried for the pRes. You can specify any number of options detailed below. The query will output the values of the specified options for the pRes.

	`default`: When this option is specified, the software returns a set of values for the specified pRes in a specific format. An example of the default output of this command is provided below:
	`name : rx3`
	`layer : m1`
	`netName : VG`
	`node1 : VG#2`
	`node2 : XX1/M0#g`
	`type : Metal`
	`x : 4.780`
	`y : 9.784`
	`current_direction : UnKnown`
	`w : 0.02`
	`l : 0.1`
	`Iavg : 3.41e-10`
	`Ipeak : 9.25e-07`
	`Irms : 1.28e-07`
	`Iavg_max : 6.6e-05`
	`Ipeak_max : 0`
	`Irms_max : 0`
	`Iacpeak_max : 0`
	`delta_T : 5`
	`Td : 0.000315`
	`r : 0.0315`
	`rmsLv2v : 1.8e+308`
	`Iavg_rule : (w*3) * 1.1 * 1 : (l <= 5)`

	`a`: Retrieves the area of the pRes.
	`w`: Retrieves the width of the pRes.
	`l`: Retrieves the Blech length of the pRes.
	`La`: Retrieves the length of the pseudo vias.
	`Lu`: Retrieves the upper metal Blech length.
	`Lb`: Retrieves the bottom metal Blech length.
	`Wu`: Retrieves the upper metal width.

© 2020 All Rights Reserved.

Examples

- The following set of commands are used to query the specified resistor on metal layer, `m1`, for the specified information:

  Step1: Search resistor id:

  ```
  set rid  [search_res_id -x 3.75 -y 8.35 -layer m1]
  ```

  Step2: Get specific information from resistor id:

  ```
  puts [res_id_info  -id $rid -type Iacpeak_max]
  0
  ```

- The following set of commands are used to query the specified resistor, `Ra12`, for all information:

  Step1: Search resistor id:

  ```
  set rid [search_res_id -name Ra12]
  ```

  Step2: Get specific information from resistor id:

  ```
  puts [res_id_info -id $rid]
  name : ra12
  layer : via1
  netName : VDD
  node1 : VDD#1
  node2 : VDD#2
  type : Via
  x : 4.825
  y : 9.269
  current_direction : Down
  w : 0.02
  l : 6
  Iavg : 1.06e-05
  Ipeak : 2.15e-05
  Irms : 1.5e-05
  Iavg_max : 1e-05
  Ipeak_max : 0
  Irms_max : 0
  Iacpeak_max : 0
  delta_T : 5
  Td : 0.00493
  r : 0.493
  a : 0.33
  N : 1
  Lu : 6
  ```


© 2020 All Rights Reserved.

```
Wu : 0.06
Lb : 0.1
Wb : 0.02
rmsLv2v : 1.8e+308
Iavg_rule : 0.01 * 1 * 1
```

Related Topics

search_res_id, search_node_id, node_id_info

search_node_id

```
search_node_id
     -x x_location
     -y y_location
     -layer extracted_layer_name
     [-net net_name]
```

Specifies the information required to search for the node to be queried in Voltus-Fi-XL.

You can also specify this command in a Tcl file (`conf.tcl`) and specify the file in the EMIR control file by using the following `emirutil` command as follows:

```
emirutil postTclCmdFile=[conf.tcl]
```


© 2020 All Rights Reserved.

Parameters

`-layer` *`extracted_layer_name`*	Specifies the name of the extracted layer on which the node is located. This is a required parameter.
`-net` *`net_name`*	Specifies the name of the net. This is an optional parameter.
`-x` *`x_location`*	Specifies the x-coordinate for the node to be searched. This is a required parameter.
`-y` *`y_location`*	Specifies the y-coordinate for the node to be searched. This is a required parameter.

Example

The following command is used to search for a node at the specified coordinates on metal layer, `m2`:

```
search_res_id
-x 3.75 \
-y 8.35 \
-layer m2
```

Related Topics

search_res_id, res_id_info, node_id_info

node_id_info

```
node_id_info
     -id node_id
     [-type {default/name/x/y/IRavg/IRpeak}]
```

Specifies the information to be retrieved for the specified resistor in Voltus-Fi-XL.

You can also specify this command in a Tcl file (`conf.tcl`) and specify the file in the EMIR control file by using the following `emirutil` command as follows:

```
emirutil postTclCmdFile=[conf.tcl]
```


© 2020 All Rights Reserved.

Parameters

`-id` *`node_id`*	Specifies the node ID for which the information is to be retrieved. This is a required parameter.
`-type`	Specifies the type of information to be retrieved for the node. You can specify from the options detailed below using this parameter.

	`default:` When this option is specified, the software returns a set of values for the specified node in a specific format. An example of the default output of this command is provided below:
	`name : VG#2`
	`x : 4.780`
	`y : 9.834`
	`IRavg : 8.54e-07`
	`IRpeak : 0.00224`
	`netName : VG`

	name: Retrieves the name of the node.
	`x`: Retrieves the x-coordinate for the node.
	`y`: Retrieves the y-coordinate for the node.
	`IRavg`: Retrieves the average voltage drop for the node.
	`IRpeak`: Retrieves the peak voltage drop for the node.

Example

The following set of commands are used to query the specified node on metal layer, `m2`, for the specified information:

Step1: Search node id:

```
set nid [search_node_id -x 3.75 -y 8.35 -layer m2]
```

Step2: Get specific information from node id:

```
puts [node_id_info -id $nid -type IRavg]
```

Related Topics

search_res_id, res_id_info, search_node_id

© 2020 All Rights Reserved.

17

Environment Variables

This chapter contains the Voltus-Fi-XL environment variable names, descriptions, types, and values. You can customize the operation and behavior of the Voltus-Fi-XL product by changing the value of a particular environment variable.

This chapter lists environment variables belonging to the following categories:

- Environment Setting (cdsenv) Variables - You can specify environment settings and initialization instructions for Voltus-Fi-XL in the `.cdsenv` and `.cdsinit` files, respectively. Using these settings, you can customize your software environment.
- Shell Environment Variables - You can specify environment variables that are set using the `setenv` command before invoking the Voltus-Fi-XL software.

© 2020 All Rights Reserved.

Environment Setting (cdsenv) Variables

You can configure the default options of the Extract xDSPF and IR/EM Results form fields by using environment variables in the `.cdsenv` and `.cdsinit` files.

- Extract xDSPF Form Variables
  - foundry
  - vfiAutoChkBox
  - vfiBlocking
  - vfiCouplingmode
  - vfiColoring
  - vfiCompression
  - vfiCustom
  - vfiDatatype
  - vfiExtractmode
  - vfiGndNet
  - vfiInputdir
  - vfiMaxviarrcount
  - vfiMulticpu
  - vfiNetsfile
  - vfiOutdir
  - vfiOutdspf
  - vfiPgnets
  - vfiPinorder
  - vfiProcess
  - vfiReduction
  - vfiRunname
  - vfiSignalem
  - vfiSubckt

© 2020 All Rights Reserved.

- vfiSubCondModel
- vfiSubNode
- vfiViaArrayspacing
- vfiViaCnt
- vfiViaSpacingEdit
- vfiViaSpacingEditChk
- vfiTechCorner
- vfiTechLib
- vfiTechName
- vfiTechType
- vfiTemperature

- IR/EM Results Form Variables
  - vfiDfiiLayerMapFile
  - vfiElementCount
  - vfiEMOnlyIctFile
  - vfiIRThreshold
  - vfiLayermapFile
  - vfiLayerIndex
  - vfiNoOfThreads
  - vfiQrcRunDir
  - vfiQrcRunName
  - vfiResultDirectoryIr
  - vfiSetShrinkFactor
  - vfiTechFile
  - vfiTypeCheckboxMax
  - vfiTypeCheckboxAvg
  - vfiTypeCheckboxAvgAbs

© 2020 All Rights Reserved.

- vfiTypeCheckboxAcpeak
- vfiTypeCheckboxRms
- vps_results_plots_distributeRange
- vps_results_plots_OutofRange

© 2020 All Rights Reserved.

Extract xDSPF Form Variables

foundry

Specifies the foundry.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "foundry" 'string "foundry_name")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF foundry string "foundry_name"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	None

vfiAutoChkBox

Specifies to set the via array spacing automatically.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiAutoChkBox" 'string "0")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiAutoChkBox string "0"
```

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Via Array Spacing - auto*

© 2020 All Rights Reserved.

vfiBlocking

Specifies the file containing parasitic blocking device cell names, if any. Once a cell is declared a blocking cell, all the parasitic RCs inside the cell are not extracted.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiBlocking" 'string "/VoltusFi_workshop/p")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiBlocking string "/VoltusFi_workshop/p"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Parasitic blocking device cells file*

vfiCouplingmode

Specifies the coupling mode for capacitance extraction. You can specify one of the following:

- `Decoupled`: specifies that decoupled capacitance extraction will be performed
- `Coupled`: specifies that coupled capacitance extraction will be performed.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiCouplingmode" 'string "decoupled")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiCouplingmode string "decoupled"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value (decoupled or coupled)
GUI Equivalent	*Cap Coupling Model*

© 2020 All Rights Reserved.

vfiColoring

Specifies to enable color-aware EM analysis.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiColoring" 'string "0")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiColoring string "0"
```

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Parasitic RES Color*

vfiCompression

Specifies to generate xDSPF in the compressed format.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiCompression" 'string "0")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiCompression string "0"
```

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Select Compression*

vfiCustom

Specifies the file with user-customized extract commands. When this file is specified, the Voltus-Fi-generated CCL will not have any extract commands. In this case, the commands from the customized extract commands file will be added in the Voltus-Fi-generated CCL.

To set this variable in the `.cdsinit` file or CIW, use the call:

© 2020 All Rights Reserved.

```
envSetVal("voltus_fi.extractDSPF" "vfiCustom" 'string "Custom")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiCustom string "Custom"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Customized Extract Commands File*

vfiDatatype

Specifies whether the input data will come from Pegasus or Calibre.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF"  "vfiDatatype" `string "Calibre")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiDatatype string "Calibre"
```

Variable Type	string
Default Value	pvs
Acceptable Values	pvs or Calibre
GUI Equivalent	*Data Type*

vfiExtractmode

Specifies the type of extraction to be performed. You can specify one of the following:

- `RC`: specifies that both parasitic resistors and capacitors will be extracted
- `R only`: specifies that only parasitic resistors will be extracted

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiExtractmode" `string "R only")
```

To set this variable in the `.cdsenv` file, use the call:

© 2020 All Rights Reserved.

```
voltus_fi.extractDSPF vfiExtractmode string "R only"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value (R Only or RC)
GUI Equivalent	*Extraction Type*

vfiGndNet

Specifies the reference node on which extraction is to be performed.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiGndNet" 'string "GNDNet")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiGndNet string "GNDNet"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Ref Node*

vfiInputdir

Specifies the path to the directory in which the input data is stored.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF"  "vfiInputdir" `string "/PVS/PVS_lvs_qrc/svdb")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiInputdir string "PVS/PVS_lvs_qrc/svdb"
```

Variable Type	string
Default Value	nil

© 2020 All Rights Reserved.

Acceptable Values	Any string value
GUI Equivalent	*Data Directory*

vfiMaxviarrcount

Specifies the number of vias per side allowed in an array. For example, if the via array count is set to 2, the number of vias allowed per array side will be 2.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiMaxviarrcount" 'string "4")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiMaxviarrcount string "4"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any integer
GUI Equivalent	*Via Array Count*

vfiMulticpu

Specifies the number of CPUs to be used during the Quantus extraction run.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiMulticpu" 'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiMulticpu string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any integer
GUI Equivalent	*No of CPUs*

© 2020 All Rights Reserved.

vfiNetsfile

Specifies the file that contains the names of the nets to be included in extraction.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiNetsfile" 'string "NetsFile")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiNetsfile string "NetsFile"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Nets File*

vfiOutdir

Specifies the output directory in which the xDSPF file will be saved.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiOutdir" 'string "/VoltusFi_workshop/mydir")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiOutdir string "/VoltusFi_workshop/mydir"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Output Directory*

vfiOutdspf

Specifies the name of the xDSPF file that will be generated.

To set this variable in the `.cdsinit` file or CIW, use the call:

© 2020 All Rights Reserved.

```
envSetVal("voltus_fi.extractDSPF" "vfiOutdspf" 'string "cellname.dspf")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiOutdspf string "cellname.dspf"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Output xDSPF File Name*

vfiPgnets

Specifies the power and ground nets that should be included in the xDSPF file.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiPgnets" 'string "PGnets")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiPgnets string "PGnets"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*PG Net(s)*

vfiPinorder

Specifies the sub-circuit pin order file to provide the order of pins to be used during extraction.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF"  "vfiPinorder" 'string
"/VoltusFi_workshop/pin2pin.conf")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiPinorder string "/VoltusFi_workshop/pin2pin.conf"
```


© 2020 All Rights Reserved.

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*SubCkt Pin Order File*

vfiProcess

Specifies the process node.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiProcess" 'string "N7")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiProcess string "N7"
```

Variable Type	string
Default Value	N5/N3
Acceptable Values	N5/N3, N7, N10-PG, N10, N20/N16/N12, N14, Others > 20
GUI Equivalent	*Foundry / Process Node*

vfiReduction

Specifies to enable the `reduce_i_cards` option in the CCL file that is used for generating the xDSPF file.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiReduction" 'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiReduction string "1"
```

Variable Type	string

© 2020 All Rights Reserved.

Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Reduce I-Cards*

vfiRunname

Specifies the name of the Pegasus or Calibre run.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF"  "vfiRunname"  'string "DiffopAmp")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiRunname string "DiffopAmp"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Run Name*

vfiSignalem

Specifies that signal nets are to be included in the xDSPF file.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiSignalem" `string "0")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiSignalem string "0"
```

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Signal EM*

© 2020 All Rights Reserved.

vfiSubckt

Specifies to comment out the top sub circuit statements from the xDSPF file.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiSubckt" 'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiSubckt string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Comment Sub Ckt*

vfiSubCondModel

Specifies that the parasitic resistor models for subconductor layers should be included during extraction.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiSubCondModel" 'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiSubCondModel string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Sub Conductor Model*

vfiSubNode

Specifies the subnode character that will be used for subnode identification.

© 2020 All Rights Reserved.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiSubNode" 'string "subnode")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiSubNode string "subnode"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Sub Node Char*

vfiViaArrayspacing

Specifies to enable via array spacing for grouping vias within the same array.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiViaArrayspacing" 'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiViaArrayspacing string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Via Array Spacing*

vfiViaCnt

Specifies the fracture via count to be used for dividing pseudo vias into segments during extraction.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiViaCnt" 'string "10")
```

To set this variable in the `.cdsenv` file, use the call:

© 2020 All Rights Reserved.

```
voltus_fi.extractDSPF vfiViaCnt string "10"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any integer
GUI Equivalent	*Fracture Via Count*

vfiViaSpacingEdit

Specifies a value for via array spacing. This means that if the distance between vias is less than the specified maximum distance, they are grouped within the same array.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiViaSpacingEdit" 'string "0.40")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiViaSpacingEdit string "0.40”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any floating value
GUI Equivalent	*Via Array Spacing* - value

vfiViaSpacingEditChk

Specifies to use the user-specified value for via array spacing.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiViaSpacingEditChk" 'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiViaSpacingEditChk string "1”
```


© 2020 All Rights Reserved.

Variable Type	string
Default Value	nil
Acceptable Values	0 or 1
GUI Equivalent	*Via Array Spacing* - not set automatically

vfiTechCorner

Specifies the name of the technology corner to be picked from the library.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiTechCorner" 'string "rcx_typical")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiTechCorner string "rcx_typical”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Corner*

vfiTechLib

Specifies the path of the technology file to be used for reading the technology data.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiTechLib" 'string
"/VoltusFi_workshop/pvtech.lib")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiTechLib string "/VoltusFi_workshop/pvtech.lib”
```

Variable Type	string

© 2020 All Rights Reserved.

Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Technology Library File*

vfiTechName

Specifies the technology filename to be used for extraction.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiTechName" 'string "gpdk045")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiTechName string "gpdk045”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Technology Name*

vfiTechType

Allows you to select either *Technology File* or *Technology Directory* from the drop-down list.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiTechType" 'string "Technology Directory")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiTechType string "Technology Directory”
```

Variable Type	string
Default Value	Technology Library File
Acceptable Values	Technology Directory or Technology Library File

© 2020 All Rights Reserved.

GUI Equivalent	*Technology Library File/Technology Directory* drop-down list

vfiTemperature

Specifies the temperature at which extraction will be performed.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.extractDSPF" "vfiTemperature" 'string "25")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.extractDSPF vfiTemperature string "25”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any floating value
GUI Equivalent	*Temperature*

© 2020 All Rights Reserved.

IR/EM Results Form Variables

vfiDfiiLayerMapFile

Specifies the APS/XPS-to-DFII layer map file that provides the mapping between the layer names in the extractor-generated xDSPF file to the DFII layer names.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiDfiiLayerMapFile"  'string
"MMSIM_EMIR_workshop/df2layermap")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiDfiiLayerMapFile string "MMSIM_EMIR_workshop/df2layermap”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*DFII Layer Map*

vfiElementCount

Specifies the number of violations you want to view in the *EM/IR* tab of Annotation Browser, and the violation markers in the plot on the Virtuoso layout.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiElementCount"  'string “100”)
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiElementCount string "100”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Temperature*

© 2020 All Rights Reserved.

vfiEMOnlyIctFile

Specifies the ICT file that includes information about process and EM models to be used for EM analysis.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiEMOnlyIctFile"  'string "/ICTfile")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiEMOnlyIctFile string "/ICTfile"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*EM Only ICT File*

vfiIRThreshold

Specifies the IR threshold for viewing IR drop violations above the specified threshold value.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiIRThreshold"  'string "0.25")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiIRThreshold string "0.25"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*IR threshold*

vfiLayermapFile

Specifies the file used to map the layer names in the simulation database to the layer names in the technology file.

© 2020 All Rights Reserved.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiLayermapFile"  'string "/df2layermap")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiLayermapFile string "/df2layermap"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Layer Map File*

vfiLayerIndex

Voltus-Fi-XL creates layers for displaying the EMIR violations when you plot them on the Virtuoso Layout. By default, the software uses more than a hundred layers for displaying violations. It creates these layers by using a layer index value or layer number that is equal to or greater than 256.

However, if you are already using the same layer number as that used by Voltus-Fi-XL, a layer number conflict will occur and a warning will be issued. To avoid this warning, specify a different value for the layer number.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiLayerIndex"  'int 270')
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiLayerIndex int 270
```

Variable Type	int
Default Value	nil
Acceptable Values	Any integer value
GUI Equivalent	*None*

© 2020 All Rights Reserved.

vfiNoOfThreads

Specifies the maximum number of threads/CPUs to be used for EM analysis.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiNoOfThreads"  'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiNoOfThreads string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any integer value
GUI Equivalent	*None*

vfiQrcRunDir

Specifies the path to the Quantus run directory.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiQrcRunDir"  'string "/diffopamp_lvs/svdb")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiQrcRunDir string "/diffopamp_lvs/svdb”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*QRC Run Directory*

vfiQrcRunName

Specifies the Quantus run name.

To set this variable in the `.cdsinit` file or CIW, use the call:

© 2020 All Rights Reserved.

```
envSetVal("voltus_fi.results"  "vfiQrcRunName"  'string "DiffOpAmp")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiQrcRunName string "DiffOpAmp"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*QRC Run Name*

vfiResultDirectoryIr

Specifies the state directory or the result file that stores the simulation results.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiResultDirectoryIr"  'string
"/emir_simulation/input.emir0_bin")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiResultDirectoryIr string "/emir_simulation/input.emir0_bin"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*State Directory/Results File*

vfiSetShrinkFactor

Specifies the factor by which xDSPF was shrunk. This is used for flows where the xDSPF is generated with a shrunk technology but the layout remains on the original technology.

```
To set this variable in the .cdsinit file or CIW, use the call:
envSetVal("voltus_fi.results"  "vfiSetShrinkFactor"  'string "0.1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiSetShrinkFactor string "0.1"
```


© 2020 All Rights Reserved.

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Shrink Factor*

vfiTechFile

Specifies the technology file.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiTechFile"  'string
"/emir_simulation/emDataFile.txt")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiTechFile string "/emir_simulation/emDataFile.txt"
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value
GUI Equivalent	*Tech File*

vfiTypeCheckboxMax

Specifies to calculate the Current Density violations based on the maximum DC current.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiTypeCheckboxMax"  'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiTypeCheckboxMax string "1"
```

Variable Type	string

© 2020 All Rights Reserved.

Default Value	nil
Acceptable Values	true or false
GUI Equivalent	*max* (EM tab)

vfiTypeCheckboxAvg

Specifies to calculate the Current Density violations based on the average value of the DC current.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiTypeCheckboxAvg"  'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiTypeCheckboxAvg string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	true or false
GUI Equivalent	*avg* (EM tab)

vfiTypeCheckboxAvgAbs

Specifies to calculate the Current Density violations based on the average value of absolute current.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiTypeCheckboxAvgAbs"  'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiTypeCheckboxAvgAbs string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	true or false

© 2020 All Rights Reserved.

GUI Equivalent	*avgabs* (EM tab)

vfiTypeCheckboxAcpeak

Specifies to calculate the Current Density violations based on peak AC current.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiTypeCheckboxAcpeak"  'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiTypeCheckboxAcpeak string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	true or false
GUI Equivalent	*AC Peak* (EM tab)

vfiTypeCheckboxRms

Specifies to calculate the Current Density violations based on the root mean square (RMS) value of the AC current.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vfiTypeCheckboxRms"  'string "1")
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vfiTypeCheckboxRms string "1"
```

Variable Type	string
Default Value	nil
Acceptable Values	true or false
GUI Equivalent	*rms* (EM tab)

© 2020 All Rights Reserved.

vps_results_plots_distributeRange

Specifies to re-distribute the plot display range to view the complete spectrum, from “deep red” to “deep blue”, for the specified layer selection for the plots in the Display form of the IR/EM Results form in the GUI.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vps_results_plots_distributeRange"  'string
“true”)
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vps_results_plots_distributeRange string “true”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value (true or false)
GUI Equivalent	*Re-distribute Range* option in the Display form of the IR/EM Results form

vps_results_plots_OutofRange

Specifies whether to plot out of range values in the Display form of the IR/EM Results form in the GUI. For this, select the *Plot out of range values* option in the *Plot Options* group box in the Display form.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("voltus_fi.results"  "vps_results_plots_OutofRange"  'string “true”)
```

To set this variable in the `.cdsenv` file, use the call:

```
voltus_fi.results vps_results_plots_OutofRange string “true”
```

Variable Type	string
Default Value	nil
Acceptable Values	Any string value (true or false)
GUI Equivalent	*Plot out of range values* option in the Display form of the IR/EM Results form

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

Shell Environment Variables

The shell environment variables are set using the `setenv` command before invoking the Voltus-Fi-XL software.

- VFI_LIC_QUEUE on page 652
- VOLTUSFI_DELTA_T on page 652
- VOLTUSFI_DYNAMIC_AC_PEAK on page 652
- VOLTUSFI_SPLIT_ACDC_RULES on page 652
- VFI_USE_OLD_SHAPE_DB on page 653
- VFI_POINT_QUERY on page 653
- VFI_ENABLE_INTERSECTING_SHAPE on page 653
- VFI_SPLIT_NON_RECTANGULAR_SHAPE on page 654
- vfi_dfii_unmap_skip on page 654
- vfi_ignore_conf_file on page 654
- CDS_MMSIM_VOLTUSFI_ROOT on page 654

© 2020 All Rights Reserved.

VFI_LIC_QUEUE

`setenv VFI_LIC_QUEUE` *`time_in_seconds`*

Specifies the duration for which the software should wait for the availability of the Voltus-Fi-XL license. By default, the software waits for 24 hours or 86400 seconds.

Example:

```
setenv VFI_LIC_QUEUE 1000
```

VOLTUSFI_DELTA_T

`setenv VOLTUSFI_DELTA_T` *`value`*

Specifies the deltaT value for EM analysis. It is used to specify maximum rise in temperature in degree Celsius, caused due to Joule Heating. It must be a positive value.

If you do not specify the rise in temperature, the software considers a default temperature of 5 degree Celsius. The variable, `deltaT`, can also be specified using either the `set_variable` command or in the Variables form. This is detailed in subsequent sections.

Example:

```
setenv VOLTUSFI_DELTA_T 15
```

VOLTUSFI_DYNAMIC_AC_PEAK

`setenv VOLTUSFI_DYNAMIC_AC_PEAK true | false | multiplePeak`

Specifies that the current density violations should be calculated based on the peak AC current for metal lines, vias, and contacts. It is applied to periodic AC or pulsed DC signals. Valid values are `true`, `false`, and `multiplePeak`. For details, see `dynamicACPeak`.

Example:

```
setenv VOLTUSFI_DYNAMIC_AC_PEAK true
```

VOLTUSFI_SPLIT_ACDC_RULES

`setenv VOLTUSFI_SPLIT_ACDC_RULES true | false`

© 2020 All Rights Reserved.

Specifies different rules for EM analysis of power and signal nets. The default value is `false`. For details, see `splitACDCRules`. You can also see Rules for Specifying the EM Analysis Type for Power and Ground Nets in the "EM Rules Specification" chapter.

Example:

```
setenv VOLTUSFI_SPLIT_ACDC_RULES true
```

VFI_USE_OLD_SHAPE_DB

```
setenv VFI_USE_OLD_SHAPE_DB true | false
```

The Voltus-Fi Quantus shape-server flow uses a new node name-based matching algorithm for displaying the shapes on the layout. This flow is enabled by default. However, in case of any issues with this flow, you can set the `VFI_USE_OLD_SHAPE_DB` variable to `true` to enable the earlier shape-server flow.

Example:

```
setenv VFI_USE_OLD_SHAPE_DB true
```

VFI_POINT_QUERY

```
setenv VFI_POINT_QUERY value
```

When set, runs the IR/EM analysis results query is performed within the defined square area around the specified point query. The value is specified in nanometer (nm). When this variable is not set, the software queries the nearest matching data point.

Set this variable before starting the GUI.

Example:

```
setenv VFI_POINT_QUERY 5
```

For a point selected on x- and y-coordinates, the software will query the following square area:

```
x-5,x+5,y-5,and y+5
```

VFI_ENABLE_INTERSECTING_SHAPE

```
setenv VFI_ENABLE_INTERSECTING_SHAPE
```


© 2020 All Rights Reserved.

Set this environment variable to enable processing of self-intersecting shapes in the layout. If this variable is not set, self-intersecting shapes are skipped during analysis and not highlighted in the layout while displaying the EMIR analysis plots.

VFI_SPLIT_NON_RECTANGULAR_SHAPE

```
setenv VFI_SPLIT_NON_RECTANGULAR_SHAPE
```

This variable sets the option in the VFI GUI to split the non-rectangular layout shapes based on the resistors extracted over them. By default, the GUI splits only the long rectangular shapes in the layout into smaller shapes to increase the display granularity of the plots.

Note: This environment variable is applicable only for the flow that uses a DFII layermap file for display.

vfi_dfii_unmap_skip

```
setenv vfi_dfii_unmap_skip
```

Set this environment variable to hide the layers that are not mapped in the DFII layer map file from the layer table of the IR/EM Display form. If this variable is not set, all layers are displayed in the layer table of the Display form.

vfi_ignore_conf_file

```
setenv vfi_ignore_conf_file true | false
```

By default, the variable settings in the `input.conf` file override the values for the same variables that are specified in the GUI. To ignore the variable settings in the `conf` file and use the specified GUI settings, set the `vfi_ignore_conf_file` environment variable to `true`. By default, this variable is set to `false`.

Example:

```
setenv vfi_ignore_conf_file true
```

CDS_MMSIM_VOLTUSFI_ROOT

```
setenv CDS_MMSIM_VOLTUSFI_ROOT <installation_dir_path>
```

When set, specifies the path of the emirreport binary to be used for EMIR report generation in the Spectre run. It overrides any emirreport version in the MMSIM/Spectre stream.

© 2020 All Rights Reserved.

Note: This variable only affects the emirreport when it is run automatically following the Spectre EMIR simulation. It does not affect the manual running of the emirreport from the command line. To use this variable, the emirreport must be present in the specified path.

Example:

```
setenv CDSHOME /home/IC6.1.7/ISR12/lnx86/64/

setenv CDS_MMSIM_VOLTUSFI_ROOT $CDSHOME
```


© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

18

Variables

- Overview on page 658
- Using the set_variable Command in the Voltus-Fi-XL Command File on page 658
- Using the emirutil Command in the EMIR Control File on page 659
- Setting the Configuration Variable in Batch Mode on page 659
- Using the Variables Form in the Voltus-Fi-XL GUI on page 660

© 2020 All Rights Reserved.

Overview

This chapter provides a list of variables that you can specify in Voltus-Fi-XL. You can set these variables both in the GUI and in batch mode.

In the GUI, you can set these variables in the Variables form, which opens when you click *Variables* in the EM tab of the IR/EM Results form. For details, see Viewing the EM Analysis Results in the "EM Analysis Results" chapter.

In batch mode, you can set these variables using the `set_variable` and `emirutil` commands.

Information about how these variables can be set in both batch mode and in the GUI is provided in the following sections:

- Using the set_variable Command in the Voltus-Fi-XL Command File
- Using the emirutil Command in the EMIR Control File
- Using the Variables Form in the Voltus-Fi-XL GUI

Using the set_variable Command in the Voltus-Fi-XL Command File

When using the `vfibatch` command to run Voltus-Fi-XL in batch mode, you can set the variables using the `set_variable` command. The syntax of the command and an example is provided below.

set_variable

`set_variable` *`var_name var_value`*

Sets the user-specified value for the variable in batch mode.

Parameters

`var_name`	Specifies the name of the variable for which you want to set the value.
`var_value`	Specifies the value that you want to set for the variable.

© 2020 All Rights Reserved.

Example:

The following command sets the value of the variable, `dbu` to 10000 database units per micron:

```
set_variable dbu 10000
```

Using the emirutil Command in the EMIR Control File

When using the EMIR control file to generate textual reports of EMIR results in Voltus-Fi-XL in batch mode, you can set the variables using the `emirutil` command in the control file. The command syntax and example is provided below.

```
emirutil <var_name=var_value>
```

Example:

The following commands set the values of variables, `deltaT`, `dbu`, and `dynamicACPeak` to `10, 500,` and `false`, respectively.

```
emirutil deltaT=10.0
emirutil dbu=500
emirutil dynamicACPeak=false
```

For details of the supported EMIR control file options, see the EMIR Flow Setup section in the "Postlayout Simulation" chapter of the *Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

Setting the Configuration Variable in Batch Mode

You can set the configuration variable, `dbu` or `cadGrid` using either the `set_variable` command or the `emirutil` command.

The configuration variable, `dbu`, specifies the number of database units (DBU) per micron for the layout view. The default value of `dbu` is 1000.

The `cadGrid` variable specifies the resolution of the software in terms of width and length and the unit of measurement is micrometer or micron. The default value of `cadGrid` is `0.001`.

Note: The value of the `dbu` variable is `1/cadGrid`. If both `dbu` and `cadGrid` are specified, the `dbu` variable value is given priority.

© 2020 All Rights Reserved.

The above variables are specified only in batch mode when the view information is not specified in either the `vfibatch` command file or in the `emir` configuration file. They are not required to be specified when running the tool in the GUI mode as it is read from the open view. If this variable is not specified, the tool uses the default value.

For more information about how this variable is set in the Virtuoso Layout, see “Layout L Forms” chapter in the *Virtuoso Layout Suite L User Guide*.

Using the Variables Form in the Voltus-Fi-XL GUI

The following variables can be specified using the Variables form that can be accessed from the EM tab of the IR/EM Results form:

- applyRThreshold
- connectingODLayers
- consolidatedReport
- coupledSH
- deltaT
- deltaTLayers
- disableLayoutScale
- dynamicACPeak
- emLimitScale
- emVariableValues
- enableContextAwareSrmsRelaxation
- enableDesignResistorPeakFactor
- enableDesignResistorScaling
- enablePWLDensityInterpolation
- enableReportScaleParam
- enableRMSHotSpotWaiving
- enableSHHeatSink
- extendHeatSinkRange

© 2020 All Rights Reserved.

- extendJouleHeating
- extendODRange
- extendMetalRange
- extendSHRegion
- geounit
- geounit_wl
- geounit_xy
- layermap
- mdLayers
- mergeMDLayers
- multiplePeakPercentThreshold
- optimisticEMRuleSelection
- powerRailGroundNets
- powerRailRules
- powerRailSupplyNets
- process_scale
- prioritizedPwlRulesLayers
- reportUnmatchedRuleResistor
- rmsMetalLineNumberPower
- rmsMetalLineNumberSignal
- rmsMultiplierPin
- rmsMultiplierSignal
- shEMEffect
- shParamFile
- shXMLFile
- simpleSingleViaDefinition
- splitACDCRules

© 2020 All Rights Reserved.

- Tj
- Tlife
- toggleNetFrequency
- toggleNetFrequencyFile
- useEncryptedWidth
- viaCountFromResistance

The above variables are described in detail below.

applyRThreshold

Specifies the threshold ratio “r” for AC-Peak analysis.

connectingODLayers

Specifies the list of layers to identify connecting ODs for heat sink effect. The default value is shown below.

```
connectingODLayers M0_OD M0_OD1 M0_OD2 MVIA0_1 MVIA0_2 M1
```

Note: This variable is only used for the SHE flow with heat-sink effect.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil connectingODLayers=”M0_OD M0_OD1 M0_OD2 MVIA0_1 MVIA0_2 M1”
```

consolidatedReport

When this variable is set to `true`, consolidated EM and IR reports are generated for all available bin files. The default value of this variable is `false`.

coupledSH

When set to `true`, specifies that the thermal-aware DC EM flow should be enabled. By default, this variable is set to `false`.

© 2020 All Rights Reserved.

deltaT

Specifies the maximum rise in temperature in degree Celsius, caused due to Joule heating. It must be a positive value.

`deltaT` *`temp_value`*

If the rise in temperature is not specified, Voltus-Fi-XL considers a default temperature of 5 degrees Celsius. This is an ideal condition wherein the design must be EM free.

Example:

```
deltaT 10.0
```

`deltaT` can be used for defining the current density limit: `Irms`. For example:

```
em_jmax_ac_rms  EQU (w - 0.004) * 1.24 * deltaT
```

The `deltaT` option can also be specified in the control file by using the following command:

`emirutil deltaT=`*`temp_value`*

deltaTLayers

Specifies layer-specific `deltaT` values. The layers for which `deltaT` value is not specified, global `deltaT` value will be used. The syntax is as follows:

`deltaTLayers "(layer=`*`layer_name1`* `deltaT=`*`temp_value1`*`)`
`(layer=`*`layer_name2`* `deltaT=`*`temp_value2`*`)"`

For example,

```
deltaTLayers "(layer=m1 deltaT=5) (layer=m2 deltaT=25)"
```

In the above example, the `deltaT` value for `m1` layer is `5` degree Celsius while that for `m2` layer is `25` degree Celsius.

This option can also be specified in the control file by using the following command:

```
emirutil deltaTLayers=[(layer=m1 deltaT=5) (layer=m2 deltaT=25)]
```

disableODSH

When set to `true`, disables the self-heating effect of devices in the SHE flow. By default, this variable is set to `false`.

© 2020 All Rights Reserved.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil disableODSH=true
```

disableHRSH

When set to `true`, disables the self-heating effect of design resistors in the SHE flow. By default, this variable is set to `false`.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil disableHRSH=true
```

disableLayoutScale

Disables the layout scale factor defined in the ICT file. The layout_scale factor is the ratio of the scaled dimensions divided by the drawn dimensions. The ICT file is defined based on the scaled dimensions. Quantus applies the layout_scale factor to the

input design during extraction so that the design dimensions and the ICT file dimensions are consistent.

When the `disableLayoutScale` variable is set to `true`, the layout_scale factor specified in the ICT file is ignored in the Voltus-Fi-XL EM analysis flow. By default, this variable is set to `false`.

dynamicACPeak

Specifies that the current density violations should be calculated based on the peak AC current for metal lines, vias, and contacts. It is applied to periodic AC or pulsed DC signals. Valid values are `true`, `false`, and `multiplePeak`. When the `multiplePeak` option is specified, the software considers multiple peaks for the AC peak calculations.

$I_{peak_ac}=I_{peak_dc}/(r)^{1/2}$

where duty ratio “r” is:

$r=\mathbf{I}_{abs\text{-}avg} / \mathbf{I}_{peak_dc}$, by default (“`false`”)

And

$r=\mathbf{T}_{d}/\mathbf{T}_{total}$, when the `dynamicACPeak` variable is set.

© 2020 All Rights Reserved.

$\mathbf{T}_{total}$=total transient time

$\mathbf{T}_d$= the time duration in micro second or the total "On Time" period, where `abs(I)>(max(abs(I)/2.0)` during transient analysis

Note: Voltus-Fi-XL does not calculate $\mathbf{T}_d$ as `max(abs(I)/ 2.0)` period. But it calculates $\mathbf{T}_d$ by measuring the pulse at half energy level due to performance reason. The variable, `pwc_threshold` specified using the below command in the EMIR configuration file defines the current threshold for finding the start of the pulse.

```
net analysis=[ipwc] pwc_threshold=1e-6
```

$\mathbf{T}_d$ has the following different values depending upon the value of `dynamicACPeak`:

- $\mathbf{T}_d$=the time duration of maximum peak, when `dynamicACPeak` is set to "`true`"
- $\mathbf{T}_d$=sum of time durations of different peaks, $\mathbf{T}_{d1}$+$\mathbf{T}_{d2}$+$\mathbf{T}_{dN}$.... (when `dynamicACPeak` is set to "`multiplePeak`")

  Note: When `dynamicACPeak` is set to either "`true`" or "`multiplePeak`", the command, `net analysis=[ipwc]` must be specified in the EMIR configuration file.

Note: The software replaces the value of "r" with the value of `applyRThreshold` if the value of "r" is < than that of `applyRThreshold`. This is because a small "r" value results in an unreasonable increase in the $\mathbf{I}_{peak_ac}$ limit. To avoid this scenario, use the applyR keyword to reset the value of "r" in the ICT file or change the value of the `applyRThreshold` variable, either in the Variables form or in batch mode.

emLimitScale

Scales the EM limits of the parasitic resistors in the design before generating the EM reports. You can either apply the same scale factor to all the nets or specify different scale factors for different nets. Both use models are detailed below.

- To apply the same scale factor to the parasitic resistors of all the nets in the design, provide the value of the scale factor using this variable. For example,

  `emLimitScale` *`scale_factor`*

  ```
  emLimitScale 0.04
  ```

  Note: This is applicable to `avg`, `avg abs`, and `rms` analyses. For the `avg` and `avg abs` analyses, the limit is directly multiplied by this factor. For `rms` analysis, the limit is multiplied by the root of this factor.

© 2020 All Rights Reserved.

- To apply different scale factors to different nets in the design, provide the name of the file that contains information about the net names and the corresponding scale factors to be applied to these nets. For example,

  `emLimitScale` *`scale_factor_netfile`*

  An example of the scale factor file is as follows:

  ```
  net_name1 scale_factor1
  net_name2 scale_factor2
  ```

emVariableValues

`emVariableValues "(variable=configA` *`value=#`*`) (variable=configB` *`value=#`*`)"`

Specifies the EM variable values for current density limit calculation in the GUI.

Example:

```
emVariableValues "(variable=configA value=20) (variable=configB
value=10)"
```

Where,

the values of configA and configB variables are 20 and 10, respectively.

Note: The EM variable values passed through the configuration file will get priority over the value of same variable specified in the techfile.

enableContextAwareSrmsRelaxation

`enableContextAwareSrmsRelaxation true | false`

When set to `true`, this option enables the context-aware RMS Relaxation flow. By default, the variable is set to `false`.

Example:

```
enableContextAwareSrmsRelaxation true
```

You can also set this variable in the EMIR control file by using the following command:

```
emirutil enableContextAwareSrmsRelaxation=true
```


© 2020 All Rights Reserved.

Note: The context-aware RMS Relaxation flow is foundry-specific. For more information, contact your Cadence representative.

enableDesignResistorPeakFactor

Specifies the addition of the factor, $1/(r)^{1/2}$, in the current limit calculation for the AC-Peak analysis of the design resistor.

By default, the software calculates the current limit for the AC-Peak analysis of the design resistor using the following equation:

$$\mathbf{I}_{peak_ac(limit)} = \mathbf{I}_{peak_dc(limit)}$$

Apply the factor, $1/(r)^{1/2}$,in your calculation, then set the `enableDesignResistorPeakFactor` variable to `true`. The current limit will then be calculated in the following manner:

$$\mathbf{I}_{peak_ac(limit)} = \mathbf{I}_{peak_dc(limit)}/(r)^{1/2}$$

By default, this variable is set to `false`.

enableDesignResistorScaling

Enables the scaling of the width and length values based on the `layout_scale` factor specified in the technology file before the EM rules are applied for EM analysis of design resistors. By default, this variable is set to `true`. In the GUI, you can set this variable in the Variables form.

In batch mode, you can set this variable in the `vfibatch` command file by using the following command:

```
set_variable enableDesignResistorScaling false
```

You can also set this variable in the EMIR control file by using the following command:

```
emirutil enableDesignResistorScaling=false
```

enablePWLDensityInterpolation

Enables the current density calculation from the given PWL using current density and width, or current density and area linear interpolation for either the metal's width or the via's area. The calculation is only done when the metal’s width or the via’s area does not match the

© 2020 All Rights Reserved.

PWL width or area values.

By default, this variable is set to `false`. In the GUI, you can set this variable in the Variables form.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil enablePWLDensityInterpolation=true
```

enableReportScaleParam

When set to `true`, this option reports the layout scale factor in the EM text report. In addition, the scaled dimensions for the `path-length`, `width`, `needed Width`, and `viaArea` fields are displayed in the EM analysis report and GUI.

When set to `false`, the drawn dimension values for the `path-length`, `width`, `needed Width`, and `viaArea` fields are displayed in the EM analysis report and GUI. By default, the variable is set to `false`. In the GUI, you can set this variable in the Variables form.

Example:

```
enableReportScaleParam true
```

You can also set this variable in the EMIR control file by using the following command:

```
emirutil enableReportScaleParam=true
```

enableRMSHotSpotWaiving

Enables the RMS EM hot spot waiving analysis. By default, this feature is disabled. In the GUI, you can set this variable in the Variables form. In batch mode, you can set this variable in the `vfibatch` command file by using the following command:

```
set_variable enableRMSHotSpotWaiving true
```

You can also set this variable in the EMIR control file by using the following command:

```
emirutil enableRMSHotSpotWaiving=true
```

Note: This variable is applicable only to the RMS EM hot spot waiving analysis. To use this feature, contact your Cadence representative to explain your usage requirements.

© 2020 All Rights Reserved.

enableSHHeatSink

When set to `true`, specifies that the heat sink effect should be included in the SHE analysis. If this variable is set to `true`, ensure that heat sink models are available.

Note: This variable is only used for the SHE flow with heat sink effect.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil enableSHHeatSink=true
```

extendHeatSinkRange

Specifies the expanded region of the OD. The heat-sink structure inside this region can only act as heat sink for the specified OD shape.

The default value of this variable is 0.4.

Note: This variable is only used for the SHE flow with heat-sink effect.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil extendHeatSinkRange=0.5
```

extendHRRange

Specifies the range of the design resistor shape expansion. The unit of this variable is um (micro meter).

Note: This parameter is applicable only to the thermal-aware DC EM flow.

The default value is 1.8 although it may be different for different process nodes.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil extendHRRange=2.0
```

extendJouleHeating

Specifies whether or not metal-shape expansion is to be done for joule-heating calculation. By default, this variable is set to `false`.

Note: This variable is applicable only to the thermal-aware DC EM flow.

© 2020 All Rights Reserved.

extendODRange

Specifies the range of the OD shape expansion. The unit of this variable is um (micro meter). The default value of this variable is `1.8`.

Note: This variable is applicable only to the thermal-aware DC EM flow.

extendMetalRange

Specifies the number of metal pitch for metal expansion. The default value of this variable is `0.`

Note: This variable is applicable only to the thermal-aware DC EM flow.

extendPWLMatch

When the EM rules are specified in terms of PWL and the width of metal or the area of via does not match with any width/area of the PWL, the EM limit will not be calculated for this metal resistor/via.

When this variable is set to `true`, the software interpolates the limit value for the resistor/via from the selected rule PWL as follows:

The minimum of PWL i^{th} value * (metal's width or via's area) / (PWL i^{th} width or PWL `area`)

extendSHRegion

Specifies whether or not the OD shape extension is to be done. By default, this variable is set to `false`.

Note: This variable is applicable only to the thermal-aware DC EM flow.

geounit

Defines the scaling of all DSPF geometry parameters (`W, L, X, Y)`. It applies only to the `*|NET` section, and not to the `instance` section. The default value is `1.0e-6`. The unit of this variable is meter.

geounit_wl

Defines the scaling of DSPF geometry parameters W and L. It applies only to the `*|NET`

© 2020 All Rights Reserved.

section, and not to the instance section. The default value is `1.0e-6`. The unit of this variable is meter.

Note: If the `geounit_WL` and `geounit` options are specified together, the `geounit_WL` option takes precedence over the `geounit` option.

geounit_xy

Defines the scaling of DSPF geometry parameters X and Y. It applies only to the `*|NET` section, and not to the `instance` section. The default value is `1.0e-6`. The unit of this variable is meter.

Note: If the `geounit_xy` and `geounit` options are specified together, the `geounit_xy` option takes precedence over the `geounit` option.

layermap

Specifies the mapping between the layer names in the simulation database to those specified in the technology files (qrcTechFile and ICT file) for the EM analysis flow. It maps the layer names in the xDSPF file to the ICT layer names. the syntax is provided below.

```
layermap (rcxLayer=rcx_layer_name1 ictLayer=ict_layer_name1)
(rcxLayer=rcx_layer_name2 ictLayer=ict_layer_name2)
```

Where

`rcxLayer` is the layer name in the simulation database

`ictLayer` is the layer name in the ICT EM file

Example:

```
layermap [(rcxLayer=M1 ictLayer=T1) (rcxLayer=M2 ictLayer=T2)]
```

This variable can also be set using the `emirutil` command. For details, see “EMIR Control File Options Supported in Voltus-Fi-XL” in the Data Preparation chapter.

This variables can also be specified in the command file as follows:

```
set_variable layermap “(rcxLayer=rcx_layer_name1
ictLayer=ict_layer_name1) (rcxLayer=rcx_layer_name2
ictLayer=ict_layer_name2)”
```


© 2020 All Rights Reserved.

mdLayers

Specifies the list of metal diffusion layers that need to be combined for EM analysis. For example, to calculate the distance between terminal vias (`Lv2v`), for thermal analysis (SHE) and so on.

The default value is shown below.

```
MD_STI MD_OD MD_STI_SRM MD_OD_SRM M0_OD M0_STI MD_STI_CPP85 MD_OD_CPP85
```

This variable can also be set using the `emirutil` command. For details, see “EMIR Control File Options Supported in Voltus-Fi-XL” in the Data Preparation chapter.

mergeMDLayers

When set to `true`, enables combining of metal diffusion layers that are specified using the variable, `mdLayers`, for calculating the effective Blech length. By default, this variable is set to `false`, which means that no merging of metal diffusion layers is done for Blech length calculation.

This variable can also be set using the `emirutil` command. For details, see “EMIR Control File Options Supported in Voltus-Fi-XL” in the Data Preparation chapter.

multiplePeakPercentThreshold

Specifies the threshold percentage for the peak current above which the `Td` for a particular peak is added to the final `Td`. This variable is applicable only when the `dynamicACPeak` analysis is set to `multiPeak`. The default value of this variables is 5.

This variable can also be set using the `emirutil` command. For details, see “EMIR Control File Options Supported in Voltus-Fi-XL” in the Data Preparation chapter.

optimisticEMRuleSelection

Specifies whether optimistic or pessimistic values are selected for the specified rules. This is applicable when multiple rules are selected after all conditions match.

By default, the variable is set to `false`. This means that the pessimistic value is selected.

© 2020 All Rights Reserved.

powerRailGroundNets

Specifies the ground nets to be used for power-rail analysis. The available options are *`list of ground nets`* and `"" (empty string)`.

- `"" (empty string)`: This is the default option. The software automatically identifies ground nets.
- *`list of supply nets`*: Specifies the list of ground nets to be used for power-rail analysis. These nets will override the nets automatically identified by the software.

Note: This variable accepts wildcards. For example:

```
powerRailGroundNets="vss vssp*"
```

powerRailRules

Specifies rules for power-rail analysis. The available options are `off, default, n5, n7,` and `advance.`

- `off`: The software ignores all EM rules specified using the `power_rail` keyword.
- `default`: Enables power-rail analysis.
- `n5`: Sets the power-rail analysis rules for the N5 process node.
- `n7`: Sets the power-rail analysis rules for the N7 process node.
- `advance`: Specifies the following:
  - EM rules will not be used for metal layers.
  - Stack vias will be considered as part of terminal vias.
  - Current direction check will not be performed for terminal vias.

powerRailSupplyNets

Specifies the supply nets to be used for power-rail analysis. The available options are *`list of supply nets`* and `"" (empty string)`.

- `"" (empty string)`: This is the default option. The software automatically identifies supply nets.
- *`list of supply nets`*: Specifies the list of supply nets to be used for power-rail analysis. These nets will override the nets automatically identified by the software.

Note: This variable accepts wildcards. For example:

© 2020 All Rights Reserved.

```
powerRailSupplyNets="vdd vddp*"
```

process_scale

Specifies the value by which the width and length values will be multiplied before the EM rules are applied. The default value of this variable is 1. In the GUI, you can set this variable in the Variables form.

In batch mode, you can set this variable in the `vfibatch` command file by using the following command:

```
set_variable process_scale 0.8
```

You can also set this variable in the EMIR control file by using the following command:

```
emirutil process_scale=0.8
```

prioritizedPwlRulesLayers

Specifies the list of layers for which area-based PWL search is performed first. When no match is found, then non-PWL rules are used. The default value of this variable is `"all"`, which means that this behavior is enabled for all layers.

To enable the behavior for specific layers, you can specify a list of layers.

To disable this behavior for all layers, set the value of this variable to `none`. In the GUI, you can set this variable in the Variables form.

In batch mode, you can set this variable in the `vfibatch` command file by using the following command:

```
set_variable prioritizedPwlRulesLayers none
```

You can also set this variable in the EMIR control file by using the following command:

```
emirutil prioritizedPwlRulesLayers=none
```


© 2020 All Rights Reserved.

removeBulkShorting

When set to `true`, this variable removes bulk-shorting in the design. Bulk-shorting in the design results in an increase in the EM analysis runtime. This variable is only used as a troubleshooting option.

By Default, this variable is set to `false`.

The syntax is as follows:

```
removeBulkShorting true | false
```

Example:

```
removeBulkShorting true
```

You can also set this variable in the EMIR control file by using the following command:

```
emirutil removeBulkShorting=true
```

reportUnmatchedRuleResistor

When set to `true`, specifies that resistors that do not match any EM rule for the specified analysis in the ICT file will be reported in the `"#.rpt_unmatch"` file. The default value of this variable is `true`. Also, the summary report file, `summary.rpt`, will report a count of such resistors in a separate column. For example, `"iacpeak_num_unmatch"`.

If you do not want to create the `#.rpt_unmatch` file, set this variable to `false`. For example:

```
emirutil reportUnmatchedRuleResistor=false
```

rmsMetalLineNumberPower

Specifies the value of power on the metal lines for RMS EM analysis of power nets.

If this option is not specified, the default value of `0` is considered, which means the software uses the `em_dc_rms_metal_line_num` value specified in the ICT file.

However, if this option is specified, it overrides the `em_dc_rms_metal_line_num` value specified in the ICT file.

You can also set this variable in the conf file in the following manner:

```
emirutil rmsMetalLineNumberPower=number
```


© 2020 All Rights Reserved.

The default value is `0`.

rmsMetalLineNumberSignal

Specifies the value of power on the metal lines for RMS EM analysis of signal nets.

If this option is not specified, the default value of `0` is considered, which means the software uses the `em_ac_rms_metal_line_num` value specified in the ICT file.

However, if this option is specified, it overrides the `em_ac_rms_metal_line_num` value specified in the ICT file.

You can also set this variable in the conf file in the following manner:

`emirutil rmsMetalLineNumberSignal=`*`number`*

The default value is `0`.

rmsMultiplierPin

Specifies the RMS relaxation factor for EM analysis of power grids. The default value is `1.0`.

rmsMultiplierSignal

Specifies the RMS relaxation factor for EM analysis of signal nets. The default value is `1.0`.

shEMEffect

Specifies the basis for the re-evaluation of EM effect. Valid values are:

- `beolT`: specifies that re-evaluation will be done on the basis of individual `metal BEOL_T`.
- *`tiles_no.of divisions in x-direction X no. of divisions in y-direction`*: specifies that re-evaluation will be done on the basis of `tile BEOL_T`.

shParamFile

Specifies the self-heating effect (SHE) analysis parameter file, which is required for plotting the SHE analysis plots. This file is used for deltaT FEOL calculation with finger and fin effect considerations.

© 2020 All Rights Reserved.

Note: For SHE with heat-sink effect, this file should include the heat-sink rules.

shXMLFile

Specifies the XML file to be used in the SHE analysis flow. This file is created by model cards with supporting self-heating as an input file for the final deltaT FEOL value calculation during simulation.

simpleSingleViaDefinition

The keyword, `single` is used to differentiate between a single square cut via and square via array. It is applicable to square via, with width equal to '`vc_width/min_width`', only. To support the `single` keyword for all the via sizes, set the `simpleSingleViaDefinition` variable to `true`.

By default, this variable is set to `false`.

You can also set this variable in the conf file in the following manner:

```
emirutil simpleSingleViaDefinition=true
```

For details of the supported EMIR control file options, see “EMIR Control File Options Supported in Voltus-Fi-XL” in the “Data Preparation” chapter.

splitACDCRules

Specifies different rules for EM analysis of power and signal nets. The default value is `false`. For details, see Rules for Specifying the EM Analysis Type for Power and Ground Nets in the “EM Rules Specification” chapter.

Tj

Specifies the temperature to be used for EM analysis. The default value is `Tsim` or the temperature used while performing simulation.

Tambient

Specifies the ambient temperature for self-heating analysis. If this value is not specified, the software calculates it as `Tj-deltaT`.

© 2020 All Rights Reserved.

You can also set this variable in the EMIR control file by using the following command:

```
emirutil Tambient=value
```

Tlife

Specifies the lifetime for which the EM analysis will be performed. The default unit for this variable is `years`. You can also specify the value in `hours`. The tool will convert it into years. For example:

```
Tlife="20000hours"
```

toggleNetFrequency

Specifies the frequency of the nets so that the tool can derive the `Td` (total on-time period) for multiple clock cycles. You can specify frequencies for different nets as shown below. Wild cards are also supported.

Syntax:

```
toggleNetFrequency "(net1,frequency1) (net#,frequency#)"
```

For example,

```
toggleNetFrequency "(CLK*,2.3e9) (NET1,4.6e9)"
```

toggleNetFrequencyFile

Specifies the name of the file that contains the net names and their corresponding frequency values. This file is used to specify the frequency of nets to compute the `Td` (total on-time period) for multiple clock cycles. This file contains net names separated by commas and the frequency values.

For example,

```
toggleNetFrequencyFile freq_file
```

A sample `freq_file` is shown below.

```
VDD, 2e08
D*, 4e08
GSS, 3e08
```


© 2020 All Rights Reserved.

useEncryptedWidth

Specifies whether encrypted (silicon) width should be used or not. The default value of this variable is `false`, in which case, the software does not use the encrypted width from the DSPF. When set to `true`, the software uses the encrypted width but does not report both the width and the minimum width.

useRecoveryFactor

Specifies whether or not layer-wise recovery factor is to be used for calculating the average current density.

If the recovery factor is specified, then the average current density calculation is as follows:

```
iavg=max(|iavgpos|,|iavgneg|)- recovery_factor*min(|iavgpos|,|iavgneg|)
```

If not specified, then

```
iavg=iavgpos+iavgneg
```

where `iavgpos` is the average current in the positive direction and `iavgneg` is the average current in the negative direction

viaCountFromResistance

Specifies the number of vias when third-party DSPF with missing via count ($N) is used for EM analysis. The default value of this variable is `false`. To understand the scenario in which this variable is used, see Appendix A-Third Party DSPF.

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

19

SKILL Functions

- Overview on page 682
- Voltus-Fi SKILL Functions on page 682

© 2020 All Rights Reserved.

Overview

This section provides syntax, description, and examples for the Cadence® SKILL functions associated with the Voltus™-Fi Custom Power Integrity Solution (Voltus-Fi) flow.

Only the functions documented in this chapter are supported for public use. Any other functions, and undocumented aspects of the functions described below, are private and subject to change or removal at any time.

Voltus-Fi SKILL Functions

The following types of functions are provided in Voltus-Fi:

- vsaCreateMarker
- vsaOpenLayout
- vsaLoadNets
- vsaLoadSecondaryLayers
- vsaSetLayersToMergeDuringResultsLoading
- vsaSetInputType
- vsaSetEMIRConfig
- vsaSetEMResultsFile
- vsaSetEMLayerMapFileName
- vsaSetEMTechFileName
- vsaSetEMTypes
- vsaSetDfiiLayerMapFileName
- vsaShowAllResistors
- vsaSetQRCData
- vsaSelectFailedNets
- vsaSetEMOnlyICTFile
- vsaSetEMPlot
- vsaLoadEM

© 2020 All Rights Reserved.

- vsaSetIRResultsFile
- vsaSetIRThreshold
- vsaSetIRPlot
- vsaSetLRPShortedLayers
- vsaLoadIR
- vsaSetFinerDisplayOff
- vsaSetParallelLoading

© 2020 All Rights Reserved.

vsaCreateMarker

```
vsaCreateMarker(
     x x_coordinate
     y y_coordinate
     message
     )
     => t / nil
```

Description

Creates a point marker with the specified attributes. When this marker is created successfully, it appears in the *Misc* tab of Annotation Browser.

Arguments

`x`	Specifies the x-axis coordinate for the marker.
`y`	Specifies the y-axis coordinate for the marker.
`message`	Specifies the message associated with the marker.

Values Returned

`t`	Marker was created at the specified coordinates.
`nil`	Returns `nil` when the marker is not created.

Example

```
vsaCreateMarker(34.45 56.67 "Marker1")
=> t
```

The above example code shows how to create `Marker1` at the specified x-axis and y-axis coordinates.

© 2020 All Rights Reserved.

vsaOpenLayout

```
vsaOpenLayout(
     libname
     cellname
     viewname
     )
```

Description

Opens the design in Virtuoso for running Voltus-Fi-XL.

Arguments

libname	Specifies the library name for the design to be opened.
cellname	Specifies the cell name for the design to be opened.
viewname	Specifies the layout view name for the design to be opened.

Values Returned

None.

Example

```
vsaOpenLayout("testlib" "testcell" "layout")
```

The above example code shows how to specify the library name, cell name, and layout view name of the design you want to open.

© 2020 All Rights Reserved.

vsaLoadNets

```
vsaLoadNets(
     netname
     )
```

Description

Specifies the nets to be loaded in the Voltus-Fi GUI.

Argument

netname	Specifies the name of the net to be loaded in the Voltus-Fi GUI. You can specify multiple net names separated by a space.

Values Returned

None.

Example

```
vsaLoadNets("VDD VSS")
```

The above example code shows how to specify the net names for the nets, `VDD` and `VSS`, that you want to load in the Voltus-Fi GUI.

© 2020 All Rights Reserved.

vsaLoadSecondaryLayers

```
vsaLoadSecondaryLayers()
```

Description

Enables the loading of data for all layers in the Voltus-Fi GUI.

Argument

None.

Values Returned

None.

Example

```
vsaLoadSecondaryLayers()
```

The above example code shows how to enable the loading of data for all layers in the Voltus-Fi GUI.

© 2020 All Rights Reserved.

vsaSetLayersToMergeDuringResultsLoading

```
vsaSetLayersToMergeDuringResultsLoading(
     metal_layer
     cut_layer
     )
```

Description

Specifies the name of the metal layer and the corresponding cut-layer to be excluded from highlighting while plotting the EMIR analysis results. This function can be called multiple times to specify different pairs of metal and cut layers. Voltus-Fi honors all these calls. This function must be called before loading results in Voltus-Fi.

Arguments

metal_layer	Specifies the name of the metal layer for which the solid shapes are to be highlighted.
cut_layer	Specifies the name of the corresponding cut-layer to be excluded from highlighting while plotting the results.

Values Returned

None.

Examples

```
vsaSetLayersToMergeDuringResultsLoading("M1 CM1A")
vsaSetLayersToMergeDuringResultsLoading("PO CPO")
```

The above examples show how to specify the cut layers, `CM1A` and `CPO` for layers, `M1` and `PO`, respectively.

© 2020 All Rights Reserved.

vsaSetInputType

```
vsaSetInputType(
     inputType
     resultsFile
     resultsType
     startTime
     stopTime)
     )
```

Description

Specifies the data base input options, parameter storage format (PSF) or vavo_db, for running IR drop and EM analyses in the Voltus-Fi-L GUI. The *resultsType*, *startTime* and *stopTime* arguments are only valid for the psf-based flow.

Arguments

inputType	Specifies whether PSF or vavo_db is used for the EMIR analysis.
resultsFile	For the psf-based flow, specifies the analysis name for DCOP (`dcOp-dc`) or transient (`tran`) analysis. For the vavo_db flow, specifies the name of the vavo_db simulation results file.
resultsType	For the psf-based flow, specifies the type of analysis, `dc` or `tran`. For the vavo_db- based flow, `nil` is specified.
startTime	Specifies the analysis start time for the psf-based flow, for both DCOP and transient analysis types. For the vavo_db-based flow, `0` is specified. **Note:** This argument must be specified along with the *stopTime* argument.

© 2020 All Rights Reserved.

stopTime Specifies the stop time for analysis for the psf-based flow, for both DCOP and transient analysis types.

For the vavo_db- based flow, `0` is specified.

Note: This argument must be specified along with the *startTime* argument.

Values Returned

None.

Examples

- `vsaSetInputType("psf" "dcOp-dc" "dc" "0" "170")`

  The above example code shows how to specify psf database, DCOP analysis type, dc analysis name, with a start time of 0s and stop time of 170s in the Voltus-Fi-L GUI.

- `vsaSetInputType("vavodb" "tran_emir_vavo.db" "nil" "0" "0")`

  The above example code shows how to specify vavodb as the vavo_db database and `tran_emir_vavo.db` as the simulation results directory.

© 2020 All Rights Reserved.

vsaSetEMIRConfig

```
vsaSetEMIRConfig(
     config_filename
     )
```

Description

Loads the EMIR configuration file to be used for EMIR analysis in Voltus-Fi-XL. This file is specified before calling the `vsaLoadEM()` function.

Argument

`config_filename`	Specifies the name of the `EMIR` configuration file to be loaded for EMIR analysis in Voltus-Fi-XL.

Values Returned

None.

Example

```
vsaSetEMIRConfig("abc/emir.conf.variables")
vsaLoadEM()
```

The above example code shows that the `emir.conf.variables` file will be loaded for EMIR analysis in Voltus-Fi-XL.

© 2020 All Rights Reserved.

vsaSetEMResultsFile

```
vsaSetEMResultsFile(
     binfilename
     )
```

Description

Specifies the simulation results file to be used for viewing the EMIR analysis results in Voltus-Fi-XL. This file contains the simulation database generated using the Spectre simulator.

Argument

binfilename	Specifies the name of the simulation results file (`emir_bin`) to be used for loading and displaying the EMIR analysis results in Voltus-Fi-XL.

Values Returned

None.

Example

```
vsaSetEMResultsFile("xps.raw/xps.emirtap.emir0_bin")
```

The above example code shows that the name of the results file specified for loading and displaying EMIR analysis results in Voltus-Fi-XL is `xps.emirtap.emir0_bin`.

© 2020 All Rights Reserved.

vsaSetEMLayerMapFileName

```
vsaSetEMLayerMapFileName(
     layermapfile
     )
```

Description

Specifies the name of the layer map file to be used for the EM technology file. This file provides the mapping between layer names in the simulation database and those in the technology file. This function is only used for EM plots.

Argument

layermapfile	Specifies the name of the layer map file. This file provides the mapping between layer names in the simulation database and those in the technology file.

Values Returned

None.

Example

```
vsaSetEMLayerMapFileName("contactmapfile")
```

The above example code shows that the layer map filename is specified as `contactmapfile`.

© 2020 All Rights Reserved.

vsaSetEMTechFileName

```
vsaSetEMTechFileName(
     techfilename
     )
```

Description

Specifies the name of the EM technology file that contains the EM rule information to be used for plotting EM results in Voltus-Fi.

Argument

techfilename	Specifies the name of technology file with EM rules to be used for viewing EM results in Voltus-Fi. You can specify `qrcTechFile`, `emDataFile`, or ICT file as input for performing EM analysis.

Values Returned

None.

Example

```
vsaSetEMTechFileName("qrcTechFile")
```

The above example code shows that the technology filename is specified as `qrcTechFile`.

© 2020 All Rights Reserved.

vsaSetEMTypes

```
vsaSetEMTypes(
     list (imax iavg irms iacpeak iavgabs)
     )
```

Description

Specifies the types of EM analyses for which you want to plot EM results in Voltus-Fi. At least one analysis type must be selected.

Argument

`list (imax iavg irms iacpeak iavgabs)`	Specifies the types of EM analyses for which you want to plot EM results. The list can include the following types of analyses: `imax`: calculates the current density violations based on maximum DC current `iavg`: calculates the current density violations based on the average value of DC current `irms`: calculates the current density violations based on the root mean square (RMS) value of AC current `iacpeak`: calculates the current density violations based on peak AC current `iavgabs`: calculates the current density violations based on the average value of absolute current

Values Returned

None.

Example

```
vsaSetEMTypes(list("imax" "iavg" "iavgabs"))
```

The above example code shows that `imax`, `iavg`, and `iavgabs` types of EM analyses are to be performed.

© 2020 All Rights Reserved.

vsaSetDfiiLayerMapFileName

```
vsaSetDfiiLayerMapFileName(
     df2layermapfile
     )
```

Description

Specifies the DFII layer map filename that provides the mapping between the extractor-generated xDSPF file layer names and the DFII layer names.

Argument

df2layermapfile	Specifies the name of the DFII layer map file that provides the mapping between the xDSPF file layer names and the DFII layer names.

Values Returned

None.

Example

```
vsaSetDfiiLayerMapFileName("df2layermap")
```

The above example code shows that the name of the DFII layer map file is `df2layermap`.

© 2020 All Rights Reserved.

vsaShowAllResistors

```
vsaShowAllResistors()
```

Description

When specified, displays all resistors on the Voltus-Fi GUI including those for which there are no layout shapes matching the specified resistor coordinates. These resistors are displayed using stick diagrams.

Argument

None.

Values Returned

None.

Example

```
vsaSetDfiiLayerMapFileName("df2layermap")
vsaShowAllResistors()
vsaSetIRResultsFile("test/xps.617/xps.raw/xps.emirtap.emir0_bin")
vsaLoadIR();
```

The above example code shows how to view all resistors including those for which there is no layout shape matching the specified resistor coordinates on the Voltus-Fi GUI.

© 2020 All Rights Reserved.

vsaSetQRCData

```
vsaSetQRCData(
     rundir
     runname
     )
```

Description

Sets the Quantus inputs to enable solid shape highlighting for EMIR plots displayed on the Virtuoso layout.

Arguments

rundir	Specifies the Quantus run directory name.
runname	Specifies the run name used while running Quantus.

Values Returned

None.

Example

```
vsaSetQRCData("./qrc_temp" "testrun")
```

The above example code shows that Quantus run directory, `qrc_temp` and run name, `testrun` will be used to enable solid shape highlighting.

© 2020 All Rights Reserved.

vsaSelectFailedNets

```
vsaSelectFailedNets()
```

Description

Specifies that EM analysis results for only those nets that failed the EM check should be plotted. It enables the *Show Only Failed* option in the GUI.

Arguments

None.

Values Returned

None.

Example

```
vsaSelectFailedNets ()
```

The above example code shows the EM analysis results for only those nets that fail the EM check will be plotted.

© 2020 All Rights Reserved.

vsaSetEMOnlyICTFile

```
vsaSetEMOnlyICTFile(
     filename
     )
```

Description

Specifies the name of the ICT file to be used for specifying the EM rules for plotting EM analysis results.

Argument

filename	Specifies the name of the ICT file.

Values Returned

None.

Example

```
vsaSetEMOnlyICTFile("./myICT.txt")
```

The above example code shows that `myICT.txt` file will be used for specifying the EM rules.

© 2020 All Rights Reserved.

vsaSetEMPlot

```
vsaSetEMPlot(
     plot_type
     )
```

Description

Specifies the default plot for EM analysis results in Voltus-Fi.

Argument

plot_type	Sets the default plot for EM analysis results. You can specify any one of the following plot types: `RJ`: sets the default plot to DC current density analysis plot `JAVG`: sets the default plot to average current density analysis plot `JACPEAK`: sets the default plot to AC peak current density analysis plot `JABSAVG`: sets the default plot to average absolute current density analysis plot `JACRMS`: sets the default plot to RMS current density analysis plot `TC`: sets the default plot to peak tap current density analysis plot `RC`: sets the default plot to peak resistor current density analysis plot

Values Returned

None.

Example

```
vsaSetEMPlot("JAVG")
```


© 2020 All Rights Reserved.

The above example code shows that the default plot for EM analysis results will be average current density analysis data (`JAVG`) plot.

© 2020 All Rights Reserved.

vsaLoadEM

```
vsaLoadEM()
```

Description

Specifies that the EM analysis results should be loaded. Before specifying this function, ensure that the EM results file and the EM technology file are specified.

Arguments

None.

Values Returned

None.

Example

```
vsaOpenLayout("testlib" "testcell" "layout")
vsaSetEMResultsFile("./xps.raw/xps.emirtap.emir0_bin")
vsaSetEMTechFileName("qrcTechFile")
vsaSetEMLayerMapFileName("contactmapfile")
vsaSetEMTypes(list("imax" "iavg" "irms" "iacpeak" "iavgabs"))
vsaSetDfiiLayerMapFileName("df2layermap")
vsaLoadEM()
```

The above example code shows the set of functions specified for loading the EM analysis results. The `vsaSetEMResultsFile` and the `vsaSetEMTechFileName` functions are required to be specified before specifying `vsaLoadEM`.

© 2020 All Rights Reserved.

vsaSetIRResultsFile

```
vsaSetIRResultsFile(
     binfilename
     )
```

Description

Specifies the results file to be used for viewing the EMIR analysis results in Voltus-Fi-XL.

Argument

binfilename	Specifies the name of the `emir_bin` file to be used for loading and displaying the EMIR analysis results in Voltus-Fi-XL.

Values Returned

None.

Example

```
vsaSetIRResultsFile("xps.raw/xps.emirtap.emir0_bin")
```

The above example code shows that the name of the results file to be used for loading and displaying EMIR analysis results in Voltus-Fi-XL is `xps.emirtap.emir0_bin`.

© 2020 All Rights Reserved.

vsaSetIRThreshold

```
vsaSetIRThreshold(
     threshold_value
     )
     => nil
```

Description

Sets the threshold value for viewing IR drop violations.

Argument

threshold_value	Specifies the threshold value for viewing the IR drop violations in Voltus-Fi.

Value Returned

`nil`	Returns `nil` when the value is not specified.

Example

```
vsaSetIRThreshold(0.01)
```

The above example code sets the threshold value to `0.01` volts and all IR drop values below `0.01` volts are filtered while plotting the IR drop results.

© 2020 All Rights Reserved.

vsaSetIRPlot

```
vsaSetIRPlot(
     plot_type
     )
```

Description

Specifies the default plot for IR drop analysis results in Voltus-Fi.

Argument

plot_type	Sets the default plot for IR drop analysis results. Specify one of the following plot types: `IR`: sets the default plot to peak IR drop analysis plot `IRAVG`: sets the default plot to average IR drop analysis plot `RC`: sets the default plot to peak current density violation plot `IV`: sets the default plot to transistor-based supply voltage data plot `Reffective`: sets the default plot to effective resistance analysis plot `PI`: sets the default plot to that for currents across resistors when power gates are connected `PV`: sets the default plot to that for IR drop across power-switch instances

Values Returned

None.

Example

```
vsaSetIRPlot("Reffective")
```


© 2020 All Rights Reserved.

The above example code shows that the default plot for IR drop analysis will be `Reffective`.

© 2020 All Rights Reserved.

vsaSetLRPShortedLayers

```
vsaSetLRPShortedLayers(
     layer1, layer2,...
     )
```

Description

Specifies the names of layers to be shorted for the least-resistive path (LRP) analysis.

Argument

(*layer1, layer2,....*)	Specifies the names of layers to be shorted for LRP analysis.

Values Returned

None.

Example

```
vsaSetLRPShortedLayers("M1, OD")
```

The above example code shows that layers, `M1` and `OD` will be shorted for LRP analysis.

© 2020 All Rights Reserved.

vsaLoadIR

```
vsaLoadIR()
```

Description

Specifies that the IR drop analysis results should be loaded. Before specifying this function, ensure that the IR drop results filename is specified.

Arguments

None.

Values Returned

None.

Example

```
vsaOpenLayout("testlib" "testcell" "layout")
vsaSetIRResultsFile("./xps.raw/xps.emirtap.emir0_bin")
vsaSetDfiiLayerMapFileName("df2layermap")
vsaLoadIR()
```

The above example code shows the set of functions specified for loading the IR drop analysis results. The `vsaSetIRResultsFile` function is required to be specified before specifying `vsaLoadIR`.

© 2020 All Rights Reserved.

vsaSetFinerDisplayOff

```
vsaSetFinerDisplayOff()
```

Description

Disables the finer gradient for the specified layers while displaying the IR/EM plots on the Virtuoso layout. This function is specified after specifying the DFII layermap file and before loading the IR drop or EM analysis results.

Arguments

None.

Values Returned

None.

Example

```
vsaSetDfiiLayerMapFileName("df2layermap")
vsaSetFinerDisplayOff()
vsaSetIRResultsFile("test/xps.617/xps.raw/xps.emirtap.emir0_bin")
vsaSetIRThreshold(0.01)
vsaLoadIR()
```

The above example code shows how to specify the DFII layermap file, disable the finer gradient display and then load the IR drop analysis results.

© 2020 All Rights Reserved.

vsaSetParallelLoading

```
vsaSetParallelLoading()
```

Description

Enables parallel loading of OpenAccess shapes in the Voltus-Fi application interface.

Arguments

None

Values Returned

None

Example

```
vsaSetParallelLoading()
```


© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

20

File Formats

- Static Current Analysis Files on page 714
  - Activity File on page 715
  - Clock File on page 716
  - Value Change Dump (VCD) File on page 717
  - Power File on page 720
- Rail Analysis Files on page 721
  - ICT File on page 722
  - EM Only ICT File on page 725
  - EM Data File on page 727
  - Layer Map File on page 729
  - DFII Layer Map File on page 731
- Self-Heating Effect Analysis File on page 733
  - shParamFile on page 733
- Power-Grid View Creation Files on page 735
  - Format and Contents of Trigger File on page 735
  - LEF layer Map File on page 738
  - PGDB layer Map File on page 739
- Common Command Language (CCL) File on page 741

© 2020 All Rights Reserved.

Static Current Analysis Files

This section provides information about the files required for performing average static current analysis in Voltus-Fi-XL using Thunder.

- Activity File
- Clock File
- Value Change Dump (VCD) File
- Power File

© 2020 All Rights Reserved.

Activity File

A file that contains the activity factor on any node of the circuit. This file is used for average current analysis. In the GUI, you can specify the activity file in the *Advanced* tab of the *Setup and Run Current Analysis* form.

Sample filename: `activity.txt`

Command: `run_static_simulation -activity_file` *`file_name`*

Format Description:

The activity file has the following format:

```
file     :rows EOF                        //a file consists of rows of information
rows     :ε                               //an empty line
         |comment                        //a comment line
         |line                           //a specification line
comment  :^[\t]*\*.* EOL                  //a line starting with * is a comment
line     :name FA_value                   //a node with transitional frequency
```

- *`Name`* can be the name of any node in the circuit.
- *`FA_value`* specifies the default nodal activity to be used. It is given as an activity factor. A value of `0.20` would be `20%` of the default frequency.

Example: This is a sample `fa_file` for node activities. The column on the left shows the node names and the column on the right shows the activity.

```
out01   2
out02   3
out03   4
```


© 2020 All Rights Reserved.

Clock File

A file that contains information about the clock nodes, subcircuit definitions, and subcircuit instance names and their frequencies. This file is only used for average static current analysis. In the GUI, you can specify the average power file in the *Advanced* tab of the *Setup and Run Current Analysis* form.

Sample filename: `clk.txt`

Command: `run_static_simulation -clock_file` *`clock_file_name`*

Format:

```
file: rows EOF //a file consists of rows
rows: ε //an empty line
     | comment //a comment
     | line //a specification line
comment:^[\t]*\*.+EOL //a line starting with a *

line :CLK name freq_value //a clock node specification
     |INST name freq_value //a subcircuit instance specification
     |INST name freq_value activity //a subcircuit instance with activity
     |CELL name freq_value //a subcircuit definition specification
     |CELL name freq_value activity //a subcircuit definition with activity
```

Format Description:

`CLK`, `INST`, and `CELL` are keywords.

- If you use the `CLK` keyword, the node name is treated as the root of a clock running at the given frequency.
- If you use the `INST` keyword, all input and output nodes—that is, node names on the instance card—are set to toggle at the specified frequency. .
- If you use the `CELL` keyword, all instances of the cell are processed instead of a single instance.

`CLK` has precedence over `INST`, and `INST` overrides `CELL` information to resolve redundant specifications.

Example: The following example specifies the `CLK` keyword in column 1, clock name in column 2, and the frequency value in column 3.

```
clk    clk_internal    120M
```


© 2020 All Rights Reserved.

Value Change Dump (VCD) File

A file that contains information about the number of transitions for each net. This file is only used for average static current analysis. In the GUI, you can specify the VCD file in the *Advanced* tab of the *Setup and Run Current Analysis* form.

Sample filename: `vcd.txt`

Command: `run_static_simulation -vcd` *`vcd_file_name`*

VCD Values

Within the VCD file, these VCD values have the following meanings during driving:

- `0`

  Forces the associated node to logic low.

- `1`

  Forces the associated node to logic high.

- `X`

  Forces the associated node to logic unknown.

- `Z`

  Releases the associated node.

VCD Command Extensions

The following extensions of the VCD format are supported. You can use these command extensions in the comment section of a VCD file to specify node characteristics.

Use these extensions to specify properties such as input voltages and rise and fall times of nodes listed in the VCD file.

- `*scope`

  Sets the scope of the `nodeset` command by using the `scope` command. *`Scope_string`* is prepended to the nodes listed in the `nodeset` command. The syntax is as follows:

  ```
  *scope scope_string *end
  ```

- `*nodeset`

  Groups nodes in the VCD file into node sets. You can then set properties of the nodes in the node set by using the `params` command on the node set. The syntax is as follows:

© 2020 All Rights Reserved.

```
*nodeset set_number [node_name]+ *end
```

- `*params`

  Assigns parameters to sets of nodes by using the `params` command. The syntax is as follows:

  ```
  *params set_number <vin0=voltage> <vin1=voltage> <vinx=voltage><tir=time>
  <tif=time> *end
  ```

Example:

Following is a sample VCD file named `vcd`.

```
$date
        Thurs Mar 07 15:09:59 2013
$end
$version
        Cadence 1.0
$end
$timescale
        1ns
$end

$scope module TOP $end

$var wire     1 !     A[7] $end
$var wire     1 "     A[6] $end
$var wire     1 #     A[5] $end
$var wire     1 $     A[4] $end
$var wire     4 %     A[3:0] $end
$upscope $end

$comment

*scope TOP *end
*nodeset 1 A[7:0] *end
*params 1 vin0=0v vin1=5v vinx=2.5v tir=1ns tif=1ns *end

$end

$enddefinitions $end

#0
1!
```


© 2020 All Rights Reserved.

```
1"
1#
1$
b1111 %
#50
0!
0"
0#
0$
b0000 %
#50
1$
b0001 %
#50
1#
0$
b0010 %
#50
1$
b0011 %
#50
1"
0#
0$
b0100 %
```


© 2020 All Rights Reserved.

Power File

A file that contains the average power for specific sub circuits. This file is only used for average static current analysis. In the GUI, you can specify the average power file in the *Advanced* tab of the *Setup and Run Current Analysis* form.

Sample filename: `power.txt`

Command: `run_static_simulation -power_file` *`power_file_name`*

Format and Example

The average power file has the format shown in the following example:

```
CELL C_MUX2_TG 2.0e-4
Xxev | xeu | xem | wpy | sdh 1.0e-3
```

The `CELL` keyword specifies that the name following it is a subcircuit definition and that all instances of the cell dissipate the same average power. If the file does not contain the CELL keyword, the name is an instance of a subcircuit dissipating the specified average power.

© 2020 All Rights Reserved.

Rail Analysis Files

This section lists the rail analysis file formats.

© 2020 All Rights Reserved.

ICT File

The ICT file specifies the EM reliability rules for EM analysis. This is a text-based file, which means you can use any text editor to enter information in this file.

Following is a sample ICT file.

```
process "abcd" {
      background_dielectric_constant 1
      temp_reference  25
   em_tref 110
   em_output_wlt drawn
   em_variables  w delta_T
   em_conductor_unit mA/um
   em_via_area_unit  mA
   em_via_unit mA
   em_dc_rms_metal_line_num 3
   em_ac_rms_metal_line_num 3
      simulation {
        manual_simulation_points {
          metal_layer "mt1" {
            simulation_widths    0.3 0.6 1.5 3
            simulation_spacings  0.3 0.6 1.5 3
          }
          metal_layer "mt2" {
            simulation_widths    0.3 0.6 1.5 3
            simulation_spacings  0.3 0.6 1.5 3
          }
        }
        manual_simulation_combinations {
          combination_layers "mt1" "mt2"
          separation_values   0 1 2 3
        }
      }
```


© 2020 All Rights Reserved.

```
}
conductor "mt1" {
        min_spacing         0.3
        min_width           0.3
        height              1.05
        thickness           0.53
        resistivity         0.1
        gate_forming_layer  false
        thermal_conductivity 0.39
        em_model {
          em_jmax_dc_rms_metal_line_factor 1 1.5 2 1.5 3 1.32 4 1.32 5 1.05 6
          1.05 7 1.05 8 1.05 9 1
          em_jmax_ac_rms_metal_line_factor 1 1.5 2 1.5 3 1.32 4 1.32 5 1.05 6 1.05
          7 1.05 8 1.05 9 1
          em_jmax_dc_avg  EQU w*1.0   jmax_factor   50 1.1  110  1.0  125 0.92 L >  5
          em_jmax_dc_avg  EQU w*3.0   jmax_factor   50 1.1  110  1.0  125 0.92 L <= 5
          em_jmax_dc_peak EQU w*1.5   jmax_factor   50 1.1  110  1.0  125 0.92 L >  5
          em_jmax_dc_peak EQU w*3.5   jmax_factor   50 1.1  110  1.0  125 0.92 L <= 5
          em_jmax_dc_rms EQU 1.5 * sqrt(2*delta_T)
          em_jmax_ac_rms EQU 1.5 * sqrt(2*delta_T)
        }
}
conductor "mt2" {
        min_spacing         0.3
        min_width           0.3
        height              2.38
        thickness           0.53
        resistivity         0.1
        gate_forming_layer  false
        thermal_conductivity 0.39
        em_model {
          em_jmax_dc_rms_metal_line_factor 1 1.5 2 1.5 3 1.32 4 1.32 5 1.05 6
          1.05 7 1.05 8 1.05 9 1
```


© 2020 All Rights Reserved.

```
          em_jmax_ac_rms_metal_line_factor 1 1.5 2 1.5 3 1.32 4 1.32 5 1.05 6 1.05
          7 1.05 8 1.05 9 1

          em_jmax_dc_avg  EQU 2.0*w*1.0   jmax_factor   50 1.1  110  1.0  125 0.92
          L >  5

          em_jmax_dc_avg  EQU 2.0*w*3.0   jmax_factor   50 1.1  110  1.0  125
          0.92 L <= 5

          em_jmax_dc_peak EQU 2.0*w*1.5   jmax_factor   50 1.1  110  1.0  125 0.92
          L >  5

          em_jmax_dc_peak EQU 2.0*w*3.5   jmax_factor   50 1.1  110  1.0  125 0.92
          L <= 5

          em_jmax_dc_rms EQU 1.5 * sqrt(2*delta_T)

          em_jmax_ac_rms EQU 1.5 * sqrt(2*delta_T)

        }

}

via "Via1" {

        top_layer         "mt2"

        bottom_layer      "mt1"

        contact_resistance 1.1

        thermal_conductivity 0.39

        em_model {

          em_jmax_dc_rms_metal_line_factor 1 1.5 2 1.5 3 1.32 4 1.32 5 1.05 6
          1.05 7 1.05 8 1.05 9 1

          em_jmax_ac_rms_metal_line_factor 1 1.5 2 1.5 3 1.32 4 1.32 5 1.05 6 1.05
          7 1.05 8 1.05 9 1

          em_vcwidth  0.12

          em_jmax_dc_avg   1.0  jmax_factor  50 1.1 110 1.0 125 0.97

          em_jmax_dc_peak  1.5  jmax_factor  50 1.1 110 1.0 125 0.97

          em_jmax_dc_rms EQU 1.5 * sqrt(2*delta_T)

          em_jmax_ac_rms EQU 1.5 * sqrt(2*delta_T)

}
```


© 2020 All Rights Reserved.

EM Only ICT File

The EM only ICT file provides the process and the EM model information for EM analysis. This is a text-based file, which means you can use any text editor to enter information in this file.

Following is a sample EM only ICT file.

```
process "gpdk090" {
   background_dielectric_constant 1.000000
   temp_reference 25
   layout_scale 0.9
#!    use_silicon_density true
   em_tref 110
   em_output_wlt drawn
   em_variables  w delta_T
   em_conductor_unit mA
   em_via_area_unit  mA
   em_via_unit mA
   em_peak_with_duty_ratio explicit
}


conductor "mt1" {
em_model {
     em_jmax_dc_avg EQU 100 * 1* ( w - 0.008 ) jmax_factor 100 2.077 105 1.434 110
     1.000 115 0.704 120 0.500 125 0.358 L > 9 W < 0.45
     em_jmax_dc_avg EQU 150 * 1.227 * ( w - 0.008 ) jmax_factor 100 2.077 105 1.434
     110 1.000 115 0.704 120 0.500 125 0.358 L <= 9 L > 4.5 W < 0.45
     em_jmax_dc_avg EQU 200 * 1.227 * ( w - 0.008 ) jmax_factor 100 2.077 105 1.434
     110 1.000 115 0.704 120 0.500 125 0.358 L > 9 W >= 0.45
     em_jmax_dc_avg EQU 200 * 1.227 * ( w - 0.008 ) jmax_factor 100 2.077 105 1.434
     110 1.000 115 0.704 120 0.500 125 0.358 L <= 9 L > 4.5 W >= 0.45
     em_jmax_dc_avg EQU 400 * 1.227 * ( w - 0.008 ) jmax_factor 100 2.077 105 1.434
     110 1.000 115 0.704 120 0.500 125 0.358 L <= 4.5
      em_jmax_dc_peak EQU 25.0 * ( w - 0.008 )
      em_jmax_ac_peak EQU 25.0 * ( w - 0.008 )
```


© 2020 All Rights Reserved.

```
      em_jmax_dc_rms EQU sqrt( 18.58 * delta_T * ( w - 0.008 )^2 * ( w - 0.008 +
     0.246 ) / ( w - 0.008 + 0.0443 ) )

      em_jmax_ac_rms EQU sqrt( 18.58 * delta_T * ( w - 0.008 )^2 * ( w - 0.008 +
     0.246 ) / ( w - 0.008 + 0.0443 ) )

   }

}
```


© 2020 All Rights Reserved.

EM Data File

An EM data input file specifies the technology information, such as current density limits, and provides a mapping between the layers for highlighting.

Note: The EM data file syntax is case-sensitive. The file must start with parenthesis.

Following is a sample EM data file.

```
(
; this file is case sensitive
routingLayers = ("Poly" "Metal1" "Metal2" "Metal3" "Metal4" "Metal5" "Metal6")
viaLayers = ("Cont" "Nimp" "Pimp" "Via1" "Via2" "Via3" "Via4" "Via5")
viaWidthList = (("Nimp" 0.2) ("Pimp" 0.2) ("Cont" 0.2) ("Via1" 0.2)
("Via2" 0.2) ("Via3" 0.2) ("Via4" 0.2) ("Via4" 0.2))
xrefLayers = (
( "POLYcont" ("Cont" "Cont"))
( "NSDcont" ("Nimp" "Nimp"))
( "PSDcont" ("Pimp" "Pimp"))
( "poly" ("Poly" "Poly"))
( "mt1" ("Metal1" "Metal1"))
( "mt2" ("Metal2" "Metal2"))
( "mt3" ("Metal3" "Metal3"))
( "mt4" ("Metal4" "Metal4"))
( "mt5" ("Metal5" "Metal5"))
( "mt6" ("Metal6" "Metal6"))
( "Via2NoCapInd" ("Via2" "Via2"))
( "Via2" ("Via2" "Via2"))
)
avgCurrentDensSpecList = (
(nil layer "Poly" minW 0.0 maxW -1.0 minL 0 maxL 5 currentDensity ((1.0 , 110)))
(nil layer "Metal1" minW 0.0 maxW -1.0 currentDensity ((1.1 , 110)))
(nil layer "Metal2" minW 0.0 maxW -1.0 currentDensity ((1.2 , 110)))
(nil layer "Metal3" minW 0.0 maxW -1.0 currentDensity ((1.3 , 110)))
(nil layer "Metal4" minW 0.0 maxW -1.0 currentDensity ((1.4 , 110)))
(nil layer "Metal5" minW 0.0 maxW -1.0 currentDensity ((1.5 , 110)))
(nil layer "Metal6" minW 0.0 maxW -1.0 currentDensity ((1.6 , 110)))
(nil layer "Cont" minW 0.0 maxW -1.0 res 1.0 currentDensity ((2.0 , 110)))
(nil layer "Nimp" minW 0.0 maxW -1.0 res 1.1 currentDensity ((2.1 , 110)))
(nil layer "Pimp" minW 0.0 maxW -1.0 res 1.2 currentDensity ((2.2 , 110)))
(nil layer "Via1" minW 0.0 maxW -1.0 res 1.3 currentDensity ((2.3 , 110)))
(nil layer "Via2" minW 0.0 maxW -1.0 res 1.4 currentDensity ((2.4 , 110)))
```


© 2020 All Rights Reserved.

```
(nil layer "Via3" minW 0.0 maxW -1.0 res 1.5 currentDensity ((2.5 , 110)))
(nil layer "Via4" minW 0.0 maxW -1.0 res 1.6 currentDensity ((2.6 , 110)))
(nil layer "Via5" minW 0.0 maxW -1.0 res 1.7 currentDensity ((2.7 , 110)))
)
peakCurrentDensSpecList = (
(nil layer "Poly" minW 0.0 maxW -1.0 currentDensity ((1.1 , 110)))
(nil layer "Metal1" minW 0.0 maxW -1.0 currentDensity ((1.2 , 110)))
(nil layer "Metal2" minW 0.0 maxW -1.0 currentDensity ((1.3 , 110)))
(nil layer "Metal3" minW 0.0 maxW -1.0 currentDensity ((1.4 , 110)))
(nil layer "Metal4" minW 0.0 maxW -1.0 currentDensity ((1.5 , 110)))
(nil layer "Metal5" minW 0.0 maxW -1.0 currentDensity ((1.6 , 110)))
(nil layer "Metal6" minW 0.0 maxW -1.0 currentDensity ((1.7 , 110)))
(nil layer "Cont" minW 0.0 maxW -1.0 res 1.0 currentDensity ((2.1 , 110)))
(nil layer "Nimp" minW 0.0 maxW -1.0 res 1.1 currentDensity ((2.2 , 110)))
(nil layer "Pimp" minW 0.0 maxW -1.0 res 1.2 currentDensity ((2.3 , 110)))
(nil layer "Via1" minW 0.0 maxW -1.0 res 1.3 currentDensity ((2.4 , 110)))
(nil layer "Via2" minW 0.0 maxW -1.0 res 1.4 currentDensity ((2.5 , 110)))
(nil layer "Via3" minW 0.0 maxW -1.0 res 1.5 currentDensity ((2.6 , 110)))
(nil layer "Via4" minW 0.0 maxW -1.0 res 1.6 currentDensity ((2.7 , 110)))
(nil layer "Via5" minW 0.0 maxW -1.0 res 1.7 currentDensity ((2.8 , 110)))
)
```


© 2020 All Rights Reserved.

Layer Map File

© 2020 All Rights Reserved.

Layer Map File

The layer map file or map file provides the mapping between the layer names in the simulation database to those specified in the technology files (`qrcTechFile` and `ICT` file) for the EM analysis flow. It maps the layer names in the xDSPF file to the ICT layer names. It is needed under either of the following circumstances:

- When a 3rd party xDSPF file with different layer names is used, or
- When Quantus package uses the `layerSetupFile`, which includes the `"model"` keywords to define parasitic resistor model names. For example,

  ```
  pro_layer=M1 ext_layer=MET1 model=metal1
  ```

  In this case, the `"metal1"` to `"M1"` mapping needs to be specified in the layer map file.

  The sample command for the above in the `.cdsinit` file will be specified as follows:

  ```
  envSetVal("spectre.envOpts" "emLayerMap" 'string "/your-path/layermapfile")
  ```

Sample filename: `contactmapfile.txt`

Format:

```
RCX keyword   simulation DB layername  ICT keyword  technology file layername
```

Format Description:

`RCX` specifies the keyword for the simulation database.

`simulation DB layername` specifies the layer names in the simulation database that are to be mapped to the technology file layer names.

`ICT` specifies the keyword for the technology file.

`technology file layername` specifies the layer names in the technology file that map to those in the simulation database.

Example:

```
<RCX keyword> <simulation DB layername> <ICT keyword>  <technology file layername>
RCX   metal1   ICT   mt1
RCX   metal2   ICT   mt2
RCX   metal3   ICT   mt3
RCX   metal4   ICT   mt4
RCX   pl1co    ICT   poly
RCX   VIA1     ICT   via1
```


© 2020 All Rights Reserved.

DFII Layer Map File

The APS/XPS-to-DFII layer map file provides the mapping between the layer names in the extractor-generated xDSPF file to the DFII layer names.

Multiple extraction layers can be mapped to the same DFII layer. This layer map file is also used for performing structural analysis.

Sample filename: `df2layermap.txt`

Format:

```
type     extraction_layer_name     dfII_layer_name
```

Format Description:

`type` specifies the type of layer; for example, metal, via, or poly.

`extraction_layer_name` specifies the name of the layer in the extractor-generated xDSPF file that is to be mapped to the DFII layer.

`dfII_layer_name` specifies the name of the DFII layer to which the extractor-generated layer name is being mapped.

Example:

```
#<type> <extraction_layer_name> <dfII_layer_name>
via    via3             Via3
metal  mt3              Metal3
via    via2_out_capind  Via2
metal  mt2              Metal2
via    via1             Via1
metal  mt1              Metal1
via    cont_ndiff       Cont
via    cont_pdiff       Cont
```

Note: The order of layers in the layer map file is important. The layer map file must have layers listed in the top-to-bottom order.

Support for User-Specified Purpose

By default, Voltus-Fi reads only the “drawing” purpose for the layers specified in the DFII layer map file. However, you can specify additional purpose names in the file for highlighting their shapes. For example,

© 2020 All Rights Reserved.

```
metal m1 M1   e1 e2
```

Where `e1` and `e2` are additional purpose names whose shapes are to be highlighted.

© 2020 All Rights Reserved.

Self-Heating Effect Analysis File

This section lists the self-heating effect analysis parameter file, which is required for plotting the self- heating effect analysis plots.

shParamFile

This file is the self-heating effect analysis parameter file, which is required for plotting the self-heating effect analysis plots.

Sample filename: `param.sh`

Example:

```
self_heat_parameters {

    K_SH_Scale   =  1.23

    beta_c1      =  0.0012
    beta_c2      =  0.0023
    beta_c3      =  0.0034

    device_layers {
        layer "OD" {
           Rth           = 1000
            finger_effect = 2.0 * (1-exp(-0.3 * finger_num))
            fin_effect    = 1.0 - ( 0.018 * (10 - fin_num) )
        }
        layer "OD_2" {
            Rth           = 500
            finger_effect = 2.0 * (1-exp(-0.4 * finger_num))
            fin_effect    = 1.0 - ( 0.019 * (30 - fin_num) )
        }
    }
    conductor_layers {
        layer "M1" {
            alpha_connecting  = 0.50
            alpha_overlapping = 0.40
        }
        layer "M2" {
            alpha_connecting  = 0.45
```


© 2020 All Rights Reserved.

```
            alpha_overlapping = 0.35
        }
    }
```


© 2020 All Rights Reserved.

Power-Grid View Creation Files

This section lists the power-grid view creation file formats.

- Format and Contents of Trigger File
- LEF layer Map File
- PGDB layer Map File

Format and Contents of Trigger File

The file format of the trigger file is:

```
CELL <cellname1>
MODE_NAME <mode1>
CONDITIONAL_INPUT <conditional input statement for cellname1>
CONDITIONAL_PIN <pin_name> (rise| fall | both)
SIM_START_TIME <time>
SIM_STOP_TIME <time>
END_MODE

END
```

where,

"|" denotes boolean OR operation

The table below lists the contents of the trigger file and provides their descriptions.

Trigger File Content	Description
#Memory cell name	
CELL *<cellname1>*	Name of the memory cell
#Mode and conditional details for trigger	
MODE_NAME *<mode1>* CONDITIONAL_INPUT *<conditional input statement for cellname1>*	Mode name and conditional input statement for the cell.

© 2020 All Rights Reserved.

Trigger File Content	Description
CONDITIONAL_PIN *<pin_name>* (rise | fall | both)	Conditional pin statement for the cell. Signal pin for which the tool should rise or fall.
#<Other mode independent parameters>	
SIM_START_TIME *<time>* SIM_STOP_TIME *<time>*	SIM_START_TIME: Specifies the simulation start time. The default unit is in nanoseconds (ns). SIM_STOP_TIME: Specifies the simulation stop time. The default unit is in nanoseconds (ns).
#End cell section	
END_MODE	End cell section

Example

```
CELL adc_sample
# MODE_NAME READ

MODE READ
CONDITIONAL_INPUT = ( WEN & !CEN )
CONDITIONAL_PIN  CLK { RISE }
SIM_START_TIME 35ns
SIM_END_TIME   50ns
END_MODE

MODE WRITE
CONDITIONAL_INPUT = ( !WEN & !CEN )
CONDITIONAL_PIN  CLK { RISE }
SIM_START_TIME 15ns
SIM_END_TIME   30ns
END_MODE

MODE STBY
CONDITIONAL_INPUT = ( CEN )
CONDITIONAL_PIN  CLK { RISE }
SIM_START_TIME 80ns
SIM_END_TIME   120ns
```


© 2020 All Rights Reserved.

```
END_MODE

END
```

where,

`"!"` denotes low state of the signal

`"&"` denotes boolean AND operation

© 2020 All Rights Reserved.

LEF layer Map File

A layer mapping file is required for the Rail Analysis library generation flow. The LEF layer map file contains the mapping information to map the layer names in the LEF file to the layer names in the technology file.

In the GUI, you can specify this file in the *Create Power Grid Views* form.

Sample filename: `lefdef.layermap`

Command: `create_pgv -lef_layer_map` *`lef_layer_map_file_name`*

Format:

```
type     technology_layer_name     lefdef lef_layer_name
```

Format Description:

`type` specifies the type of library layer, for example, metal or via.

`technology_layer_name` specifies the name of the library layer to map to the LEF layer.

`lefdef` specifies the keyword for the LEF layer mapping.

`lef_layer_name` specifies the name of the LEF layer to map to the library layer.

Example:

```
type    technology_layer_name      lefdef    M1

metal    METAL_1                   lefdef    M1
via      VIA_1                     lefdef    VIA1
metal    METAL_2                   lefdef    M2
via      VIA_2                     lefdef    VIA2
```

where “type” can be poly/via/metal/diff.

© 2020 All Rights Reserved.

PGDB layer Map File

A layer mapping file is required for the Rail Analysis library generation flow. The PGDB layer map file contains the mapping information to map the xDSPF layer names that are written in the simulation database to the layer names in the technology file.

In the GUI, you can specify this file in the *Create Power Grid Views* form.

Sample filename: `pgdb.map`

Command: `create_pgv pgdb_layer_map` *`pgdb_layermap_file_name`*

Format:

```
type    technology_layer_name    pgdb    xDSPF_layer_name
```

Format Description:

`type` specifies the type of library layer. It can be poly/via/metal/diff.

`technology_layer_name` specifies the name of the library layer to map to the xDSPF layer name.

`pgdb` specifies the keyword for the xDSPF layer mapping.

`xDSPF_layer_name` specifies the xDSPF layer name.

Example:

```
  type    technology_layer_name  pgdb  pgdb_layer_name
# ----   --------------------- ---- ---------------
  poly   POLYCIDE              pgdb    poly
  via    CONT                  pgdb    pl1co
  via    CONT                  pgdb    odCont1
  via    CONT                  pgdb    odCont2
  metal  METAL_1               pgdb    metal1
  via    VIA_1                 pgdb    VIA1
  metal  METAL_2               pgdb    metal2
  via    VIA_2                 pgdb    VIA2
  diff   OD                    pgdb    nwires
  diff   OD                    pgdb    mwires
```

Note:

- It is recommended to map the S/D diffusion layer to the original layer name instead of `mwires` in the PGDB layer map file.
- When generating PGVs from Voltus-Fi, some of the layers are not required for analysis, but must be preserved to maintain connectivity. Voltus-Fi supports mapping of these

© 2020 All Rights Reserved.

layers to a special layer "`connectivity`" in the layer map file. An example is shown below.

Example:

```
type    technology_layer_name  pgdb    xDSPF_layer_name
diff    connectivity           pgdb    nwires
diff    connectivity           pgdb    mwires
```


© 2020 All Rights Reserved.

Common Command Language (CCL) File

The common command language (CCL) file or the Quantus command file is used to specify the various commands and options that define the extraction run. This command file defines the various input files and libraries that make up the design, the output format, and the desired parasitics to extract from the design. The format and example of a CCL file generated in Voltus-Fi XL is provided below.

Format:

Lines in the CCL file are in the following general Tcl format:

`command {argument_list}`

where *`argument_list`* is a list of name-value pairs.

For more information about the Quantus command file or CCL format, see the “Quantus Command Files” chapter in the *Quantus Extraction Users Manual*.

Example:

An example of the Voltus-Fi-generated CCL file is below. The customized extract command file, `myextract.txt`, is also shown below:

The `myextract.txt` is as follows:

```
extract \
     -selection "all" \
     -type "rc_coupled"
```

The Voltus-Fi-generated CCL file used to extract xDSPF is as follows:

```
extract \
     -selection "all" \
     -type "rc_decoupled"

extraction_setup \
     -analysis "em" \
     -array_vias_spacing "0.35" \
     -max_fracture_length infinite \
     -max_fracture_length_unit "MICRONS" \
     -max_fracture_via_count 1 \
     -max_via_array_count_by_layer "Via1 1" "Via2 1" "Via3 1"  "Via4 1" "Via5 4"
     "Via6 4" "Via7 4" "Via8 4" "Via9 4" "Via10 4" "Via11 4" \
     -net_name_space "SCHEMATIC"

hierarchical_extract \
     -split_feedthrough_pins "true"
```


© 2020 All Rights Reserved.

```
filter_res \
     -min_res 0.001 \
     -remove_dangling_res false

filter_cap \
     -exclude_floating_nets "true" \
     -exclude_floating_nets_limit 50000

output_db \
     -type dspf \
     -subtype extended \
     -add_explicit_vias true \
     -busbit_delimiter "\[\]" \
     -disable_instances false \
     -hierarchy_delimiter "/" \
     -include_cap_model "false" \
     -include_res_model "false" \
     -include_parasitic_cap_model "false" \
     -include_parasitic_res_model "comment" \
     -include_parasitic_res_length "true" \
     -include_parasitic_res_width false \
     -include_parasitic_res_width_drawn true \
     -output_xy  \
     "CANONICAL_RES" \
     "PARASITIC_RES" \
     "GENERIC" \
     -netlist_coupling_values "single" \
     -add_bulk_terminal false \
     -sub_node_char "#" \
     -device_finger_delimiter "@" \
     -include_parasitic_res_temp_coeff "true" \
     -reduce_i_cards "true"

output_setup \
     -directory_name "vfi_quantus101" \
     -file_name "vfi_quantus101/testrun.dspf" \
     -net_name_space "SCHEMATIC" \
     -temporary_directory_name "vfi_temp" \
     -compressed false

process_technology \
     -technology_library_file "tech.lib" \
     -technology_name "tech" \
     -temperature 25.000000

input_db \
     -type pegasus \
     -directory_name "testdir" \
     -hierarchy_delimiter "/" \
     -run_name "testrun"
```


© 2020 All Rights Reserved.

21

Voltus-Fi Form Descriptions

- Create Power Grid Views Form
- Hierarchical Top Simulation Form
- Extract xDSPF Form

© 2020 All Rights Reserved.

Create Power Grid Views Form

Field	Description
LEF Files and Layermap Files	
OA Libs	Displays the name of the OA database library.
Macro Name	Displays the name of the top cell.
Library Name	Displays the name of the library.
Extraction Tech File	Displays the name of the technology file that will be used for reading the technology data.
Tech Lef File	Displays the name of the file that includes the LEF information for the technology data.
Macro Lef File	Displays the name of the file that includes the LEF information for the design.
Lef Def Layer Map File	Displays the name of the file that provides the mapping between LEF layer names and technology layer names.
PGDB Layer Map File	Displays the name of the file that provides the mapping between xDSPF layer names (which are written in the simulation database) and the technology layer names.
Simulation Directory	Provides the path to the directory that contains the current (`*.ptiavg`) files.
Simulation Prefix	Indicates the prefix used for creating an instance while performing the simulation.
UTI Ron File	Specifies the Spectre-generated file that contains multiple on-resistance (Ron) values, that is minimum, maximum, and average, for powergates.
Trigger File	Specifies the file used during dynamic current simulation to run a detailed analysis for determining current distribution.
Net Name	Lets you specify a net.
Voltage	Lets you specify a voltage value for the net.

© 2020 All Rights Reserved.

State Directory / Results File	Provides the path to the directory that contains the simulation result file.
Add	Adds the net, the corresponding voltage, and the EMIR Results DB file location to the list.
Remove	Removes a net from the list.
Output	
Results Directory	Specifies the output directory in which the PGVs will be saved.
Ok	Generates the PGVs.
Cancel	Closes the form without executing the command.
Default	Displays all the default values.
Save	Saves the configuration file.
Load	Loads the configuration file.
Help	Opens help information in Cadence Help.

© 2020 All Rights Reserved.

Hierarchical Top Simulation Form

Field	Description
Block	
xDSPF Type	Displays the type of xDSPF to be used from the PGV directory. The following options are available: ■ *Detailed*: Specifies the detailed block xDSPF file. ■ *Port*: Specifies the port block xDSPF file. ■ *Via Snapped*: Specifies the via layer xDSPF generated using via layer snapping.
PGV dir for block xDSPF file	Provides the path to the PGV directory that contains the *xDSPF file*.
Instance(s)	Indicates the instance name of the block.
Current Type	Lets you specify whether the current analysis will be *Static* or *Dynamic*.
Current File	Displays the name of the piece-wise linear (PWL) file that is generated while running block simulation.
Add Block	Adds the specified block data to the table.
Add Block instance	Adds another instance to the already added block with a different current file.
Remove	Removes any block data from the table.
Top	
xDSPF File	Displays the name of the top-level xDSPF file.
Ckt File	Specifies the input circuit file for performing top-level simulation. This file contains information about models/vectors and other parameters required for top-level simulation.

© 2020 All Rights Reserved.

Bus Bit Char	Specifies the bus delimiter character. The following options are available: *<>*, *[]*, *{}*.
EMIR Config file	
EM Techfile	Specifies the EM technology file to be used in the EMIR configuration file.
Instance Hierarchy	Specifies the instance hierarchy name to be used in the EMIR configuration file.
Create New	Generates a default EMIR configuration file, `emir_vfi.conf`.
Output	
Results Directory	Specifies the directory that stores the results of the top-level simulation.
Run	Runs the top-level simulation.
Cancel	Closes the form without executing the command.
Save	Saves the configuration file.
Load	Loads the configuration file.
Help	Opens help information in Cadence Help.

© 2020 All Rights Reserved.

Extract xDSPF Form

Field	Description
Input	
Data Type	Lets you specify whether the input data will come from *PVS/Pegasus* or *Calibre*. By default, *PVS/Pegasus* is selected.
Data Directory	Provides the path to the directory in which the input data is stored.
Run Name	Specifies the name of the PVS/Pegasus or Calibre run.
Post Processing Script	Specifies the name of the post-processing script file.
Hierarchical EMIR flow	Enables hierarchical EMIR analysis. **Note:** This field is specific to the hierarchical EMIR flow.
PGV Library Name	Specifies the block PGV library that is used for the block-level extraction performed for the hierarchical flow. **Note:** This field is specific to the hierarchical EMIR flow.

© 2020 All Rights Reserved.

Process	
Foundry / Process Node	Lets you specify the process node. The following options are available: ■ *N5/N3:* for N5 and N3 process nodes ■ *N7:* for N7 process nodes ■ *N10-PG*: for N10 process nodes with PG rules ■ *N10*: for N10 process nodes ■ *N20/N16/N12*: for N20, N16, and N12 process nodes ■ *N14*: for N14 process nodes ■ *Others > 20*: for process nodes greater than N20

© 2020 All Rights Reserved.

Signal EM	Indicates that signal nets should be included in the xDSPF file.
	
	When *Signal EM* is selected, the following command is added to the Quantus command file or the common command language (CCL) file, to perform decoupled RC extraction `(rc_decoupled)` for all nets.
	
	Note: In `rc_decoupled` extraction, an RC network for the specified nets with both parasitic resistance and capacitance decoupled to ground is extracted.
	
	`extract \`
	`  -selection "all" \`
	`  -type "rc_decoupled"`
	
	When *Signal EM* is not selected, you need to provide the names of the power or ground nets on which `rc_decoupled` extraction will be performed. For all other nets, `c_only_decoupled` extraction is performed.
	
	Note: In `c_only_decoupled` extraction, only parasitic capacitance between the specified nets and adjacent nets decoupled to the specified cap ground is extracted.
	
	For example, when the following command is specified, `c_only_decoupled` extraction is performed on all nets:
	
	`extract \`
	`  -selection "all" \`
	`  -type "c_only_decoupled"`
	
	When the following command is specified, `rc_decoupled` extraction is performed on nets specified in the `vfiQrc_nets` file. For all other nets, `c_only_decoupled` extraction is performed.
	
	`extract \`
	`  -selection "nets_file ./vfiQrc_nets"`
	`  -type "rc_decoupled"`

© 2020 All Rights Reserved.

PG Net(s)	Provides information about the power and ground nets that should be included in the xDSPF file. This option is specified only when the *Signal EM* option is deselected. If the *Signal EM* option is not selected and *PG Nets* are not provided, a pop-up window opens asking for the list of PG Nets.
Nets File	Displays the names of the nets to be included in extraction. When the *Nets File* option is checked, the *Signal EM* and *PG Net(s)* options are ignored.
Extraction Type	Specifies the extraction type. ■ *RC*: Specifies that both parasitic resistors and capacitors will be extracted. ■ *R only*: Specifies that only parasitic resistors will be extracted. When *Extraction Type* selected is *R only*, the simulation time is short. While selecting the extraction type, consider the type of simulation being run. For example, to run a DC operating point (DCOP) simulation, select *R only* as the *Extraction Type* because the capacitance values are not required.

© 2020 All Rights Reserved.

Cap Coupling Model	Specifies the coupling mode. ■ *De-coupled*: Specifies that decoupled capacitance extraction will be performed. ■ *Coupled*: Specifies that coupled capacitance extraction will be performed. By default, *De-coupled* mode is selected. While specifying the coupling mode, ensure that there is no `extract` command defined in the additional CCL file. The `extract` command in the additional CCL file will override the above options.
Parasitic RES Color	Enables color-aware EM analysis.
Compression	Generates xDSPF in a compressed format.
Reduce I-Cards	Enables the `reduce_i_cards` option in the CCL file that is used for generating the xDSPF file. If a net is “capacitance only”, it is qualified for reduced I card or instance card treatment. This means that when this option is selected, the instance cards and the corresponding dummy resistors are not printed in the results. **Note:** It is recommended not to select this option while performing extraction for the ESD analysis flow.
Comment Sub Ckt	Comments out the top sub circuit statements from the xDSPF file.
Sub Conductor Model	Provides information about the parasitic resistor models for the subconductor layers that should be included during extraction. This is used to support different EM rules for poly layers during EM analysis.

© 2020 All Rights Reserved.

Filter RES Center	Includes sub-nodes after eliminating resistors with small resistance values. When this option is selected, the survivor sub-nodes, among a group of sub-nodes that are formed by the filtered resistors, are chosen based on their proximity to the geometric center of the group. This field is specific to process nodes 5nm and below.
Filter RES By Layer	Lets you specify the LVS layer name for which the minimum resistance value for the extracted parasitic resistors is to be applied. If the layer name is not specified, the default value, `RH_TN_6 0`, is used. This field is specific to process nodes 5nm and below.
Ref Node	Specifies the reference node on which the extraction is to be performed. By default, `0` is specified.
Sub-Node Char	Indicates the subnode identification character. By default, `#` is specified.
Fracture Via Count	Indicates the number that will be used to divide pseudo vias into segments during extraction.
Via Array Spacing	Indicates the distance that will be used as the maximum distance criteria for grouping vias within the same array. This means that if the distance between vias is less than the specified maximum distance, they are grouped within the same array. The default value of the via array spacing is 0.35 times the via size for N10, N16, N12, and N20 processes. For other processes, the default is *auto*, in which case Voltus-Fi sets the via spacing automatically.
auto	Indicates that Voltus-Fi sets the via spacing automatically.

© 2020 All Rights Reserved.

Via Array Count	Indicates the number of vias per side allowed in an array (without the per via layer count option). The default value is `4`. For example, if the via array count is set to 2, the number of vias allowed per array side will be 2.
Technology	
Technology Library File	Displays the location of the technology file to be used for reading the technology data.
Technology Directory	Displays the name of the directory that contains the technology file.
Technology Name	Displays the name of the technology file to be used for extraction. This field is optional if the technology directory has been specified.
Corner	Displays the name of the technology corner to be picked from the library.

© 2020 All Rights Reserved.

Temperature	Indicates the temperature at which the extraction will be performed.
	
	By default, a temperature of 25 degree Celsius is used. The temperature specified for extraction is optional. The options, temperature and `temp_coeff`, specified in the CCL file are mutually exclusive. The following scenarios hold true.
	
	■ If you specify a temperature for extraction in the Extract xDSPF form or by using the `-temperature` parameter of the `extract_xdspf` command in batch mode, the following option in the CCL file will be set to `false`:
	
	`-include_parasitic_res_temp_coeff "false" \`
	
	The software displays the following message:
	
	`User specified temperature will be used for extraction. The option include_parasitic_res_temp_coeff will be set as false.`
	
	■ If you do not specify a temperature for extraction, the following option in the CCL file will be set to `true`:
	
	`-include_parasitic_res_temp_coeff "true" \`
	
	The software displays the following message:
	
	`No user specified temperature value so the option include_parasitic_res_temp_coeff will be set as true`

No of CPUs	Displays the number of CPUs to be used during the Quantus extraction run.

© 2020 All Rights Reserved.

SubCkt Pin Order File	Provides the order of pins to be used during extraction.
	
	The pin order is specified in the xDSPF file using the `.SUBCKT` and `.ENDS` statements. You can use this option to provide a file with a different pin order that will override the order specified using the default `.SUBCKT` statement in the xDSPF file.
	
	Below are examples of pin orders in the xDSPF file before and after specifying the pin order file.
	
	Without the sub-circuit pin order file
	
	`.SUBCKT top In_001 VDD Out_001 VSS In_002 Out_002 In_003 Out_004 ...`
	`.ENDS`
	
	With the sub-circuit pin order file
	
	`.SUBCKT top VDD VSS In_001 Out_001 In_002 Out_002 In_003 Out_004 ...`
	`.ENDS`

Customized Extract Commands File	Lets you specify the file with user-customized extract commands.
Add Extract Command	Indicates that Voltus-Fi adds the extract commands to the CCL file. By default, when a `customextract` file is specified, all commands, including the extract commands, from that file are added to the Voltus-Fi-generated CCL file. However, when you select *Add Extract Command*, Voltus-Fi adds the extract commands to the CCL file, and all other commands specified in the `customextract` file are added to the CCL file.

© 2020 All Rights Reserved.

Parasitic blocking device cells file	Lets you specify the file containing parasitic blocking device cell names, if any. Once a cell is declared a blocking cell, all the parasitic RCs inside the cell are not extracted. If the cell content overlaps top cell routing, then, the top cell routing's RCs in the overlapping regions are also not extracted.
Output	
Directory	Specifies the output directory in which the xDSPF file will be saved. If this file is not specified, a pop-up window opens asking for the name of the directory.
xDSPF File Name	Specifies the name of the xDSPF file that will be generated. If you do not specify the filename, Voltus-Fi generates a file with the name, `<runName>.dspf`, by default.
Autorun Spfchecker	Runs the SPF Checker utility after the extraction run. SPF Checker checks the syntax of the DSPF/SPEF followed by opening the output log. It also compares the terminal names in the DSPF and the device models and populates the summary section with the recommended mapping statements or the `spf aliasterm` statements.
Generate CCL	Creates the CCL or the Quantus command file that will be used to run Quantus.
Generate and Run	Creates the CCL file and runs Quantus to create the xDSPF file in a single step.
Run	Selects the CCL file to use for running Quantus.
Reset	Displays all the default values.
Cancel	Closes the form without executing the command.
Save	Saves the xDSPF file.
Load	Loads the configuration file.
Help	Opens help information in Cadence Help.

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

A

Third-Party DSPF

Troubleshooting Information Related to Third-Party DSPF File

Problem – Missing Number of Vias

The DSPF files are created using the parasitic extraction tools, and the content and format of these files are heavily dependent on the extraction tool and the extraction runset.

In the EM analysis flow, after you calculate the current density of a presistor, you can compare it with the current density limits from the `qrcTechFile`, the ICT file, or the `emDataFile` that you have provided in order to locate violations. The current density limit is width-based for the metal or diffusion layers and area-based for via layers. The value of `$L` or `$W` is used to select the current density limit for metal layers, and the value of `$A` is used to select the current density limit for via layers. In case of a via array, the total area of the via specified in the DSPF file is divided by the value of `$N`, which is the number of vias, in order to obtain the area of a single via, and the current density related to that area is selected from the technology file to check the EM violations.

You can view two examples below, one of a Quantus-extracted DSPF file and the other of a third-party-extracted DSPF file.

Example A-1 Quantus-Extracted DSPF

```
Rk173 AVSS#200 AVSS#201 0.020247 $metal1_conn  $L=0.17 $W=0.54 $X=13.301 $Y=0.43
Rw174 AVSS#44 AVSS#45 0.250000 $Via1  $A=0.0098 $N=2 $X=18.545 $Y=16.18
```

Example A-2 Third-Party-Extracted DSPF

```
Rg1 XR1#PLUS A#2 1.209331 $metal1  $L=0.56205 $W=0.1071 $X=5.582 $Y=0.13
Ro31 net06 net06#7 12.500000 $VIA1  $A=0.003969 $X=23.085 $Y=0.135
```


© 2020 All Rights Reserved.

Information about the number of vias ($N) is missing in the third-party-extracted DSPF file example. The EMIR analysis performed using such files leads to erroneous results because the correct EM rules are not picked from the ICT file or the emDataFile.

As in the case of the example shown above, the resistance value, area, and coordinates of the via resistor are given, but the number of vias or $N is not mentioned. Without this information, Voltus-Fi fails to calculate the area of a single via and check the current density limit related to that via area.

Solution

To resolve the problem of missing number of vias or $N information in a third-party DSPF file, use one of the following options in batch mode:

- Set the following variable in the vfibatch command file before running the load_em_results command:

  ```
  set_variable viaCountFromResistance true
  ```

  Alternatively, include the following option in the EMIR control file before running simulation:

  ```
  emirutil viaCountFromResistance=[true]
  ```

  Note: Specifying the viaCountResistance variable using the set_variable command will generate EM results (taking care of $N) during the batch mode run. Specifying the viaCountResistance variable using the emirutil command will generate EM results (taking care of $N) in the report file *rpt_em, during Spectre simulation itself. The file, *rpt_em is created along with the simulation result file, *.emir#_bin file.

- Include the “area_resistance” of vias in the ICT file. The format is as follows:

  ```
  area_resistance <resistance value 1> <area 1> <resistance value 2> <area 2> …. <resistance value N> <area N>
  ```

- Include the "contact_resistance" of vias in the ICT file. The resistance of a single via is equal to contact_resistance.

  Note: If no area_resistance value is matching, Voltus-Fi will use contact_resistance.

Voltus-Fi-XL calculates the number of vias (without $N) from the following:

- Resistance value of the via array resistor (from the DSPF)
- Resistance value of a single via, which is equal to the contact_resistance (from the ICT file)

© 2020 All Rights Reserved.

OR

Resistance value of all vias, which is equal to `area_resistance` (from the ICT file)

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

B

Converting the EM Data File to ICT File

Voltus-Fi-XL supports the conversion of an electromigration (EM) data file (`emDataFile`) into an interconnect technology (ICT) file in batch mode. The converted ICT file can then be used, both in batch and GUI modes, to perform EM analysis in Voltus-Fi-XL.

To convert an EM data file into an ICT file, perform the following step.

- Provide information about the `convert_emdatafile_to_ict` command in the command file, which is specified using the `vfibatch` command to run voltus-Fi-XL.

  ```
  vfibatch -cmd cmd-file-name
  ```

  The `convert_emdatafile_to_ict` command converts the specified EM data file into an ICT file.

convert_emdatafile_to_ict

```
convert_emdatafile_to_ict
     -emDataFile emDataFile_path \
     [-ictfile converted_ict_file] \
     [-cmdfile additional_cmd_file] \
     -extracted_lib_name lib_name \
     -extracted_cell_name cell_name \
     -extracted_view_name view_name
```

Converts the specified electromigration (EM) data file (`emDataFile`) into an interconnect technology (ICT) file to be used for performing EM analysis in Voltus-Fi-XL.

© 2020 All Rights Reserved.

Parameters

`-cmdfile` *`additional_cmd_file`*

Specifies the name of the command file that is generated from the specified `emDataFile`. If this parameter is not specified, a file with the name, `additional.cmd` is created in the current working directory.

This is an optional parameter.

`-emDataFile` *`emDataFile_path`*

Specifies the location of the `emDataFile` that is to be converted into the ICT file.

This is a required parameter.

`-extracted_cell_name` *`cell_name`*

Specifies the cell name for the design.

This is a required parameter.

`-extracted_lib_name` *`lib_name`*

Specifies the library name for the design.

This is a required parameter.

`-extracted_view_name` *`view_name`*

Specifies the view name for the design.

This is a required parameter.

`-ictfile` *`converted_ict_file`*

Specifies the name of the converted ICT file. If this parameter is not specified, a file with the name, `toolgenerated.ict` is generated in the current working directory.

This is an optional parameter.

Example

The following command generates `myict.ict` file from the specified `emdataFile.txt` file:

```
convert_emdatafile_to_ict \
-emDataFile emDataFile.txt \
```


© 2020 All Rights Reserved.

```
-ictfile myict.ict \
-cmdfile mycmd.cmd \
-extracted_lib_name amsPLL \
-extracted_cell_name vco \
-extracted_view_name layout
```

Supported EM Rules for Converting EM Data File into ICT File

The following electromigration (EM) rules are supported in the conversion of an electromigration (EM) data file (`emDataFile`) into an interconnect technology (ICT) file:

- `currentDensityMPV`: Specifies the Current Density of via in milliamps per via instead of the standard milliamps per micron.
- `deltaT`: Specifies the maximum rise in temperature in degree Celsius, caused due to Joule heating. It must be a positive value. If this is not specified, the software uses a default value of 5 degrees Celsius. This is an ideal condition where the design must be EM free.

  `deltaT` can be used in defining the Current Density limit: I_{rms}

  For example,

  ```
  (nil layer "M1" minW 0.09 maxW 12.0 currentDensity
  (((sqrt(18.33*deltaT*(w-.016)*(w-.016)*(w-.016+.352)/(w-.016+.0443) )/w),
  110)))
  ```

- `recoveryFactor`: Calculates the effective DC electromigration in bipolar current wave forms. This accounts for some 'sweep back recovery' of EM during the negative portion of the current wave form. The recovery factor is specified in the EM Rules specification file as:

  `recoveryFactor=<value>`

  where the user-specified value ranges from 0 to 1.0. The default value is 1.

  The following equation shows how the recovery factor is used by the software while calculating the total average EM current:

  ```
  I (avg) = max( I+, I- ) - (recovery factor)* min( I+, I-)
  ```

  where `I+` is the average of `I` in the positive direction and `I-` is the average of `I` in the negative direction

© 2020 All Rights Reserved.

- `recoveryFactorList`: Specifies the list of layers and the corresponding recovery factors to be used for these layers during EM analysis. The following example shows how different recovery factors are specified for layers, `Via1` and `mt1`.

  ```
  recoveryFactorList=(("Via1" 0.7)("mt1" 0.3))
  ```

  Note: If both `recoveryFactor` and `recoveryFactorList` are specified, the list of layers and their corresponding values specified using the `recoveryFactorList` keyword get preference.

- `cadGrid`: Specifies the minimum width of parasitic resistor. The unit of measurement is micrometer (`um`) or micron.

- `routingLayers`: Specifies the names of the DFII layers that are used for routing. Include only the poly and metals. For example:

  ```
  routingLayers=("poly1" "m1" "m2" "m3" "m4" "m5" "m6" "m7" "m8" "m9" "MD")
  ```

- `viaLayers`: Specifies the DFII layer names of via layers.

  ```
  ( Metal1      Cont            Poly)//the via layer is Cont
  ```

  An example of the viaLayers syntax follows:

  ```
  viaLayers=("cw" "v1" "v2" "v3" "v4" "v5" "v6" "v7" "v8")
  ```

- `xrefLayers`: These are the cross-reference layers that provide mapping between the Quantus layer names, Current Density names, and the DFII layer names. If a layer is not present in `xreflayers` mapping, its DFII name is used as the Current Density name and Quantus name.

  ```
  (("QuantusName" ("currentDensityName" "DFIIName"))
  ```

- `CurrentDensSpecList`: Refers to the Current Density specification declared by the foundry. The syntax allows one layer to support two different Current Density specifications according to the width of the material. The following Current Density specifications are supported: `avgCurrentDensSpecList, avgAbsCurrentDensSpecList, rmsCurrentDensSpecList, DCrmsCurrentDensSpecList, peakCurrentDensSpecList, and ACpeakCurrentDensSpecList`

  The following optional parameters are supported:

  ```
  minW, maxW, minL, maxL, minLu, maxLu, minWu, maxWu, minWb, maxWb, minLb, maxLb
  ```

  Note: The conversion procedure of `emDataFile` to ICT file categorizes the Current Density specification rule in the following two ways:

  Standard Form: `(EQU/w)*JmaxFactor`

  Non-Standard Form: Any form except the above. The software uses interpolation for specifying the EM Rules in the ICT file.

© 2020 All Rights Reserved.

- `LayerStack`: The layer stack contains a list of via names and the corresponding upper metal and bottom metal layer names.

  ```
  (("LayerName" ("upperMetalLayerName" "bottomMetalLayerName"))
  ```

  - *`LayerName`* is the layer name specified in the layout view
  - *`upperMetalLayerName`* is the upper metal layer name in the layout view
  - *`bottomMetalLayerName`* is the bottom metal layer name in the layout view

- `viaWidthList`: Specifies the minimum via width. For example:

  ```
  viaWidthList=(("Cont" 0.2)("Via1" 0.2))
  ```

  Where,

  `Cont` is the DFII name of via layer.

  `0.2` is the minimum width of the default via for this layer.

- `extViaWidthList`: Specifies the minimum via width. For example:

  ```
  extViaWidthList=(("extCont" 0.2)("extVia1" 0.2))
  ```

  Where

  `extCont` is the Quantus (Extraction) name of via layer.

  `0.2` is the minimum width of the default via for this layer.

  Note: `viaWidthList` will be ignored if `extViaWidthList` is specified.

- `DesignResistors`: Specifies the list of design resistor cell names. For each cell name, you can specify the resistor property name, width property name, length property name, and units for width and length property names for which the design resistors are to be analyzed.

  The default unit for both width and length property name is meter. Specifying the resistor property name, width property name, length property name, and unit name is optional. If any of these property names is not specified, the software uses “r”, “w”, and “l” as the default values for these properties.

  The width and length property names are specified when the names of these properties are different from the “w” and “l” in the layout view.

  Syntax:

  ```
  DesignResistors = ("CellName1" "CellName2" "CellName3")
  DesignResistors = ("CellName.*")
  ```

- `junctionTemp`: Specifies the temperature to be used for EM analysis. Used as `tj` in ICT.

© 2020 All Rights Reserved.

- `applyRThreshold`: Specifies the threshold ratio "r" for AC-Peak analysis.
- `rmsMultiplierPin`: Specifies the RMS relaxation factor for EM power-grid analysis. The default value is `1.0`.
- `rmsMultiplierSignal`: Specifies the RMS relaxation factor for signal EM analysis. The default value is `1.0`.

Unsupported EM Rules for Converting EM Data File into ICT File

The following electromigration (EM) rules are not supported in the conversion of an electromigration (EM) data file (`emDataFile`) into an interconnect technology (ICT) file:

- `Jmaxlife`: Specifies the scaling factor that applies to the Current Density limits for different lifetimes.
- `BulkTerminal`: Specifies the bulk terminals of the design resistors that are to be ignored while performing EM analysis.
- `minI`: Used to limit the reporting of small currents that are of little or no interest.
- `durationTime`: Specifies the maximum time period for which violations are allowed.
- `technologyNode`: Specifies the technology-dependent calculation.
- `viaLengthList`: Specifies the via length.
- `extViaLengthList`: Specifies the via length. It get preference over `viaLengthList`.
- `selectResistors`: Selects resistors by their names. All other resistors are ignored for the analysis.
- `selectLayers`: Selects resistors based on the layer name. All resistors on other layers will be ignored for the analysis.
- `ignoreLayers`: Ignores resistors based on the extracted layer name. All resistors on other layers will be selected for the analysis.
- `ignoreResistors`: Ignores resistors by their name. All other resistors will be selected for the analysis.
- `meshR`: Specifies regions for mesh resistance extraction.
- `fracture_vias`: Specifies whether or not there are dummy vias present.
- `Thickness`: Specifies the thickness of the layers.

© 2020 All Rights Reserved.

- `preExecuteFiles`: Includes the list of pre-loaded custom files when reading the `emDataFile`.

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

Index

A

analysis
  static 141
Average (avg) 275

B

batch commands
  pin_2_pin_res 568
  set_variable 658
Batchmode
  batch commands IR 564, 567, 568, 570, 572, 573
  batch commands PGV 519, 521, 600, 609
  batch commands query 612, 614, 618, 619
  batch commands RON 611
  batch commands SHE 577
  batch commands xDSPF 548

C

Cadence Help, launching 42
create_pgv 600
current density 275

D

data preparation 51
data requirements
  EM rule file requirements 92
    EM data file 122
  simulation requirements 56
  xDSPF requirements 55

E

Electromigration analysis 474
EM analysis
  types of EM analysis 275
EM rules in techfile
  jmax parameters 95
  limit-based parameters 93
extract_xdspf 548

F

file formats
  pgv creation files 735
    LEF layermap file 738
    PGDB layermap file 739
    trigger files 735
  rail analysis files 721
    DFII layermap file 731
    EM data file 727
    EM Only ICT file 725
    ICT file 722
    Layer Map file 730
    Layer map file 729
  self-heating effect files 733

G

ground_pins 559

H

help, launching Cadence Help 42

I

IR drop analysis results
  displaying results 184
  flow 172
IR plots
  solid shape highlighting 165

L

load_em_results 572

© 2020 All Rights Reserved.

load_ir_results 564

N

node_id_info 619

O

overview
- IR drop analysis results 158
- static current analysis 142
- types of IR plots 215
- what-if analysis 446

P

pin_2_pin_res 568
power_pin 559
power-grid view 31, 45, 47, 474
Power-grid views
- generating PGVs 481

print_em_report 573
print_ir_report 564, 570
print_rlrp_report 567
print_thermal_report 577

R

related documents 22
- other documentation 22
- Voltus-Fi documentation 22

report_vfi_XL_ron 611
Reports
- default reports 254
- reports using EMIR control file 252

res_id_info 614
run_static_simulation 560

S

search_node_id 618
search_res_id 612
set_block_data 519, 521
set_mmsim_pgv_nets 609
setup_simulation 563
Spectre
- direct EMIR analysis 52
- Iterated EMIR analysis 52

static
- current analysis 141
- EM analysis 141
- rail analysis 141
- static current analysis 142

structural analysis
- checks 426
  - bottom via missing 429
  - missing via connections 429
  - skewed via ratio 432
  - top via missing 427
  - via coverage ratio 431
- examples 436
  - bottom/top via ratio 438
  - missing via connections 439
  - unconnected from bottom 437
  - unconnected from top 436
- flow 434

T

Types of EM plots
- current density - avgabs 296
- current density - peak 300
- current density - rms 298
- current density average 294
- current density max 293
- tap current 220

types of IR plots
- IR drop (ir) 216
- least-resistive path (lrp) 230
- powergate current (pi) 224
- powergate voltage (pv) 227
- REffective (REff) 229
- resistor current (rc) 217
- transistor voltage (iv) 219

V

Variables 660
- connectingODLayers 662
- consolidatedReport 662
- dynamicACPeak 667
- emLimitScale 665
- enableDesignResistorScaling 667
- enablePWLDensityInterpolation 667
- enableReportScaleParam 668

© 2020 All Rights Reserved.

enableRMSHotSpotWaiving 668
enableSHHeatSink 669
prioritizedPwlRulesLayers 674
rmsMetalLineNumberPower 675
rmsMetalLineNumberSignal 676
shParamFile 676
Tlife 677
useRecoveryFactor 679
variables
applyRThreshold 662
dynamicACPeak 664
geounit 670
geounit_xy 670, 671
optimisticEMRuleSelection 672
powerRailGroundNets 673
powerRailRules 673
powerRailSupplyNets 673
process_scale 674
rmsMultiplierPin 676
shEMEffect 676
Tj 677
Tlife 678
UnmatchedRuleResistor 675
useEncryptedWidth 679
viaCountFromResistance 679
VoltusFIXL
data requirements 55
Voltus-Fi-XL
flow overview 52
VoltusFiXL
product features 45

© 2020 All Rights Reserved.