HSPICE® Reference Manual: Commands and Control Options

Version B-2008.09, September 2008

SYNOPSYS®

Copyright Notice and Proprietary Information

Right to Copy Documentation

The license agreement with Synopsys permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any. Licensee must assign sequential numbers to all copies. These copies shall contain the following legend on the cover page:

> "This document is duplicated with the permission of Synopsys, Inc., for the exclusive use of _____ and its employees. This is copy number _____."

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Synopsys, AMPS, Astro, Cadabra, CATS, Design Compiler, DesignWare, Formality, HSPICE, iN-Phase, Leda, MAST, ModelTools, NanoSim, OpenVera, PathMill, Physical Compiler, PrimeTime, SiVL, SNUG, SolvNet, TetraMAX, VCS, Vera, and YIELDirector are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, Columbia, Columbia-CE, Cosmos, CosmosLE, CosmosScope, CRITIC, DC Expert, DC Professional, DC Ultra, Design Analyzer, DesignPower, Design Vision, DesignerHDL, Direct Silicon Access, Discovery, Eclypse, Encore, EPIC, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, HSIM, HSIM^{plus}, in-Sync, iN-Tandem, i-Virtual Stepper, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport, Library Compiler, Magellan, Mars, Mars-Rail, Mars-Xtalk, Milkyway, ModelSource, Module Compiler, Planet, Planet-PL, Polaris, Power Compiler, Raphael, Saturn, Scirocco, Scirocco-i, Star-RCXT, Star-SimXT, System Compiler, Taurus, TSUPREM-4, VCS Express, VCSi, VHDL Compiler, VirSim, and VMC are trademarks of Synopsys, Inc.

Service Marks (^{sm})

MAP-in, SVP Café, and TAP-in are service marks of Synopsys, Inc.

SystemC is a trademark of the Open SystemC Initiative and is used under license.
ARM and AMBA are registered trademarks of ARM Limited.
Saber is a registered trademark of SabreMark Limited Partnership and is used under license.
All other product or company names may be trademarks of their respective owners.

Contents

1. HSPICE and HSPICE RF Application Commands 1

hspice. 1

hspicerf 10

2. HSPICE and HSPICE RF Netlist Commands 13

HSPICE and RF Commands Overview 14

.AC 19

.ACMATCH. 22

.ACPHASENOISE 24

.ALIAS 26

.ALTER. 28

.APPENDMODEL 30

.BIASCHK 31

.CHECK EDGE 37

.CHECK FALL 39

.CHECK GLOBAL_LEVEL. 40

.CHECK HOLD 41

.CHECK IRDROP 43

.CHECK RISE 45

.CHECK SETUP 47

.CHECK SLEW 49

.CONNECT 51

.DATA 54

.DC. 61

.DCMATCH 66

.DCVOLT 68

.DEL LIB. 69

.DESIGN_EXPLORATION 72
.DISTO 74
.DOUT 76
.EBD. 78
.ELSE 80
.ELSEIF 81
.END 82
.ENDDATA 83
.ENDIF 84
.ENDL 85
.ENDS 86
.ENV. 87
.ENVFFT 88
.ENVOSC 90
.EOM 91
.FFT 92
.FLAT 97
.FOUR 98
.FSOPTIONS 99
.GLOBAL 102
.HB 103
.HBAC 107
.HBLIN 108
.HBLSP 110
.HBNOISE 112
.HBOSC 115
.HBXF 120
.HDL. 121
.IBIS 124
.IC 128
.ICM 130

.IF 132
.INCLUDE 134
.LAYERSTACK 135
.LIB 137
.LIN 140
.LOAD 144
.LPRINT 146
.MACRO 147
.MALIAS 149
.MATERIAL 151
.MEASURE (or) .MEAS 153
.MEASURE (Rise, Fall, and Delay Measurements) 155
.MEASURE (FIND and WHEN) 158
.MEASURE (Equation Evaluation/ Arithmetic Expression) 162
.MEASURE (AVG, EM_AVG, INTEG, MIN, MAX, PP, and RMS) 164
.MEASURE (Integral Function) 168
.MEASURE (Derivative Function) 170
.MEASURE (Error Function) 173
.MEASURE PHASENOISE 175
.MEASURE PTDNOISE 179
.MEASURE (Pushout Bisection) 180
.MEASURE(ACMATCH) 182
.MEASURE(DCMATCH) 183
.MEASURE FFT 185
.MODEL 188
.MOSRA 194
.MOSRAPRINT 197
.NODESET 198
.NOISE 200
.OP 203
.OPTION 205

.PARAM 207
.PAT 211
.PHASENOISE. 213
.PKG 216
.POWER 218
.POWERDC 220
.PRINT 221
.PROBE 225
.PROTECT or .PROT 227
.PTDNOISE 228
.PZ 231
.SAMPLE 232
.SAVE 233
.SENS 235
.SHAPE 237
.SHAPE (Defining Rectangles) 238
.SHAPE (Defining Circles) 239
.SHAPE (Defining Polygons) 240
.SHAPE (Defining Strip Polygons) 242
.SN 243
.SNAC 245
.SNFT. 246
.SNNOISE 249
.SNOSC 251
.SNXF 254
.STATEYE 256
.STIM 258
.SUBCKT 263
.SURGE 266
.SWEEPBLOCK 267
.TEMP (or) .TEMPERATURE. 269

.TF 271
.TITLE 272
.TRAN 273
.UNPROTECT or .UNPROT 281
.VARIATION 282
.VEC 285

3. HSPICE and RF Netlist Simulation Control Options 287
HSPICE Control Options Grouped By Function 289
.OPTION ABSH 298
.OPTION ABSI 299
.OPTION ABSMOS 300
.OPTION ABSTOL 301
.OPTION ABSV 302
.OPTION ABSVAR 303
.OPTION ABSVDC 304
.OPTION ACCT 305
.OPTION ACCURATE 306
.OPTION ACOUT 307
.OPTION ALTCC 308
.OPTION ALTCHK 309
.OPTION APPENDALL 310
.OPTION ARTIST 311
.OPTION ASPEC 312
.OPTION AUTOSTOP (or) .OPTION AUTOST 313
.OPTION BA_ACTIVE 314
.OPTION BA_ERROR 315
.OPTION BA_FILE 316
.OPTION BA_TERMINAL 317
.OPTION BADCHR 318
.OPTION BDFATOL 319

.OPTION BDFRTOL 320
.OPTION BEEP 321
.OPTION BIASFILE 322
.OPTION BIASINTERVAL 323
.OPTION BIASNODE 324
.OPTION BIASPARALLEL 325
.OPTION BIAWARN 326
.OPTION BINPRNT 327
.OPTION BPNMATCHTOL 328
.OPTION BRIEF 329
.OPTION BSIM4PDS 330
.OPTION BYPASS 331
.OPTION BYTOL 332
.OPTION CAPTAB 333
.OPTION CHGTOL 334
.OPTION CMIFLAG 335
.OPTION CMIPATH 336
.OPTION CMIUSRFLAG 337
.OPTION CONVERGE 338
.OPTION CPTIME 339
.OPTION CSCAL 340
.OPTION CSDF 341
.OPTION CSHDC 342
.OPTION CSHUNT 343
.OPTION CUSTCMI 344
.OPTION CVTOL 345
.OPTION D_IBIS 346
.OPTION DCAP 347
.OPTION DCCAP 348
.OPTION DCFOR 349
.OPTION DCHOLD 350

.OPTION DCIC 351
.OPTION DCON 352
.OPTION DCSTEP 353
.OPTION DCTRAN 354
.OPTION DEFAD 355
.OPTION DEFAS 356
.OPTION DEFL 357
.OPTION DEFNRD 358
.OPTION DEFNRS 359
.OPTION DEFPD 360
.OPTION DEFPS 361
.OPTION DEFSA 362
.OPTION DEFSB 363
.OPTION DEFSD 364
.OPTION DEFW. 365
.OPTION DELMAX 366
.OPTION DI 367
.OPTION DIAGNOSTIC (or) .OPTION DIAGNO 368
.OPTION DLENCSDF 369
.OPTION DV 370
.OPTION DVDT 371
.OPTION DVTR 372
.OPTION EM_RECOVERY 373
.OPTION EPSMIN 374
.OPTION EXPLI 375
.OPTION EXPMAX 376
.OPTION FAST 377
.OPTION FFT_ACCURATE 378
.OPTION FFTOUT 379
.OPTION FMAX 380
.OPTION FS 381

.OPTION FSCAL 382
.OPTION FT 383
.OPTION GDCPATH 384
.OPTION GENK 385
.OPTION GEOSHRINK 386
.OPTION GMAX 387
.OPTION GMIN 388
.OPTION GMINDC 389
.OPTION GRAMP 390
.OPTION GSCAL 391
.OPTION GSHDC 392
.OPTION GSHUNT 393
.OPTION HBACKRYLOVDIM 394
.OPTION HBACKRYLOVITR 395
.OPTION HBACTOL 396
.OPTION HBCONTINUE 397
.OPTION HBFREQABSTOL 398
.OPTION HBFREQRELTOL 399
.OPTION HB_GIBBS 400
.OPTION HBJREUSE 401
.OPTION HBJREUSETOL 402
.OPTION HBKRYLOVDIM 403
.OPTION HBKRYLOVTOL 404
.OPTION HBKRYLOVMAXITER 405
.OPTION HBLINESEARCHFAC 406
.OPTION HBMAXITER 407
.OPTION HBMAXOSCITER 408
.OPTION HBPROBETOL 409
.OPTION HBSOLVER 410
.OPTION HBTOL 411
.OPTION HBTRANFREQSEARCH 412

.OPTION HBTRANINIT 413
.OPTION HBTRANPTS 414
.OPTION HBTRANSTEP 415
.OPTION HIER_DELIM 416
.OPTION HIER_SCALE 417
.OPTION ICSWEEP 418
.OPTION IMAX 419
.OPTION IMIN 420
.OPTION INGOLD 421
.OPTION INTERP 422
.OPTION IPROP 423
.OPTION ITL1 424
.OPTION ITL2 425
.OPTION ITL3 426
.OPTION ITL4 427
.OPTION ITL5 428
.OPTION ITLPTRAN 429
.OPTION ITLPZ 430
.OPTION ITRPRT 431
.OPTION KCLTEST 432
.OPTION KLIM 433
.OPTION LA_FREQ 434
.OPTION LA_MAXR 435
.OPTION LA_MINC 436
.OPTION LA_TIME 437
.OPTION LA_TOL 438
.OPTION LENNAM 439
.OPTION LIMPTS 440
.OPTION LIMTIM 441
.OPTION LISLVL 442
.OPTION LIST 443

.OPTION LOADHB 444
.OPTION LOADSNINIT 445
.OPTION LSCAL 446
.OPTION LVLTIM 448
.OPTION MACMOD 449
.OPTION MAXAMP 451
.OPTION MAXORD 452
.OPTION MBYPASS 453
.OPTION MCBRIEF 454
.OPTION MEASDGT 455
.OPTION MEASFAIL 456
.OPTION MEASFILE 457
.OPTION MEASOUT 458
.OPTION METHOD 459
.OPTION MODMONTE 462
.OPTION MONTECON 464
.OPTION MOSRALIFE 465
.OPTION MOSRASORT 466
.OPTION MTTHRESH 467
.OPTION MU 468
.OPTION NCFILTER 469
.OPTION NCWARN 470
.OPTION NEWTOL 471
.OPTION NODE 472
.OPTION NOELCK 473
.OPTION NOISEMINFREQ 474
.OPTION NOMOD 475
.OPTION NOPAGE 476
.OPTION NOPIV 477
.OPTION NOTOP 478
.OPTION NOWARN 479

.OPTION NUMDGT 480
.OPTION NUMERICAL_DERIVATIVES 481
.OPTION NXX 482
.OPTION OFF 483
.OPTION OPFILE 484
.OPTION OPTCON 485
.OPTION OPTLST 487
.OPTION OPTS 488
.OPTION PARHIER (or) .OPTION PARHIE 489
.OPTION PATHNUM 490
.OPTION PHASENOISEKRYLOVDIM 491
.OPTION PHASENOISEKRYLOVITER 492
.OPTION PHASENOISETOL 493
.OPTION PHNOISELORENTZ 494
.OPTION PHNOISEAMPM 495
.OPTION PIVOT 496
.OPTION PIVREF 498
.OPTION PIVREL 499
.OPTION PIVTOL 500
.OPTION POST 501
.OPTION POSTLVL 503
.OPTION POST_VERSION 504
.OPTION POSTTOP 506
.OPTION PROBE. 507
.OPTION PSF 508
.OPTION PURETP. 509
.OPTION PUTMEAS 510
.OPTION PZABS 511
.OPTION PZTOL 512
.OPTION RANDGEN 513
.OPTION RELH 514

.OPTION RELI 515
.OPTION RELMOS 516
.OPTION RELQ 517
.OPTION RELTOL 518
.OPTION RELV 519
.OPTION RELVAR 520
.OPTION RELVDC 521
.OPTION RESMIN 522
.OPTION RISETIME (or) .OPTION RISETI 523
.OPTION RITOL 525
.OPTION RMAX 526
.OPTION RMIN 527
.OPTION RUNLVL 528
.OPTION SAVEHB 533
.OPTION SAVESNINIT 534
.OPTION SCALE 535
.OPTION SCALM 536
.OPTION SEARCH 537
.OPTION SEED 538
.OPTION SIM_ACCURACY 539
.OPTION SIM_DELTAI 540
.OPTION SIM_DELTAV 541
.OPTION SIM_DSPF 542
.OPTION SIM_DSPF_ACTIVE 544
.OPTION SIM_DSPF_INSERROR 545
.OPTION SIM_DSPF_LUMPCAPS 546
.OPTION SIM_DSPF_MAX_ITER 547
.OPTION SIM_DSPF_RAIL 548
.OPTION SIM_DSPF_SCALEC 549
.OPTION SIM_DSPF_SCALER 550
.OPTION SIM_DSPF_VTOL 551

.OPTION SIM_LA 553
.OPTION SIM_LA_FREQ 554
.OPTION SIM_LA_MAXR 555
.OPTION SIM_LA_MINC 556
.OPTION SIM_LA_MINMODE 557
.OPTION SIM_LA_TIME 558
.OPTION SIM_LA_TOL 559
.OPTION SIM_ORDER 560
.OPTION SIM_OSC_DETECT_TOL 561
.OPTION SIM_POSTAT 562
.OPTION SIM_POSTDOWN 563
.OPTION SIM_POSTSCOPE 564
.OPTION SIM_POSTSKIP 565
.OPTION SIM_POSTTOP 566
.OPTION SIM_POWER_ANALYSIS 567
.OPTION SIM_POWER_TOP 569
.OPTION SIM_POWERDC_ACCURACY 570
.OPTION SIM_POWERDC_HSPICE 571
.OPTION SIM_POWERPOST 572
.OPTION SIM_POWERSTART 573
.OPTION SIM_POWERSTOP 574
.OPTION SIM_SPEF 575
.OPTION SIM_SPEF_ACTIVE 576
.OPTION SIM_SPEF_INSERROR 577
.OPTION SIM_SPEF_LUMPCAPS 578
.OPTION SIM_SPEF_MAX_ITER 579
.OPTION SIM_SPEF_PARVALUE 580
.OPTION SIM_SPEF_RAIL 581
.OPTION SIM_SPEF_SCALEC 582
.OPTION SIM_SPEF_SCALER 583
.OPTION SIM_SPEF_VTOL 584

.OPTION SIM_TG_THETA 585
.OPTION SIM_TRAP 586
.OPTION SLOPETOL 587
.OPTION SNACCURACY 588
.OPTION SNMAXITER 589
.OPTION SPMODEL 590
.OPTION STATFL. 591
.OPTION SYMB 592
.OPTION TIMERES 593
.OPTION TMIFLAG 594
.OPTION TMIPATH 595
.OPTION TNOM. 596
.OPTION TRANFORHB 597
.OPTION TRTOL 598
.OPTION UNWRAP 599
.OPTION VAMODEL 600
.OPTION VERIFY 601
.OPTION VFLOOR 602
.OPTION VNTOL 603
.OPTION WACC 604
.OPTION WARNLIMIT (or) .OPTION WARNLIM 605
.OPTION WDELAYOPT 606
.OPTION WDF 607
.OPTION WINCLUDEGDIMAG 608
.OPTION WL 609
.OPTION WNFLAG 610
.OPTION XDTEMP 611
.OPTION (X0R,X0I) 612
.OPTION (X1R,X1I) 613
.OPTION (X2R,X21) 614
.VARIATION Block Control Options 615

.DESIGN_EXPLORATION Block Control Options 618

4. Digital Vector File Commands 619

ENABLE 620

IDELAY 621

IO 623

ODELAY 624

OUT or OUTZ 626

PERIOD 627

RADIX 628

SLOPE 630

TDELAY 632

TFALL 634

TRISE 636

TRIZ 638

TSKIP 639

TUNIT 640

VIH 642

VIL 643

VNAME 644

VOH 646

VOL 648

VREF 650

VTH 652

.GRAPH 656

.MODEL Command for .GRAPH 657

.NET 658

.PLOT 660

.WIDTH 661

.OPTION ALT999 or ALT9999 662

.OPTION BKPSIZ 662

.OPTION CDS 662
.OPTION CO 663
.OPTION H9007. 663
.OPTION MEASSORT 664
.OPTION MENTOR 664
.OPTION MODSRH 665
.OPTION PLIM. 666
.OPTION SDA 666
.OPTION TRCON 667
.OPTION ZUKEN 668
GEAR Method 672
ACCURATE 672
FAST 672
GEAR Method, ACCURATE 673
ACCURATE, GEAR Method 674
ACCURATE, FAST. 674
GEAR Method, FAST. 675
GEAR Method, ACCURATE, FAST 675
RUNLVL=N. 676
RUNLVL, ACCURATE, FAST, GEAR method 677
DVDT=1,2,3 677
LVLTIM=0,2,3 677
KCLTEST 678
BRIEF 678
Option Notes 678
Finding the Golden Reference for Options. 679

About This Manual

This manual describes the individual HSPICE commands you can use to simulate and analyze your circuit designs.

Inside This Manual

This manual contains the chapters described below. For descriptions of the other manuals in the HSPICE documentation set, see the next section, The HSPICE Documentation Set.

Chapter	Description
Chapter 1, HSPICE and HSPICE RF Application Commands	Describes the commands you use to start HSPICE or HSPICE RF, including syntax, arguments, and examples.
Chapter 2, HSPICE and HSPICE RF Netlist Commands	Describes the commands you can use in HSPICE and HSPICE RF netlists.
Chapter 3, HSPICE and RF Netlist Simulation Control Options	Describes the HSPICE and HSPICE RF simulation control options you can set using various forms of the .OPTION command.
Chapter 4, Digital Vector File Commands	Contains an alphabetical listing of the HSPICE commands you can use in an digital vector file.
Appendix A, Obsolete Commands and Options	Describes commands and options no longer commonly used in HSPICE.
Appendix B, How Options Affect other Options	Describes the effects of specifying control options on other options in the netlist.

The HSPICE Documentation Set

This manual is a part of the HSPICE documentation set, which includes the following manuals:

Manual	Description
HSPICE User Guide: Simulation and Analysis	Describes how to use HSPICE to simulate and analyze your circuit designs, and includes simulation applications. This is the main HSPICE user guide.
HSPICE User Guide: Signal Integrity	Describes how to use HSPICE to maintain signal integrity in your chip design.
HSPICE User Guide: RF Analysis	Describes how to use special set of analysis and design capabilities added to HSPICE to support RF and high-speed circuit design.
HSPICE Reference Manual: Commands and Control Options	Provides reference information for HSPICE and HSPICE RF commands and options.
HSPICE Reference Manual: Elements and Device Models	Describes standard models you can use when simulating your circuit designs in HSPICE, including passive devices, diodes, JFET and MESFET devices, and BJT devices.
HSPICE Reference Manual: MOSFET Models	Describes available MOSFET models you can use when simulating your circuit designs in HSPICE.
HSPICE Integration to Cadence™ Virtuoso® Analog Design Environment User Guide	Describes use of the HSPICE simulator integration to the Cadence tool.
AMS Discovery Simulation Interface Guide for HSPICE	Describes use of the Simulation Interface with other EDA tools for HSPICE.
AvanWaves User Guide	Describes the AvanWaves tool, which you can use to display waveforms generated during HSPICE circuit design simulation.

Searching Across the HSPICE Documentation Set

You can access the PDF format documentation from your install directory for the current release by entering `-docs` on the terminal command line when the HSPICE tool is open.

Synopsys includes an index with your HSPICE documentation that lets you search the entire HSPICE documentation set for a particular topic or keyword. In a single operation, you can instantly generate a list of hits that are hyper-linked to the occurrences of your search term. For information on how to perform searches across multiple PDF documents, see the HSPICE release notes.

Note:

To use this feature, the HSPICE documentation files, the Index directory, and the index.pdx file must reside in the same directory. (This is the default installation for Synopsys documentation.) Also, Adobe Acrobat must be invoked as a standalone application rather than as a plug-in to your web browser.

You can also invoke HSPICE and RF documentation in a browser-based help system by entering `-help` on your terminal command line when the HSPICE tool is open. This provides access to all the HSPICE manuals with the exception of the *AvanWaves User Guide* which is available in PDF format only.

Known Limitations and Resolved STARs

You can find information about known problems and limitations and resolved Synopsys Technical Action Requests (STARs) in the *HSPICE Release Notes* shipped with this release. For updates, go to SolvNet.

To access the *HSPICE Release Notes*:

1. Go to https://solvnet.synopsys.com/ReleaseNotes. (If prompted, enter your user name and password. If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.)
2. Select Download Center> HSPICE> version number> Release Notes.

Conventions

The following conventions are used in Synopsys HSPICE documentation.

Table 1 Typographical conventions

Convention	Description
`Courier`	Indicates command syntax.
`Italic`	Indicates a user-defined value, such as *`object_name`*.
`Bold`	Indicates user input—text you type verbatim—in syntax and examples.
[ ]	Denotes optional parameters, such as: `write_file [-f filename]`
`...`	Indicates that parameters can be repeated as many times as necessary: *`pin1 pin2 ... pinN`*
|	Indicates a choice among alternatives, such as `low | medium | high`
+	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Edit > Copy	Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.
Control-c	Indicates a keyboard combination, such as holding down the Control key and pressing c.

Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Support Center.

Accessing SolvNet

SolvNet includes an electronic knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. SolvNet also gives you access to a wide range of Synopsys online services, which include downloading software, viewing Documentation on the Web, and entering a call to the Support Center.

To access SolvNet:

1. Go to the SolvNet Web page at http://solvnet.synopsys.com.
2. If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.)

If you need help using SolvNet, click Help on the SolvNet menu bar.

Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a call to your local support center from the Web by going to http://solvnet.synopsys.com/EnterACall (Synopsys user name and password required).
- Send an e-mail message to your local support center.
  - E-mail support_center@synopsys.com from within North America.
  - Find other local support center e-mail addresses at http://www.synopsys.com/support/support_ctr.
- Telephone your local support center.
  - Call (800) 245-8005 from within the continental United States.
  - Call (650) 584-4200 from Canada.
  - Find other local support center telephone numbers at http://www.synopsys.com/support/support_ctr.

1

HSPICE and HSPICE RF Application Commands

Describes the commands you use to start HSPICE or HSPICE RF, including syntax, arguments, and examples.

This chapter provides the syntax and arguments for the `hspice` and `hspicerf` application commands. You enter these commands at the command-line prompt to start HSPICE or HSPICE RF. This chapter also includes examples for starting HSPICE and the syntax for calculating new measurements from previous simulation results.

The following sections show you how to invoke:

- hspice
- hspicerf

hspice

Invokes HSPICE.

Syntax

```
hspice [-i path/input_file] [-o path/output_file]
   [-n number] [-html path/html_file] [-d]
   [-C path/input_file] [-CC path/input_file] [-I] [-K]
   [-L command_file] [-S] [-mp [number]] [-mt number]
   [-meas measure_file] [-top subcktname]
   [-hdl file_name][-hdlpath pathname]
   [-vamodel name] [-vamodel name2...]
   [-help] [-doc] [-h] [-v] [-x]
```

Argument	Description
`-i` *`path/input_file`*	Input netlist file name for which an extension *.*ext* is optional. If you do not specify an input file name extension in the command, HSPICE searches ▪ for a *.sp# file, or ▪ for a *.tr#, *.ac#, or *.sw# file (PSF files are not supported). HSPICE uses the input file name as the root for the output files. HSPICE also checks for an initial conditions file (.ic) that has the input file root name. The following is an example of an input file name: `/usr/sim/work/rb_design.sp` In this file name: ▪ `/usr/sim/work/` is the directory path to the design ▪ `rb_design` is the design root name ▪ `.sp` is the file name suffix
`-o` *`path/output_file`*	Name of the output file. If you do not specify an extension, HSPICE assigns *.lis*. Everything up to the last period is the root file name and everything after the last period is the file name extension. ▪ If you either do not use this option or you use it without specifying a file name, HSPICE uses the output root file name specified in the `-html` option. ▪ If you do not specify an output file name in either this or the `-html` option, HSPICE uses the input root file name as the output file root file name. ▪ If you include the *.lis* extension in the file name that you enter using this option, then HSPICE does not append another *.lis* extension to the root file name of the output file. ▪ If you do not specify an output file name, HSPICE directs output to stdout. For the .meas option, some case results differ from the measure result HSPICE produces.
`-n` *`number`*	Starting number for numbering output data file revisions (`output_file.tr#`, `output_file.ac#`, `output_file.sw#`, where `#` is between 0 and 9999.).

Argument	Description
`-html` *`path/html_file`*	HTML output file. ▪ If a path is unspecified, HSPICE saves the HTML output file in the same directory that you specified in the `-o` option. ▪ If you do not specify an `-o` option, HSPICE saves the HTML output in the working directory. ▪ If you do not specify an output file name in either the `-o` or `-html` option, then HSPICE uses the input root file name as the output file root file name. If you add .option itrprt = 1 to your netlist to print output variables at their internal time points, and you use the `-html` option when invoking HSPICE, then HSPICE prints the values to a separate file (**.printtr0*).
`-d`	(UNIX) Displays the content of *.st0* files on screen while running HSPICE. For example, to show the status during simulation.
`-C` *`path/input_file`*	Client/Server (C/S) mode. ▪ Entering hspice -C checks out an HSPICE license and starts client/server mode. ▪ Entering hspice -C *path/input_file* simulates your netlist. ▪ Entering hspice -C -K releases the HSPICE license and exits. For additional information, see Using HSPICE in Client/Server Mode in the *HSPICE User Guide: Simulation and Analysis*.
`-CC` *`path/input_file`*	Advanced Client/Server mode. ▪ Entering hspice -CC checks out an HSPICE license and starts client/server mode. ▪ Entering hspice -CC path/input_file simulates your netlist. ▪ Entering hspice -CC -K releases the HSPICE license and exits. For additional information, see Launching the Advanced Client/Server (C/S) Mode in the *HSPICE User Guide: Simulation and Analysis*

Argument	Description
`-I`	Interactive mode. ▪ Entering hspice -I invokes interactive mode. ▪ Entering help at the HSPICE prompt lists supported commands. ▪ Entering hspice -I -L *file_name* runs a command file. ▪ Entering quit at the HSPICE exits interactive mode. For additional information, see "Using Interactive Mode" in the *HSPICE User Guide: Simulation and Analysis*.
`-K`	Used with -C option to terminate client/server mode and exit.
`-L file_name`	Used with -I option to run commands contained in a command file.
`-S`	Performs as a server. Accepts data from SPEED2000, simulates the circuit, and returns results to SPEED2000. ▪ On UNIX and Linux, HSPICE waits for successive simulations after invocation. ▪ On Windows you must re-invoke for each successive simulation.
`-mp` [*number*]	Activates multiprocessing while running ALTER cases, transient sweeps, and Monte Carlo analyses on one machine with multiple processors/cores. If you specify the number of CPUs you can limit the number of CPUs to avoid overtaxing performance scalability. If a CPU number is not specified, HSPICE auto-determines the child processes by the number of available CPUs. For full details see Multiprocessing .ALTER Cases, Transient Sweeps, Monte Carlo in the *HSPICE User Guide: Simulation and Analysis*.
`-mt number`	Invokes multithreading and specifies the number of processors for a multithreaded simulation. For additional information, see "Running Multithreading or Multiprocessing HSPICE Simulations" in the *HSPICE User Guide: Simulation and Analysis*. See also .OPTION MTTHRESH in this manual.

Argument	Description
`-meas` *`measure_file`*	Re-invokes the measure file to calculate new measurements from a previous simulation. The format of measure_file is similar to the HSPICE netlist format. The first line is a comment line and the last line is an .END command. The following netlist commands are supported: ▪ .MEASURE ▪ .PARAM ▪ .TEMP ▪ .OPTION ▪ .DATA ▪ .ENDDATA ▪ .FFT ▪ .MEASURE FFT ▪ .END Note: The .DATA command in the measure file must be consistent with the .DATA command in the wavefile. The following types of .OPTION commands are supported: ▪ MEASFAIL ▪ NUMDGT ▪ INGOLD ▪ MEASDGT Warnings are issued if other options or commands are used. Wave files formatted as PSF and CSDF are not supported. Syntax to perform spectrum analysis measurements from previous simulation results: `hspice -i *.tr0 -meas` *`measure_file`*
`-top` *`subcktname`*	Top level subcircuit name. Effectively eliminates the need for defining "`.subckt` *`subcktname`*" and corresponding "`.ends`" statements.

Argument	Description
`-hdl` *`file_name`*	Verilog-A module. The Verilog-A file is assumed to have a *.va extension when only a prefix is provided. One -hdl option can include one Verilog-A file, use multiple -hdl options if multiple Verilog-A files are needed. This example loads the amp.va Verilog-A source file: `hspice amp.sp -hdl amp.va` When a module to be loaded has the same name as a previously-loaded module or the names differ in case only, the latter one is ignored and the simulator issues a warning message. If a Verilog-A module file is not found or the Compiled Model Library file has an incompatible version, the simulation exits and an error message is issued.
`-hdlpath` *`pathname`*	Search path for a Verilog-A file if HSPICE cannot find it in the current working directory. The search order for Verilog-A files is: 1. Current working directory 2. Path defined by command-line argument `-hdlpath` 3. Path defined by environment variable `HSP_HDL_PATH` The path defined by either `-hdlpath` or `HSP_HDL_PATH` can consist a set of directory names. The path separator must follow HSPICE conventions or platform conventions (";" on UNIX). Path entries that do not exist are ignored and no error or warning messages are issued. This example first searches the current working directory and when a *.va file is not found, the relative location ./my_modules directory is searched: hspice amp.sp -hdlpath ./my_modules
`-vamodel` *`name`* `-vamodel` *`name2...`*	Cell names for Verilog-A definitions. *name* is the cell name that uses a Verilog-A definition rather than a subcircuit definition when both exist. Each `-vamodel` option can take no more than one name. Repeat this option if multiple Verilog-A modules are defined. If no name is supplied after `-vamodel`, then the Verilog-A definition will be used whenever it is available.

Argument	Description
`-help`	Searchable browser-based help system for HSPICE/ HSPICFE RF. An html browser must be installed on your machine to access this help system.
`-doc`	PDF documentation set user manuals for HSPICE and RF flow. Requires Adobe Acrobat Reader to be installed on your system. You can do full text searches of the documentation set. See the Release Notes for instructions.
`-h`	Displays a help message and exits.
`-v`	Outputs version information and exits.
`-x`	Enables HSPICE to handle parsing and back-annotating of parasitic elements to annotate circuits. The options below can be used with the `-x` flag. See the HSPICE Netlist Control Options chapter for discussion of these options: ▪ .OPTION BA_FILE ▪ .OPTION BA_ACTIVE ▪ .OPTION BA_ERROR ▪ .OPTION BA_TERMINAL

Examples of Starting HSPICE

The following are more examples of commands to start running HSPICE.

- `hspice demo.sp -n 7 > demo.out`

  This command redirects output to a file instead of stdout. `demo.sp` is the input netlist file. The `.sp` extension is optional. The `-n 7` starts the output data file revision numbers at 7; for example: `demo.tr7`, `demo.ac7`, `demo.sw7`, and so forth. The `>` redirects the program output listing to file demo.out.

- `hspice -i demo.sp`

  `demo` is the root input file name. Without the `-o` argument and without redirection, HSPICE does not generate an output listing file.

- `hspice -i demo.sp -o demo`

`demo` is the output file root name (designated with the `-o` option). Output files are named demo.lis, demo.tr0, demo.st0, and demo.ic0.

- `hspice -i rbdir/demo.sp`

  `demo` is the input root file name. HSPICE writes the demo.lis, demo.tr0, and demo.st0 output files into the directory where you executed the HSPICE command. It also writes the demo.ic0 output file into the same directory as the input source—that is, rbdir.

- `hspice -i a.b.sp`

  `a.b` is the root name. The output files are ./a.b.lis, ./a.b.tr0, ./a.b.st0, and ./a.b.ic0.

- `hspice -i a.b -o d.e`

  `a.b` is the root name for the input file. `d.e` is the root output file name, except for the .ic file to which HSPICE assigns the `a.b` input file root name. The output files are d.e.lis, d.e.tr0, d.e.st0, and a.b.ic0.

- `hspice -i a.b.sp -o outdir/d.e`

  `a.b` is the root for the .ic0 file. HSPICE writes the .ic0 file into a file named a.b.ic0. `d.e` is the root for other output files. Output files are outdir/d.e.lis, outdir/d.e.tr0, and outdir/d.e.st0.

- `hspice -i indir/a.b.sp -o outdir/d.e.lis`

  `a.b` is the root for the .ic file. HSPICE writes the .ic0 file into a file named indir/a.b.ic0. `d.e` is the root for the output files.

- `hspice test.sp -o test.lis -html test.html`

  This command creates output file in both .lis and .html format after simulating the test.sp input netlist.

- `hspice test.sp -html test.html`

  This command creates only a .html output file after simulating the test.sp input netlist.

- `hspice test.sp -o test.lis`

  This command creates only a .lis output file after simulating the test.sp input netlist.

- `hspice -i test.sp -o -html outdir/a.html`

  This command creates output files in both .lis and .html format. Both files are in the outdir directory and their root file name is a.

- `hspice -i test.sp -o out1/a.lis -html out2/b.html`

  This command creates output files in both .lis and .html format. The .lis file is in the out1 directory and its root file name is a. The .html file is in the out2 directory and its root file name is b.

- `hspice -i test.sp -o test -x`

  This command launches a full parasitic back-annotation for the file named `test.sp`.

Using HSPICE for Calculating New Measurements

When you want to calculate new measurements from previous simulation results produced by HSPICE you can use the following mode to rerun HSPICE without having to do another simulation:

`hspice -meas` *`measurefile`* `-i` *`wavefile`* `-o` *`outputfile`*

See the following table for arguments and descriptions.

Argument	Description
-meas *measurefile*	This format is similar to the HSPICE netlist format. The first line is a comment line and the last line is an .END command. Seven commands are supported: ▪ .MEASURE ▪ .PARAM ▪ .TEMP ▪ .OPTION ▪ .DATA ▪ .ENDDATA ▪ .END Note: The .DATA command in the measure file must be consistent with the .DATA command in the wavefile. The .OPTION command support four types: ▪ MEASFAIL ▪ NUMDGT ▪ INGOLD ▪ MEASDGT Warnings are issued if other options or commands are used. Wave files formatted as PSF and CSDF are not supported.

Argument	Description
-i *wavefile*	**.tr#*, **.ac#*, and **.sw#* files produced by HSPICE. Waveform files formatted as PSF are not supported. If a plot fails to open, it is due to one of the following reasons: ▪ Waveform file format is not supported. ▪ File format is not understood ▪ File is not found ▪ File larger than max size of *x*
-o *outputfile*	Same output files as HSPICE. Some case results are different from the measure result HSPICE produces due to an accuracy problem.
-h	Displays a help message and exits.
-v	Outputs version information and exits.

hspicerf

Invokes HSPICE RF.

Syntax

`hspicerf [-a] input_file [output_file] [-n] [-h] [-v]`

Argument	Description

`-a`	Generates output to stdout in ASCII format. For example,
	`% hspicerf -a ckt.in`
	You can redirect the ASCII output to another file. For example,
	`% hspicerf -a ckt.in > xt`
	Output from a .PRINT command goes to an ASCII file with a .print# or .printac# file extension.

`input_file`	Name of the input netlist.
`output_file`	Name of the output listing file. If specified, the simulation output is written to this file and given a *.lis* file extension. For example, `% hspicerf ckt.in xt` automatically sets -a and generates output to xt.lis.

Argument	Description
`-n` *number*	Starting number for numbering output data file revisions (`output_file.tr#`, `output_file.ac#`, `output_file.sw#`, where `#` is between 0 and 9999.).
`-h`	Returns a help message.
`-v`	Returns version information.

2

HSPICE and HSPICE RF Netlist Commands

Describes the commands you can use in HSPICE/HSPICE RF netlists.

This chapter provides a list of the HSPICE and HSPICE RF netlist commands, arranged by task, followed by detailed descriptions of the individual commands.

The netlist commands described in this chapter fall into the following categories:

- Alter Block
- Analysis
- Conditional Block
- Digital Vector
- Encryption
- Field Solver
- Files
- Input/Output Buffer Information Specification (IBIS)
- Library Management
- Model and Variation Definition
- Node Naming
- Output Porting
- Setup
- Simulation Runs
- Subcircuits
- Verilog-A

HSPICE and RF Commands Overview

Analysis

Use these commands in your netlist to start different types of HSPICE analysis to save the simulation results into a file and to load the results of a previous simulation into a new simulation.

HSPICE

.AC	.DCMATCH	.FOUR	.OP	.SAMPLE	.TEMP (or) .TEMPERATURE
.ACMATCH	.DISTO	.LIN	.PAT	.SENS	.TF
.DC	.FFT	.NOISE	.PZ	.STATEYE	.TRAN

HSPICE RF Analysis

Use these commands in your RF netlist to run different types of HSPICE RF analyses, save the simulation results into a file, and to load the results of a previous simulation into a new simulation.

.AC	.ENVFFT	.LIN	.SN
.ACPHASENOISE	.ENVOSC	.LPRINT	.SNAC
.CHECK EDGE	.FFT	.MEASURE PHASENOISE	.SNFT
.CHECK FALL	.FOUR	.MEASURE PTDNOISE	.SNNOISE
.CHECK GLOBAL_LEVEL	.HB	.NOISE	.SNOSC
.CHECK HOLD	.HBAC	.OP	.SNXF
.CHECK IRDROP	.HBLIN	.PHASENOISE	.SURGE
.CHECK RISE	.HBLSP	.POWER	.SWEEPBLOCK

.CHECK SETUP	.HBNOISE	.POWERDC	.TEMP (or) .TEMPERATURE
.CHECK SLEW	.HBOSC	.PTDNOISE	.TF
.DC	.HBXF	..PZ	.TRAN
.ENV			.

Alter Block

Use these commands in your netlist to run alternative simulations of your netlist by using different data.

.ALTER	.DEL LIB	.TEMP (or) .TEMPERATURE

Conditional Block

Use these commands in your HSPICE netlist to setup a conditional block. HSPICE does not execute the commands in the conditional block unless the specified conditions are true.

.ELSE	.ELSEIF	.ENDIF	.IF

Digital Vector

Use these commands in your digital vector (VEC) file.

ENABLE	SLOPE	VIH
IDELAY	TDELAY	VIL
IO	TFALL	VNAME
ODELAY	TRISE	VOH
OUT or OUTZ	TRIZ	VOL
PERIOD	TSKIP	VREF

RADIX TUNIT VTH

Encryption

Use these commands in your netlist to mark the start and end of a traditionally (Freelib) encrypted section of a netlist.

.PROTECT or .PROT .UNPROTECT or .UNPROT

Field Solver

Use these commands in your netlist to define a field solver.

.FSOPTIONS .LAYERSTACK .MATERIAL .SHAPE

Files

Use this command in your netlist to call other files that are not part of the netlist.

.VEC

Input/Output Buffer Information Specification (IBIS)

Use these commands in your netlist for specifying input/output buffer information.

.EBD .IBIS .ICM .PKG

Library Management

Use these commands in your netlist to manage libraries of circuit designs and to call other files when simulating your netlist.

.DEL LIB .ENDL .INCLUDE .LIB .LOAD

Model and Variation Definition

Use these commands in your netlist to define models:

.ALIAS .APPENDMODEL .MALIAS .MODEL .MOSRA .MOSRAPRINT

.VARIATION

Node Naming

Use these commands in your netlist to name nodes in circuit designs.

.CONNECT .GLOBAL

Output Porting

Use these commands in your netlist to specify the output of a simulation to a printer or graph. You can also define the parameters to measure and to report in the simulation output.

.BIASCHK .MEASURE (or) .MEAS .PROBE .DOUT .PRINT .STIM

Setup

Use these commands in your netlist to set up your netlist for simulation.

.DATA .ENDDATA .IC .NODESET .PARAM .TITLE

.DCVOLT .GLOBAL .LOAD .OPTION .SAVE

Simulation Runs

Use these commands in your netlist to mark the start and end of individual simulation runs and conditions that apply throughout an individual simulation run.

.END	.TEMP (or) .TEMPERATURE	.TITLE

Subcircuits

Use these commands in your netlist to define subcircuits and to add instances of subcircuits to your netlist.

.ENDS	.INCLUDE	.MODEL
.EOM	.MACRO	.SUBCKT

Verilog-A

Use the following command in your netlist to declare the Verilog-A source name and path within the netlist.

.HDL

.AC

Performs several types of AC analyses.

Syntax

Single or Double Sweep

```
.AC type np fstart fstop
.AC type np fstart fstop [SWEEP var [START=]start
+ [STOP=]stop [STEP=]incr]
.AC type np fstart fstop [SWEEP var type np start stop]
.AC type np fstart fstop
+ [SWEEP var START="param_expr1"
+ STOP="param_expr2" STEP="param_expr3"]
.AC type np fstart fstop [SWEEP var start_expr
+ stop_expr step_expr]
```

Sweep Using Parameters

```
.AC type np fstart fstop [SWEEP DATA=datanm]
.AC DATA=datanm
.AC DATA=datanm [SWEEP var [START=]start [STOP=]stop
+ [STEP=]incr]
.AC DATA=datanm [SWEEP var type np start stop]
.AC DATA=datanm [SWEEP var START="param_expr"
+ STOP="param_expr2" STEP="param_expr3"]
.AC DATA=datanm [SWEEP var start_expr stop_expr
+ step_expr]
```

Optimization

```
.AC DATA=datanm OPTIMIZE=opt_par_fun
+ RESULTS=measnames MODEL=optmod
```

Monte Carlo

```
.AC type np fstart fstop [SWEEP MONTE=MCcommand]
```

Arguments

Argument	Description
DATA=datanm	Data name, referenced in the .AC command.

Argument	Description
incr	Increment value of the voltage, current, element, or model parameter. If you use type variation, specify the np (number of points) instead of incr.
fstart	Starting frequency. If you use POI (list of points) type variation, use a list of frequency values, not fstart fstop.
fstop	Final frequency.
MONTE= MCcommand	Where MCcommand can be any of the following: ■ *val* Specifies the number of random samples to produce. ■ *val firstnum=num* Specifies the sample number on which the simulation starts. ■ *list num* Specifies the sample number to execute. ■ list(*num1:num2 num3 num4:num5*) Samples from *num1* to *num2*, sample *num3*, and samples from *num4* to *num5* are executed (parentheses are optional).
np	Number of points or points per decade or octave, depending on which keyword precedes it.
start	Starting voltage or current or any parameter value for an element or model.
stop	Final voltage or current or any parameter value for an element or a model.
SWEEP	Second sweep.
TEMP	Temperature sweep
type	Any of the following keywords: ■ DEC – decade variation. ■ OCT – octave variation. ■ LIN – linear variation. ■ POI – list of points.
var	Name of an independent voltage or current source, element or model parameter or the TEMP (temperature sweep) keyword. HSPICE supports source value sweep, referring to the source name (SPICE style). If you select a parameter sweep, a .DATA command and a temperature sweep, then you must choose a parameter name for the source value. You must also later refer to it in the .AC command. The parameter name cannot start with V or I.

Argument	Description
firstrun	The *val* value specifies the number of Monte Carlo iterations to perform. The *firstrun* value specifies the desired number of iterations. HSPICE runs from num1 to num1+val-1.
list	Iterations at which HSPICE performs a Monte Carlo analysis. You can write more than one number after a list. The colon represents “from ... to ...". Specifying only one number causes to HSPICE run at only the specified point.

Description

The .AC command is usable in several different formats, depending on the application as shown in the examples. You can also use the .AC command to perform data-driven analysis in HSPICE.

If the input file includes an .AC command, HSPICE runs AC analysis for the circuit over a selected frequency range for each parameter in the second sweep.

For AC analysis, the data file must include at least one independent AC source element command (for example, VI INPUT GND AC 1V). HSPICE checks for this condition and reports a fatal error if you did not specify such AC sources.

Example

```
.AC DEC 10 1K 100MEG
```

This example performs a frequency sweep by 10 points per decade from 1kHz to 100MHz.

See Also

.DC
.TRAN
Using the .AC Statement

.ACMATCH

Calculates the effects of variations in device characteristics on a circuit's AC response.

Syntax

```
.ACMATCH OUTVAR [THRESHOLD=T] [FILE=string] [INTERVAL=Int]
```

Arguments

Argument	Description
OutVar	OutputVariable can be one or several output voltages, difference voltages or branch current through an independent voltage source. The voltage or current specifier is followed by an identifier of the AC quantity of interest: M: magnitude P: phase R: real part I: imaginary part
Threshold	Only devices with variation contributions above Threshold are reported in the table. Results for all devices are displayed if Threshold=0 is set. The maximum value for Threshold is 1.0, but at least 10 devices (or all) are displayed. Default is 0.01.
File	Valid file name for the output tables. Default is *basename.am#*, where # is the regular HSPICE sequence number.
Interval	This option applies to the frequency sweep definition in he `.AC` command. A table is printed at the first sweep point, then for each subsequent increment of SweepValue, and at the final sweep point.

Description

Use to calculate the effects of variations in device characteristics on a circuit's AC response. If more than one ACMatch analysis is specified per simulation, only the last command is executed. dB syntax is supported in `.ACMatch` for Vdb and Idb, for local, global, and element variation.

Note:

ACMatch does not support Spatial Variations.

Example

```
.ACMATCH VM(out) VP(out)
.AC dec 10 1k 10Meg interval=10
```

See Also

.AC
.MEASURE (or) .MEAS
.MEASURE(ACMATCH)
.OPTION POST

.ACPHASENOISE

Helps you interpret signal and noise quantities as phase variables for accumulated jitter for closed-loop PLL analysis.

Syntax

```
.ACPHASENOISE output input [interval] carrier=freq
+ [listfreq=(frequencies|none|all)]
+ [listcount=val] [listfloor=val]
+ [listsources=(1|0)]
```

Argument	Description
output	Node voltage or branch current output variable
input	Independent source used as input reference
interval	Number of intervals used to dump jitter and noise summary information
carrier	Device or circuit, for example, a fundamental multiplied phase-locked coaxial resonator oscillator
freq	Frequency (in Hz) of the fundamental carrier upon which the noise transformations are based
listfreq	Dumps the element phase noise value to the .lis file. You can specify which frequencies the element phase noise value dumps. The frequencies must match the sweep_frequency values defined in the parameter_sweep, otherwise they are ignored. In the element phase noise output, the elements that contribute the largest phase noise are dumped first. The frequency values can be specified with the NONE or ALL keyword, which either dumps no frequencies or every frequency defined in the parameter_sweep. Frequency values must be enclosed in parentheses. For example: listfreq=(none) listfreq=(all) listfreq=(1.0G) listfreq=(1.0G, 2.0G) The default value is the first frequency value.

Argument	Description
listcount	Dumps the element phase noise value to the .lis file, which is sorted from the largest to smallest value. You do not need to dump every noise element. instead, you can define listcount to dump the number of element phase-noise frequencies. For example, listcount=5 means that only the top 5 noise contributors are dumped. The default value is 20.
listsources	Dumps the element phase-noise value to the .lis file. When the element has multiple noise sources, such as a level 54 MOSFET, which contains the thermal, shot, and 1/f noise sources. When dumping the element phase-noise value you can decide if you need to dump the contribution from each noise source. You can specify either ON (1) or OFF (0): ON dumps the contribution from each noise source and OFF does not. The default value is OFF.

Description

The `.ACPHASENOISE` command aids in the ability to compute “Accumulated Jitter” or “Timing Jitter” for the closed loop PLL. The accumulated jitter response is essentially an integral transformation of the closed-loop PLL response. The `.ACPHASENOISE` analysis outputs raw data to **.pn0* and **.printpn0* files. The `PHNOISE` data is given in units of dBc/Hz, i.e., dB relative to the carrier, per Hz, across the output nodes specified by the `.ACPHASENOISE` command. The data plot is a function of offset frequency. If the “`JITTER`” keyword is present, `.ACPHASENOISE` also outputs the accumulated TIE jitter data to **.jt0* and **.printjt0* data files. These data are plotted as a function of time in units of seconds. The Timing Jitter data itself has units of seconds. The timing jitter calculations make use of the parameters given in the `.ACPHASENOISE` syntax, such as “`freq`” and “`interval`”.

For details, see Small-Signal Phase-Domain Noise Analysis (.ACPHASENOISE) in the *HSPICE User Guide: RF Analysis*.

.ALIAS

Renames a model or library containing a model; deletes an entire library of models.

Syntax

`.ALIAS` *`model_name1 model_name2`*

Description

Use in instances when you have used `.ALTER` commands to rename a model, to rename a library containing a model, or to delete an entire library of models in HSPICE. If your netlist references the old model name, then after you use one of these types of `.ALTER` commands, HSPICE no longer finds this model.

For example, if you use `.DEL LIB` in the `.ALTER` block to delete a library, the `.ALTER` command deletes all models in this library. If your netlist references one or more models in the deleted library, then HSPICE no longer finds the models.

To resolve this issue, HSPICE provides an `.ALIAS` command to let you use another model name in place of the old model name HSPICE can find in the existing model libraries.

Example 1

You delete a library named `poweramp` that contains a model named `pa1`. Another library contains an equivalent model named `par1`. You can then convert the `pa1` model name to the `par1` model name:

```
.ALIAS pa1 par1
```

During simulation when HSPICE encounters a model named `pa1` in your netlist, it initially cannot find this model because you used an `.ALTER` command to delete the library that contained the model. However, the `.ALIAS` command indicates to use the `par1` model in place of the old `pa1` model and HSPICE *does* find this new model in another library so simulation continues.

You must specify an old model name and a new model name to use in its place. You cannot use `.ALIAS` without any model names:

```
.ALIAS
```

or with only *one* model name:

```
.ALIAS pa1
```

You also cannot alias a model name to *more than one* model name because the simulator cannot determine which of these new models to use in place of the deleted or renamed model:

```
.ALIAS pa1 par1 par2
```

For the same reason you cannot substitute a model name to a second model name and then substitute the second model name to a third model name:

```
.ALIAS pa1 par1
.ALIAS par1 par2
```

If your netlist does not contain an `.ALTER` command and if the `.ALIAS` does not report a usage error, then the `.ALIAS` does not affect the simulation results.

Example 2

Your netlist might contain the command:

```
.ALIAS myfet nfet
```

Without an `.ALTER` command, HSPICE does not use `nfet` to replace `myfet` during simulation.

If your netlist contains one or more `.ALTER` commands, the first simulation uses the original `myfet` model. After the first simulation if the netlist references `myfet` from a deleted library, `.ALIAS` substitutes `nfet` in place of the missing model.

- If HSPICE finds model definitions for both `myfet` and `nfet`, it reports an error and aborts.
- If HSPICE finds a model definition for `myfet`, but not for `nfet`, it reports a warning and simulation continues by using the original `myfet` model.
- If HSPICE finds a model definition for `nfet`, but not for `myfet`, it reports a “replacement successful” message.

See Also

.ALTER
.MALIAS

.ALTER

Reruns an HSPICE/HSPICE RF simulation using different parameters and data.

Syntax

`.ALTER` *`title_string`*

Arguments

Argument	Description
title_string	Any string up to 72 characters. HSPICE prints the appropriate title string for each .ALTER run in each section heading of the output listing and in the graphical data (.tr#) files.

Description

Use this command to rerun an HSPICE simulation using different parameters and data. Use parameter (variable) values for `.PRINT` commands before you alter them. The `.ALTER` block cannot include `.PRINT`, or any other input/output commands. You can include analysis commands (`.DC`, `.AC`, `.TRAN`, `.FOUR`, `.DISTO`, `.PZ`, and so on) in a `.ALTER` block in an input netlist file.

However, if you change only the analysis type and you do not change the circuit itself, then the simulation runs faster if you specify all analysis types in one block, instead of using separate `.ALTER` blocks for each analysis type.

To activate multiprocessing while running .ALTER cases, enter **hspice -mp** on the command line. While running in parallel mode, HSPICE checks if the input case has `.ALTER` commands. If it has, HSPICE splits the input case into several subcases, then fork HSPICE processes to run each subcase at the same time. After all HSPICE processes finish running the subcases, HSPICE merges all the output files of the subcases.

The `.ALTER` sequence or block can contain the following commands:

- Element commands (except E, F, G, H, I, and V source elements)
- .AC commands
- .ALIAS commands
- .DATA commands
- .DC commands

- .DEL LIB commands
- .HDL commands
- .IC (initial condition) commands
- .INCLUDE commands
- .LIB commands
- .MODEL commands
- .NODESET commands
- .OP commands
- .OPTION commands
- .PARAM commands
- .TEMP (or) .TEMPERATURE commands
- .TF commands
- .TRAN commands
- .VARIATION commands

Note:

The `.MALIAS` command is not officially supported in `.ALTER` blocks.

Example

```
.ALTER simulation_run2
```

See Also

.OPTION ALTCC
.OPTION MEASFILE
.OPTION OPTCON
Multiprocessing .ALTER Cases, Transient Sweeps, Monte Carlo

.APPENDMODEL

Appends the .MOSRA (model reliability) parameters to a model card.

Syntax

.APPENDMODEL *SrcModel ModelKeyword1 DestModel ModelKeyword2*

Argument	Description
SrcModel	Source model name, e.g., the name of the MOSRA model.
ModelKeyword	Model type for SrcModel. For example, the keyword mosra.
DestModel	Destination model name, e.g, the original model in the model library.
ModelKeyword2	Model type for DestModel. For example, 'nmos'.

Description

Appends the parameter values from the source model card (SrcModel) to the destination model card (DestModel). All arguments are required. There is wildcard support for the .APPENDMODEL command. The auxiliary command .OPTION APPENDALL refines how .APPENDMODEL works.

Example

The following example appends the content of the model card hci_1 to the b3_nch BSIM3 model card.

```
.appendmodel hci_1 mosra b3_nch nmos
```

Wildcard Examples

In the example below, model p1_ra is appended to all of the pmos models. Quotation marks are required if the model name is defined only by a wildcard.

```
.appendmodel p1_ra mosra  "*"   pmos
```

In the following example, the model p1_ra is appended to all of the pmos models that are named pch* (pch1, pch2, pch_tt, etc.).

```
.appendmodel p1_ra mosra   pch*   pmos
```

See Also

.MODEL
.MOSRA
.OPTION APPENDALL

.BIASCHK

Monitors the voltage bias, current, device size, expression, and region.

Syntax

As an expression monitor

```
.BIASCHK 'expression' [limit=lim] [noise=ns]
+ [max=max] [min=min]
+ [simulation=op|dc|tr|all] [monitor=v|i|w|l]
+ [tstart=time1] [tstop=time2] [autostop]
+ [interval=time]
```

As an element and model monitor

```
.BIASCHK type
 terminal1=t1 [terminal2=t2]
+ [limit=lim] [noise=ns] [max=max] [min=min]
+ [simulation=op|dc|tr|all] [monitor=v|i]
+ [name=name1,name2,...]
+ [mname=modname_1,modname_2,...]
+ [tstart=time1] [tstop=time2] [autostop]
+ [except=name_1,name_2,...]
+ [interval=time] [sname=subckt_name1,subckt_name2,...]
```

As a region monitor

```
.BIASCHK MOS [region=cutoff|linear|saturation]
+ [simulation=op|dc|tr|all]
+ [name=name1,name2, ...]
+ [mname=modname_1,modname_2,...]
+ [tstart=time1] [tstop=time2] [autostop]
+ [except=name1,name2,...]
+ [interval=time] [sname=subckt_name1,subckt_name2,...]
```

As a length and width monitor

```
.BIASCHK type monitor=w|l
+ [limit=lim] [noise=ns] [max=max] [min=min]
+ [simulation=op|dc|tr|all]
+ [name=devname_1,devname_2,...]
+ [name=devname_n,devname_n+1,...]
+ [mname=modelname_1,modelname_2,...]
+ [tstart=time1] [tstop=time2] [autostop]
+ [interval=time] [sname=subckt_name1,subckt_name2,...]
```

Arguments

Argument	Description
type	Element type to check. MOS (C, BJT, ...) For a monitor, *type* can be DIODE, BIPOLAR, BJT, JFET, MOS, NMOS, PMOS, C, or SUBCKT. When used with REGION, *type* can be MOS only.
terminal 1, 2	Terminals between which HSPICE checks (that is, checks between *terminal1* and *terminal2*): ■ For MOS level 57: nd, ng, ns, ne, np, n6 ■ For MOS level 58: nd, ngf, ns, ngb ■ For MOS level 59: nd, ng, ns, ne, np ■ For other MOS level: nd, ng, ns, nb ■ For capacitor: n1, n2 ■ For diode: np, nn ■ For bipolar: nc, nb, ne, ns ■ For JFET: nd, ng, ns, nb For type=subckt, the terminal names are those pins defined by the subcircuit definition of mname.
limit	Bias check limit that you define. Reports an error if the bias voltage (between appointed terminals of appointed elements and models) is larger than the limit.
noise	Bias check noise that you define. The default is 0.1v. Noise-filter some of the results (the local maximum bias voltage that is larger than the limit). The next local max replaces the local max if all of the following conditions are satisfied: `local_max-local_min` *`noise`*. `next local_max-local_min` *`noise`*. This local max is smaller than the next local max. For a parasitic diode, HSPICE ignores the smaller local max biased voltage and does not output this voltage. To disable this feature, set the noise detection level to 0.
max	Maximum value.

Argument	Description
min	Minimum value.
name	Element name to check. If name and mname are not both set for the element type, the elements of this type are all checked. You can define more than one element name in keyword name with a comma (,) delimiter. If doing bias checking for subcircuits: ▪ When both mname and name are defined while multiple name definitions are allowed if a name is also an instance of mname, then only those names are checked, others will be ignored. ▪ This command is ignored if no name is an instance of mname. ▪ For name definitions which are not of the type defined in mname will be ignored. ▪ If a mname is not defined, the subcircuit type is determined by the first name definition.
mname	Model name. If you are doing bias checking for a subcircuit, it is the subcircuit definition name. HSPICE checks elements of the model for bias. If you define mname, then HSPICE checks all devices of this model. You can define more than one model name in the keyword mname with the comma (,) delimiter. If *mname* and *name* are not both set for the element *type*, the elements of this type are all checked. If doing bias checking for subcircuits: ▪ Once there is one and only one mname defined, the terminal names for this command are those pins defined by the subckt definition of mname. ▪ Multiple mname definitions are not allowed. ▪ Wild carding is not supported for mname. ▪ If only mname is specified in a subckt bias check, then all subcircuits will be checked. See also sname below.
region	Values can be cutoff, linear, or saturation. HSPICE monitors when the MOS device, defined in the .BIASCHK command, enters and leaves the specified region (such as cutoff).
simulation	Simulation type you want to monitor. You can specify op, dc, tr (transient), and all (op, dc, and tr). The tr option is the default simulation type.

Argument	Description
monitor	Type of value you want to monitor. You can specify v (voltage), i (current), w, and l (device size) for the element type. This parameter is not used for an expression-type monitor.
tstart	Bias check start time during transient analysis. The default is 0.
tstop	Bias check end time during transient analysis. The analysis ends on its own by default if you do not set this parameter.
autostop	When set, HSPICE supports an autostop for a biaschk card so that it can report error messages and stop the simulation immediately.
except	Specify the element or instance that you do not want to bias check.
interval	Active when .OPTION BIASINTERVAL is set to a nonzero value. This argument prevents reporting intervals that are less than or equal to the time specified.
sname	Name of the subcircuit definition that the *type* of element of lies in. HSPICE checks all elements in this subcircuit for bias. You can define more than one subcircuit name in the keyword *sname* with a comma (,) delimiter. If you are doing bias checking for a subcircuit, sname = the X-element name.

Description

Use this command to monitor the voltage bias, current, device size, expression, and region during analysis. The output reports:

- Element (instance) name
- Time
- Terminals
- Bias that exceeds the limit
- Number of times the bias exceeds the limit for an element

HSPICE saves the information as both a warning and a bias check summary in the **.lis* file or a file you define in the BIASFILE option. You can use this command only for active elements, capacitors, and subcircuits.

More than one simulation type or all simulation types can be set in a single .BIASCHK command. Also, more than one region can be set in a single .BIASCHK command.

After a simulation that uses the .BIASCHK command runs, HSPICE outputs a results summary including the element name, time, terminals, model name, and the number of times the bias exceeded the limit for a specified element.

The keywords *name*, *mname*, and *sname* act as OR'd filters for element selection. Also, if type is subckt in a .BIASCHK command that tries to check the ports of a subcircuit, the keyword *sname* then behaves identically to the *name* keyword.

Element and model names can contain wildcards, either “?” (stands for one character) or “*” (stands for 0 or more characters).

If a model name that is referenced in an active element command contains a period (.), then .BIASCHK reports an error. This occurs because it is unclear whether a reference such as x.123 is a model name or a subcircuit name (123 model in “x” subcircuit).

If you do not specify an element and model name, HSPICE checks all elements of this type for bias voltage (you must include type in the BIASCHK card). However, if type is subckt at least one element or model name must be specified in the .BIASCHK command; otherwise, a warning message is issued and this command is ignored.

Example 1

This example uses the .BIASCHK command to monitor an expression:

```
.biaschk 'v(1)' min='v(2)*2' simulation= op
```

Example 2

These examples use the .BIASCHK commands to monitor element and model types between to specified terminals.

- Monitor MOSFET element m1

  ```
  .biaschk nmos terminal1=ng terminal2=ns simulation=tr name=m1
  ```

- Monitor MOSFET model m1 whose bias voltage exceeds 2.5 V and interval exceeds 5 ns

  ```
  .biaschk nmos terminal1=nb terminal2=ng limit=2.5
  + mname=m1 interval=5n
  ```

Example 3

These examples use .BIASCHK commands that do not require terminal specifications.

- Monitor MOS transistor region of operation

```
.biaschk mos region=saturation name=x1.m1 mname=nch name=m2
```

- Monitor MOS transistor length and width

```
.biaschk mos monitor=l mname=m* p* min=1u minu=op
```

Interactions with Other Options

If you set `.OPTION BIAWARN` to `1`, HSPICE immediately outputs a warning message that includes the element name, time, terminals and model name when the limit is exceeded during the analysis you define. If you set the `autostop` keyword, HSPICE automatically stops at that situation.

If you set `.OPTION BIASFILE`, HSPICE outputs the summary into a file defined in that option. Otherwise, it is output to a *.lis file.

If you set `.OPTION BIASINTERVAL` to `0`, the keyword `interval` is then neglected. `BIASINTERVAL` values `1`, `2`, or `3` provide different details in warning messages. For example, when all violation regions of elements are expected, set `interval=0` and `.OPTION BIASINTERVAL=3`.

If you set `.OPTION BIASPARALLEL` to `1`, the keyword *mname* must be used and `monitor` must be set to `v` to invoke parallel element elimination.

If you set `.OPTION BIASNODE` to `1`, the name of the node in the netlist is used instead of the output port name for each element.

See Also

.OPTION BIASFILE
.OPTION BIASINTERVAL
.OPTION BIASNODE
.OPTION BIASPARALLEL
.OPTION BIAWARN

.CHECK EDGE

Verifies that a triggering event provokes an appropriate RISE or FALL action in HSPICE RF.

Syntax

```
.CHECK EDGE (ref RISE | FALL min max RISE | FALL)
+ node1 [node2 ...] (hi lo hi_th low_th)
```

Arguments

Argument	Description
ref	Name of the reference signal.
min	Minimum time.
max	Maximum time.
node1 *node2* ...	List of nodes to which you apply the edge condition.
hi lo hi_th lo_th	Logic levels for the timing check.

Description

Use a `.CHECK EDGE` command to verify that a triggering event provokes an appropriate RISE or FALL action within the specified time window.

Example

This example sets the condition that the rising action of the clock (clk) triggers the falling edge of VOUTA within 1 to 3 ns, as shown in Figure 1:

```
.CHECK EDGE (clk RISE 1ns 3ns FALL) VOUTA
```

Values for hi, lo, and the thresholds were defined in a `.CHECK GLOBAL_LEVEL` command placed earlier in the netlist.

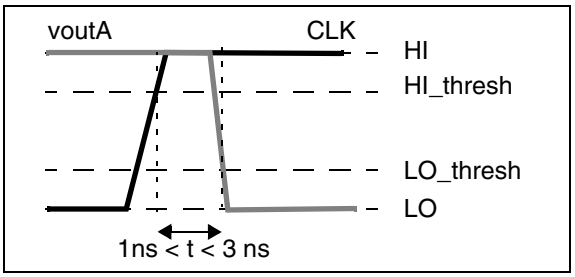

Figure 1 EDGE Example

See Also

.CHECK HOLD
.CHECK GLOBAL_LEVEL
.CHECK SETUP

.CHECK FALL

Verifies that a fall time occurs within a specified time window in HSPICE RF.

Syntax

```
.CHECK FALL (min max) node1 [node2 ...]
(hi lo hi_th lo_th)
```

Arguments

Argument	Description
min	Lower boundary for the time window.
max	Upper limit for the time window.
node1 node2 ...	List of all nodes to check.
hi lo hi_th lo_th	Logic levels for the timing check.

Description

Use a `.CHECK FALL` command verifies that a fall time occurs within the specified window of time.

See Also

.CHECK GLOBAL_LEVEL
.CHECK RISE
.CHECK SLEW

.CHECK GLOBAL_LEVEL

Globally sets specified high and low definitions for all CHECK commands in HSPICE RF.

Syntax

```
.CHECK GLOBAL_LEVEL (hi lo hi_th lo_th)
```

Arguments

Argument	Description
hi	Value for logic high.
lo	Value for logic low.
hi_th	Is the minimum value considered high.
lo_th	Is the maximum value considered low.

Description

Use this command to globally set the desired high and low definitions for all CHECK commands. The high and low definitions can be either numbers or expressions, and *hi_th* and *lo_th* can be either absolute values or percentages if punctuated with the % symbol. You can also locally set different logic levels for individual timing checks.

Example 1

This example defines a logic high as 5 volts and a logic low as 0 volts. A voltage value as small as 4 V is considered high, while a value up to 1 V is low.

```
.CHECK GLOBAL_LEVEL (5 0 4 1)
```

Example 2

This example illustrates an alternative definition for the first example:

```
.CHECK GLOBAL_LEVEL (5 0 80% 20%)
```

See Also

.CHECK EDGE
.CHECK FALL
.CHECK HOLD
.CHECK IRDROP
.CHECK RISE
.CHECK SLEW

.CHECK HOLD

Ensures that specified signals do not switch for a specified period of time in HSPICE RF.

Syntax

```
.CHECK HOLD (ref RISE | FALL duration RISE | FALL)
+ node1 [node2 ...] (hi lo hi_th low_th)
```

Arguments

Argument	Description
ref	Reference or trigger signal.
duration	Minimum time required after the triggering event before the specified nodes can rise or fall.
node1 node2 ...	List of nodes for which the HOLD condition applies.
hi lo hi_th lo_th	Logic levels for the timing check.

Description

Use this command to ensure that the specified signals do not switch for a specific period of time.

Example

This example specifies that vin* (such as vin1, vin2, and so on), must not switch for 2ns after every falling edge of nodeA (see Figure 2).

```
.CHECK HOLD (nodeA FALL 2ns RISE) vin*
```

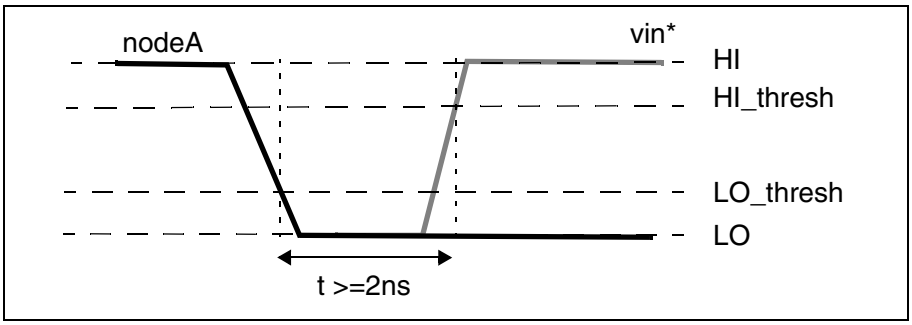

Figure 2 HOLD Example

See Also

.CHECK EDGE
.CHECK GLOBAL_LEVEL
.CHECK SETUP

.CHECK IRDROP

Verifies that IR drop does not fall below or exceed a specified value in HSPICE RF.

Syntax

```
.CHECK IRDROP (volt_val time duration) node1 [node2 ...]
+ (hi lo hi_th low_th)
```

Arguments

Argument	Description
volt_val	Limiting voltage value. ■ A positive *volt_val* (voltage value) indicates ground bounce checking. ■ A negative *volt_val* denotes VDD drop.
duration	Maximum allowable time. If you set duration to 0, then HSPICE RF reports every glitch that strays beyond the specified *volt_val*.
node1 < node2 ... >	List of nodes for which the IR drop checking applies.
hi lo hi_th lo_th	Logic levels for the timing check.

Description

Use this command to verify that the IR drop does not fall below or exceed a specified value for a specified duration.

Example

This example specifies that `v1` must not fall below -2 volts for any duration exceeding 1ns (see Figure 3).

```
.CHECK IRDROP (-2 1ns) v1
```

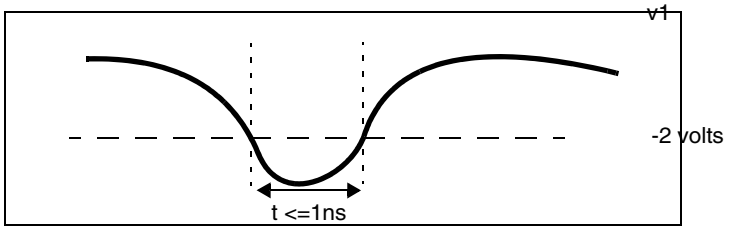

Figure 3 IR Drop Example

See Also

.CHECK EDGE
.CHECK GLOBAL_LEVEL
.CHECK SETUP

.CHECK RISE

Verifies that a rise time occurs within a specified time window in HSPICE RF.

Syntax

```
.CHECK RISE (min max) node1 [node2 ...] (hi lo hi_th lo_th)
```

Arguments

Argument	Description
min	Lower boundary for the time window.
max	Upper limit for the time window.
node1 < node2 ... >	List of all nodes to check.
hi lo hi_th lo_th	Logic levels for the timing check.

Description

Use this command to verify that a rise time occurs within the specified window of time.

Example

This example defines a window between 1.5ns and 2.2ns wide, in which the va and vb signals must complete their rise transition (see Figure 4). Values for the HI, LO, and the thresholds were defined in a `.CHECK GLOBAL_LEVEL` command placed earlier in the netlist.

```
.CHECK RISE (1.5ns 2.2ns) va vb
```

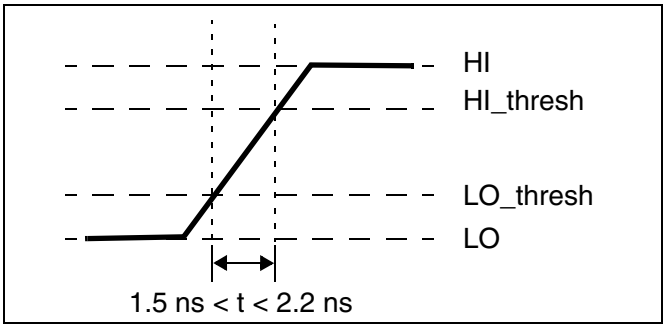

Figure 4 RISE Time Example

See Also

.CHECK GLOBAL_LEVEL
.CHECK FALL
.CHECK SLEW

.CHECK SETUP

Verifies that specified signals do not switch for a specified period of time in HSPICE RF.

Syntax

```
.CHECK SETUP (ref RISE | FALL duration RISE | FALL)
+ node1 [node2 ...] (hi lo hi_th low_th)
```

Arguments

Argument	Description
ref	Reference or trigger signal.
duration	Minimum time before the triggering event during which the specified nodes cannot rise or fall
node1 node2 ...	List of nodes for which the HOLD condition applies.
hi lo hi_th lo_th	Logic levels for the timing check.

Description

Use this command to verify that the specified signals do not switch for a specified period of time.

Example

This example specifies that v1 and v2 must not switch for 2 ns before every rising edge of nodeA (see Figure 5).

```
.CHECK SETUP (nodeA RISE 2ns FALL) v1 v2
```

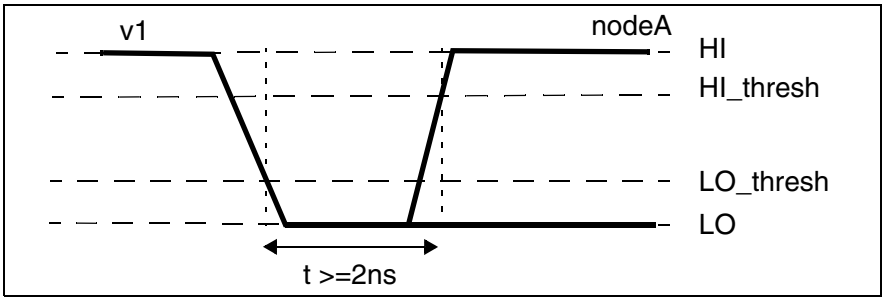

Figure 5 SETUP Example

See Also

.CHECK EDGE
.CHECK GLOBAL_LEVEL
.CHECK HOLD

.CHECK SLEW

Verifies that a slew rate occurs within a specified time window in HSPICE RF.

Syntax

```
.CHECK SLEW (min max) node1 [node2 ...](hi lo hi_th lo_th)
```

Arguments

Argument	Description
min	Lower boundary for the time window.
max	Upper limit for the time window.
node1 node2 ...	List of all nodes to check.
hi lo hi_th lo_th	Logic levels for the timing check.

Description

Use this command to verify that a slew rate occurs within specified time range.

Example

This example sets the condition that nodes starting with a* nodes must have a slew rate between (HI_thresh - LO_thresh)/3ns and (HI_thresh - LO_thresh)/1ns. If either node has a slew rate greater than that defined in the `.CHECK SLEW` command, HSPICE RF reports the violation in the .err file.

```
.CHECK SLEW (1ns 3ns) a* (3.3 0 2.6 0.7)
```

The slew rate check in Figure 6 defines its own *hi*, *lo*, and corresponding threshold values, as indicated by the four values after the node names.

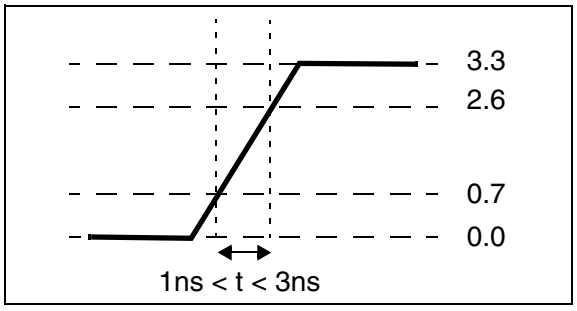

Figure 6 SLEW Example

See Also

.CHECK FALL
.CHECK GLOBAL_LEVEL
.CHECK RISE

.CONNECT

Connects two nodes together; the first node replaces the second node in the simulation.

Syntax

```
.CONNECT node1 node2
```

Arguments

Argument	Description
node1	Name of the first of two nodes to connect together
node2	Name of the second of two nodes to connect together. This node is replaced by Node1, which is the first node, in the simulation.

Description

Use this command to connect two nodes together in your netlist. This causes the simulation to evaluate the two nodes as if they were only one node that uses the name of the first node. The name of the second node is not recognized in the simulation. Both nodes must be at the same level in the circuit design that you are simulating: you cannot connect nodes that belong to different subcircuits.

Example 1

```
     ...
.subckt eye_diagram node1 node2 ...
.connect node1 node2
...
.ends
```

This is now the same as the following:

```
    ...
.subckt eye_diagram node1 node1 ...
...
.ends
...
```

HSPICE reports the following error message:

```
**error**: subcircuit definition duplicates node node1
```

To apply any HSPICE command to *node2*, apply it to *node1,* instead. Then, to change the netlist construction to recognize *node2*, use an .ALTER command.

Example 2

```
*example for  .connect
vcc 0 cc 5v
r1  0 1  5k
r2  1 cc 5k
.tran 1n 10n
.print i(vcc) v(1)
.alter
.connect cc 1
.end
```

The first .TRAN simulation includes two resistors. Later simulations have only one resistor because r2 is short-circuited by connecting cc with 1. v(1) does not print out, but v(cc) prints out instead.

Use multiple .CONNECT commands to connect several nodes together.

Example 3

```
.CONNECT node1 node2
.CONNECT node2 node3
```

This example connects both *node2* and *node3* to *node1*. All connected nodes must be in the same subcircuit or all in the main circuit. The first HSPICE simulation evaluates only *node1*; *node2* and *node3* are the same node as *node1*. Use .ALTER commands to simulate *node2* and *node3*.

If you set .OPTION NODE, then HSPICE prints out a node connection table.

Example 4

```
vcc cc 0 5v
r1  cc net1  5k
r2  net1 net2 5k
c1 net2 0 1n
.tran 1n 10n
.connect net2 0
.print i(vcc) v(net2)
.end
```

This causes the circuit elements to be connected as:

```
vcc cc net2 5v
r1  cc net1  5k
r2  net1 net2 5k
c1 net2 net2 1n
.tran 1n 10n
.connect net2 0
.print i(vcc) v(net2)
.end
```

HSPICE reports the following error message:

```
**error**  no dc path to ground from node
```

...for the elements `vcc r1` and `r2`, since there is now no ground node in the netlist. The correct way to connect `net2` to ground is to specify the `.CONNECT` command as:

```
.connect 0 net2
```

See Also

.ALTER
.OPTION NODE

.DATA

Concatenates or column-laminates data sets to optimize measured I-V, C-V, transient, or S-parameter data.

Syntax

Inline command

```
.DATA datanm pnam1 [pnam2 pnam3 ... pnamxxx]
+   pval1 [pval2  pval3 ... pvalxxx]
+   pval1' [pval2' pval3' ... pvalxxx']
.ENDDATA
```

External File command for concatenated data files

```
.DATA datanm MER
+ FILE='filename1' pname1=colnum [pname2=colnum ...]
+ [FILE='filename2' pname1=colnum
+ [pname2=colnum ...] ... [OUT='fileout']
.ENDDATA
```

Column Laminated command (not available for HSPICE RF)

```
.DATA datanm LAM
+ FILE='filename1' pname1=colnum
+ [pname2=colnum ...]
+ [FILE='filename2' pname1=colnum
+ pname2=colnum ...] ... [OUT='fileout']
.ENDDATA
```

Arguments

Argument	Description
column	Column number in the data file for the parameter value. The column does not need to be the same between files.
datanm	Data name—referenced in the .TRAN, .DC, or .AC command.
filenamei	Data file to read. HSPICE concatenates files in the order they appear in the .DATA command. You can specify up to 10 files.

Argument	Description
fileouti	Data file name, where simulation writes concatenated data. This file contains the full syntax for an inline .DATA command and can replace the .DATA command that created it in the netlist. You can output the file and use it to generate one data file from many.
LAM	Column-laminated (parallel merging) data files to use.
MER	Concatenated (series merging) data files to use.
pnami	Parameter names—used for source value, element value, device size, model parameter value, and so on. You must declare these names in a .PARAM command.
pvali	Parameter value.

Description

Use the .DATA command to concatenate or column-laminate data sets to optimize measured I-V, C-V, transient, or S-parameter data.

You can also use the .DATA command for a first or second sweep variable when you characterize cells and test worst-case corners. Simulation reads data measured in a lab, such as transistor I-V data, one transistor at a time in an outer analysis loop. Within the outer loop, the analysis reads data for each transistor (IDS curve, GDS curve, and so on), one curve at a time in an inner analysis loop.

Data-driven analysis syntax requires a .DATA command and an analysis command that contains a DATA=dataname keyword.

The .DATA command specifies parameters that change values, and the sets of values to assign during each simulation. The required simulations run as an internal loop. This bypasses reading-in the netlist and setting-up the simulation, which saves computing time. In internal loop simulation you can also plot simulation results against each other and print them in a single output.

You can enter any number of parameters in a .DATA block. The .AC, .DC, and .TRAN commands can use external and inline data provided in .DATA commands. For example, to specify the circuit temperature for an HSPICE simulation you can use the .TEMP command, the TEMP parameter in the .DC, .AC, and .TRAN commands, or the TEMP/TEMPER parameter in the first column of the .DATA command.

The number of data values per line does not need to correspond to the number of parameters. For example, you do not need to enter 20 values on each line in the .DATA block if each simulation pass requires 20 parameters: the program reads 20 values on each pass, however the values are formatted.

Each .DATA command can contain up to 50 parameters. If you need more than 50 parameters in a single .DATA command, place 50 or less parameters in the .DATA command, and use .ALTER commands for the other parameters.

HSPICE refers to .DATA commands by their data names so each data name must be unique. HSPICE supports three .DATA command formats:

- Inline data, which is parameter data, listed in a .DATA command block. The datanm parameter in a .DC, .AC, or .TRAN analysis command, calls this command. The number of parameters that HSPICE reads determines the number of columns of data. The physical number of data numbers per line does not need to correspond to the number of parameters. For example, if the simulation needs 20 parameters you do not need 20 numbers per line.
- Data that is concatenated from external files. Concatenated data files are files with the same number of columns, placed one after another.
- Data that is column-laminated from external files. Column laminated data are columns of files with the same number of rows, arranged side-by-side.

To use external files with the .DATA format:

- Use the MER and LAM keywords to prepare HSPICE for external file data, rather than inline data.
- Use the FILE keyword to specify the external filename.
- Use simple file names, such as out.dat without single or double quotation marks (‘ ’ or “ ”), but use quotation marks when file names start with numbers, such as “1234.dat”.
- Use the proper case, since file names are case sensitive on UNIX systems.

For data-driven analysis, specify the start time (time 0) in the analysis command so that the analysis correctly calculates the stop time.

The following shows how different types of analyses use .DATA commands.

Operating point:

```
.DC DATA=dataname
```

DC sweep:

```
.DC vin 1 5 .25 SWEEP DATA=dataname
```

AC sweep:

```
.AC dec 10 100 10meg SWEEP DATA=dataname
```

TRAN sweep:

```
.TRAN 1n 10n SWEEP DATA=dataname
```

Example 1

```
* Inline .DATA statement
   .TRAN    1n   100n      SWEEP DATA=devinf
   .AC DEC    10   1hz   10khz   SWEEP DATA=devinf
   .DC TEMP   -55   125   10   SWEEP DATA=devinf
   .DATA   devinf   width   length   thresh   cap
  +      50u   30u   1.2v   1.2pf
  +       25u    15u   1.0v    0.8pf
  +       5u   2u   0.7v   0.6pf
.ENDDATA
```

HSPICE performs these analyses for each set of parameter values defined in the `.DATA` command. For example, the program first uses the `width=50u`, `length=30u`, `thresh=1.2v`, and `cap=1.2pf` parameters to perform `.TRAN`, `.AC`, and `.DC` analyses.

HSPICE then repeats the analyses for `width=25u`, `length=15u`, `thresh=1.0v`, and `cap=0.8pf`, and again for the values on each subsequent line in the `.DATA` block.

Example 2

```
* .DATA as the inner sweep
M1 1 2 3 0 N   W=50u   L=LN
   VGS 2 0 0.0v
   VBS 3 0 VBS
   VDS 1 0 VDS
   .PARAM VDS=0 VBS=0 L=1.0u
   .DC DATA=vdot
   .DATA vdot
      VBS   VDS   L
       0    0.1   1.5u
        0   0.1   1.0u
        0   0.1   0.8u
       -1    0.1   1.0u
      -2    0.1    1.0u
       -3   0.1   1.0u
        0   1.0    1.0u
        0    5.0   1.0u
.ENDDATA
```

This example performs a DC sweep analysis for each set of VBS, VDS, and L parameters in the .DATA vdot block. That is, HSPICE runs eight DC analyses one for each line of parameter values in the .DATA block.

Example 3

```
* .DATA as the outer sweep
   .PARAM W1=50u W2=50u L=1u CAP=0
   .TRAN 1n 100n SWEEP DATA=d1
   .DATA d1
      W1     W2    L      CAP
      50u    40u   1.0u   1.2pf
      25u    20u   0.8u   0.9pf
.ENDDATA
```

In this example:

- The default start time for the .TRAN analysis is 0.
- The time increment is 1 ns.
- The stop time is 100 ns.

These values result in transient analyses at every time value from 0 to 100 ns in steps of 1 ns by using the first set of parameter values in the .DATA d1 block. Then HSPICE reads the next set of parameter values and does another 100 transient analyses. It sweeps time from 0 to 100 ns in 1 ns steps. The outer sweep is time and the inner sweep varies the parameter values. HSPICE performs 200 analyses: 100 time increments, times 2 sets of parameter values.

Example 4

```
* External File .DATA for concatenated data files
.DATA datanm MER
   + FILE=filename1 pname1 = colnum
   + <pname2=colnum ...>
   + <FILE=filename2 pname1=colnum
   + <pname2=colnum ...>>
   + ...
   + <OUT=fileout>
.ENDDATA
```

Example 5

If you concatenate the three files (*file1*, *file2*, and *file3*).

```
   file1   file2   file3
   a a a   b b b   c c c
   a a a   b b b   c c c
   a a a
```

The data appears as follows:

```
a a a
a a a
a a a
b b b
b b b
c c c
c c c
```

The number of lines (rows) of data in each file does not need to be the same. The simulator assumes that the associated parameter of each column of the A file is the same as each column of the other files.

The `.DATA` command for this example is:

```
* External File .DATA statement
.DATA inputdata MER
   FILE='file1' p1=1 p2=3 p3=4
   FILE='file2' p1=1
   FILE='file3'
.ENDDATA
```

This listing concatenates *file1*, *file2*, and *file3* to form the `inputdata` data set. The data in *file1* is at the top of the file, followed by the data in *file2*, and *file3*. The `inputdata` in the `.DATA` command references the data name specified in either the `.DC`, `.AC`, or `.TRAN` analysis commands. The parameter fields specify the column that contains the parameters (you must already have defined the parameter names in `.PARAM` commands). For example, the values for the p1 parameter are in column 1 of *file1* and *file2*. The values for the *p2* parameter are in column 3 of *file1*.

For data files with fewer columns than others, HSPICE assigns values of zero to the missing parameters.

Example 6

(HSPICE only) Three files (D, E, and F) contain the following columns of data:

```
File D   File E   File F
d1 d2 d3   e4 e5   f6
d1 d2 d3   e4 e5   f6
d1 d2 d3   e4 e5   f6
```

The laminated data appears as follows:

```
d1 d2 d3   e4 e5   f6
d1 d2 d3   e4 e5   f6
d1 d2 d3   e4 e5   f6
```

The number of columns of data does not need to be the same in the three files.

The number of lines (rows) of data in each file does not need to be the same. HSPICE interprets missing data points as zero.

The .DATA command for this example is:

```
* Column-Laminated .DATA statement
.DATA dataname LAM
   FILE='file1' p1=1 p2=2 p3=3
   FILE='file2' p4=1 p5=2
   OUT='fileout'
.ENDDATA
```

This listing laminates columns from *file1* and *file2* into the *fileout* output file. Columns one, two, and three of *file1* and columns one and two of *file2* are designated as the columns to place in the output file. You can specify up to 10 files per .DATA command.

If you run HSPICE on a different machine than the one on which the input data files reside (such as when you work over a network), use full path names instead of aliases. Aliases might have different definitions on different machines.

See Also

.AC
.DC
.ENDDATA
.PARAM
.TRAN

.DC

Performs several types of sweeps during DC analysis.

Syntax

Sweep or Parameterized Sweep:

```
.DC var1 START=start1 STOP=stop1 STEP=incr1
.DC var1 START=[param_expr1]
+ STOP=[param_expr2] STEP=[param_expr3]
.DC var1 start1 stop1 incr1
+ [SWEEP var2 type np start2 stop2]
.DC var1 start1 stop1 incr1 [var2 start2 stop2 incr2]
```

Data-Driven Sweep:

```
.DC var1 type np start1 stop1 [SWEEP DATA=datanm]
.DC DATA=datanm [SWEEP var2 start2 stop2 incr2]
.DC DATA=datanm
```

Monte Carlo:

```
.DC var1 type np start1 stop1 [SWEEP MONTE=MCcommand]
.DC MONTE=MCcommand
```

Optimization:

```
.DC DATA=datanm OPTIMIZE=opt_par_fun
+ RESULTS=measnames MODEL=optmod
.DC var1 start1 stop1 SWEEP OPTIMIZE=OPTxxx
+ RESULTS=measname MODEL=optmod
```

Arguments

Argument	Description
DATA=*datanm*	*Datanm* is the reference name of a .DATA command.
incr1 ...	Voltage, current, element, or model parameters; or temperature increments.
MODEL	Optimization reference name. The .MODEL OPT command uses this name in an optimization analysis

Argument	Description
MONTE= MCcommand	Where MCcommand can be any of the following: ▪ *val* Specifies the number of random samples to produce. ▪ *val firstnum=num* Specifies the sample number on which the simulation starts. ▪ *list num* Specifies the sample number to execute. ▪ list(*num1:num2 num3 num4:num5*) Samples from *num1* to *num2*, sample *num3*, and samples from *num4* to *num5* are executed (parentheses are optional).
np	Number of points per decade or per octave or just number of points, based on which keyword precedes it.
OPTIMIZE	Specifies the parameter reference name, used for optimization in the `.PARAM` command
RESULTS	Measure name used for optimization in the `.MEASURE` command
start1 ...	Starting voltage, current, element, or model parameters; or temperature values. If you use the POI (list of points) variation type, specify a list of parameter values, instead of *start stop*. HSPICE supports the *start* and *stop* syntax; HSPICE RF does not.
stop1 ...	Final voltage, current, any element, model parameter, or temperature values.
SWEEP	Second sweep has a different type of variation (DEC, OCT, LIN, POI, or DATA command; or MONTE=*val*)
TEMP	Temperature sweep.
type	Can be any of the following keywords: ▪ DEC — decade variation ▪ OCT — octave variation ▪ LIN — linear variation ▪ POI — list of points

Argument	Description
var1 ...	▪ Name of an independent voltage or current source, or ▪ Name of any element or model parameter, or ▪ TEMP keyword (indicating a temperature sweep). HSPICE supports a source value sweep, which refers to the source name (SPICE style). However, if you select a parameter sweep, a .DATA command, and a temperature sweep, then you must select a parameter name for the source value. A later .DC command must refer to this name. The parameter must not start with the TEMP keyword. The *var1* parameter should be defined in advance using the .PARAM command.
firstrun	The *val* value specifies the number of Monte Carlo iterations to perform. The *firstrun* value specifies the desired number of iterations. HSPICE runs from num1 to num1+val-1.
list	The iterations at which HSPICE performs a Monte Carlo analysis. You can write more than one number after list. The colon represents “from ... to ...". Specifying only one number makes HSPICE run at only the specified point.

Description

You can use the .DC command in DC analysis to:

- Sweep any parameter value.
- Sweep any source value.
- Sweep temperature range.
- Perform a DC Monte Carlo (random sweep) analysis.
- Perform a data-driven sweep.
- Perform a DC circuit optimization for a data-driven sweep.
- Perform a DC circuit optimization by using start and stop.
- Perform a DC model characterization.

The format for the .DC command depends on the application that uses it. The DC sweep functionality is enhanced by use of the GSHUNT algorithm.

Example 1

```
.DC VIN 0.25 5.0 0.25
```

This example sweeps the value of the VIN voltage source from 0.25 volts to 5.0 volts in increments of 0.25 volts.

Example 2

```
.DC VDS 0 10 0.5 VGS 0 5 1
```

Example 2 sweeps the drain-to-source voltage from 0 to 10 V in 0.5 V increments at VGS values of 0, 1, 2, 3, 4, and 5 V.

Example 3

```
.DC TEMP -55 125 10
```

Example 3 starts a DC analysis of the circuit from -55° C to 125° C in 10° C increments.

Example 4

```
.DC TEMP POI 5 0 30 50 100 125
```

This script runs a DC analysis at five temperatures: 0, 30, 50, 100, and 125° C.

Example 5

```
.DC xval 1k 10k .5k SWEEP TEMP LIN 5 25 125
```

Example 5 runs a DC analysis on the circuit at each temperature value. The temperatures result from a linear temperature sweep from 25° C to 125° C (five points), which sweeps a resistor value named `xval` from 1 k to 10 k in 0.5 k increments.

Example 6

```
.DC DATA=datanm SWEEP par1 DEC 10 1k 100k
```

Example 6 specifies a sweep of the *par1* value from 1 k to 100 k in increments of 10 points per decade.

Example 7

```
.DC par1 DEC 10 1k 100k SWEEP DATA=datanm
```

Example 7 requests a DC analysis at specified parameters in the `.DATA datanm` command. It also sweeps the *par1* parameter from 1k to 100k in increments of 10 points per decade.

Example 8

```
.DC par1 DEC 10 1k 100k SWEEP MONTE=30
```

Example 8 invokes a DC sweep of the *par1* parameter from 1k to 100k by 10 points per decade by using 30 randomly generated (Monte Carlo) values.

Example 9

```
*file: bjtschmt.sp   bipolar schmitt trigger
.OPTION post=2
vcc 6 0 dc 12
vin 1 0 dc 0 pwl(0,0 2.5u,12 5u,0)
cb1 2 4 .1pf
rc1 6 2 1k
rc2 6 5 1k
rb1 2 4 5.6k
rb2 4 0 4.7k
re 3 0 .47k
diode 0 1 dmod
q1 2 1 3 bmod 1 ic=0,8
q2 5 4 3 bmod 1 ic=.5,0.2
.dc vin 0,12,.1
.model dmod d is=1e-15 rs=10
.model bmod npn is=1e-15 bf=80 tf=1n
+ cjc=2pf cje=1pf rc=50 rb=100 vaf=200
.probe v(1) v(5)
.print
.end
```

Example 9 is a Schmitt Trigger script.

Example 10

```
.DC par1 DEC 10 1k 100k SWEEP MONTE=10 firstrun=11
```

Example 10 invokes a DC sweep of the *par1* parameter from 1k to 100k by 10 points per decade and uses 10 Monte Carlo) values from 11th to 20th trials.

Example 11

```
.DC par1 DEC 10 1k 100k SWEEP MONTE=list(10 20:30 35:40 50)
```

Example 11 invokes a DC sweep of the *par1* parameter from 1k to 100k by 10 points per decade and a Monte Carlo analysis at the 10th trial, then from the 20th to the 30th trials, followed by the 35th to 40th trials and finally at the 50th trial.

See Also

.MODEL
.OPTION DCIC
.PARAM

.DCMATCH

Calculates the effects of variations on a circuit's DC characteristics.

Syntax

`.DCMATCH` *`OUTVAR`* `[THRESHOLD=`*`T`*`] [FILE=`*`string`*`] [INTERVAL=`*`Int`*`]`

Arguments

Argument	Description
OUTVAR	One or more node voltages, voltage differences for a node pair, or currents through an independent voltage source.
THRESHOLD	Report devices with a relative contribution above Threshold in the summary table. ▪ T=0: reports results for all devices ▪ T<0: suppresses table output; however, individual results are still available through .PROBE or .MEASURE commands. The upper limit for T is 1, but at least 10 devices are reported or all if there are less than 10. Default value is 0.01.
FILE	Valid file name for the output tables. Default is basename.dm# where “#” is the usual sequence number for HSPICE output files.
INTERVAL	Applies only if a DC sweep is specified. `Int` is a positive integer. A summary is printed at the first sweep point, then for each subsequent increment of *Int* and then if not already printed at the final sweep point. Only single sweeps are supported.

Description

Use this command to calculate the effects of variations in device characteristics on the DC solution of a circuit.

You can perform only one DCMATCH analysis per simulation. Only the last `.DCMATCH` command is used in case more than one in present. The others are discarded with warnings.

Example 1

```
.DCMatch V(9) V(4,2) I(VCC)
```

HSPICE reports DCmatch variations on the voltage of node 9, the voltage difference between nodes 4 and 2, and on the current through the source VCC.

Example 2

```
.DC XVal Start=1K Stop=9K Step=1K
.DCMATCH V(vcc) interval=3
```

The variable XVal is being swept in the .DC command. It takes nine values in sequence from 1k to 9k in increments of 1k. Tabular output for the .DCMATCH command is only generated for the set XVal={1k, 4k, 7k, 9k}.

See Also

.DC
.MEASURE(DCMATCH)
.PROBE

.DCVOLT

Sets initial conditions in HSPICE.

Syntax

```
.DCVOLT V(node1)=val1 V(node2)=val2 ...
.DCVOLT V node1 val1 [node2 val2 ...]
```

Arguments

Argument	Description
val1 ...	Voltages. The significance of these voltages depends on whether you specify the UIC parameter in the .TRAN command.
node1 ...	Node numbers or names can include full paths or circuit numbers.

Description

Use the .IC command or the .DCVOLT command to set transient initial conditions in HSPICE. How it initializes depends on whether the .TRAN analysis command includes the UIC parameter.

If you specify the UIC parameter in the .TRAN command, HSPICE does not calculate the initial DC operating point but directly enters transient analysis. Transient analysis uses the .IC initialization values as part of the solution for timepoint zero (calculating the zero timepoint applies a fixed equivalent voltage source). The .IC command is equivalent to specifying the IC parameter on each element command but is more convenient. You can still specify the IC parameter, but it does not take precedence over values set in the .IC command.

If you do *not* specify the UIC parameter in the .TRAN command, HSPICE computes the DC operating point solution before the transient analysis. The node voltages that you specify in the .IC command are fixed to determine the DC operating point. Transient analysis releases the initialized nodes to calculate the second and later time points.

Example

```
.DCVOLT 11 5 4 -5 2 2.2
```

See Also

.IC
.TRAN

.DEL LIB

Removes library data from memory for HSPICE/HSPICE RF.

Syntax

```
.DEL LIB '[file_path]file_name' entry_name
.DEL LIB libnumber entryname
```

Arguments

Argument	Description
entry_name	Name of entry used in the library call command to delete.
filen_ame	Name of a file to delete from the data file; the file path, plus the file name, can be up to 256 characters long. You can use any file name that is valid for the operating system that you use. Enclose the file path and file name in single or double quotation marks.
file_path	Path name of a file if the operating system supports tree-structured directories.
libnumber	Library number, used in the library call command to delete.

Description

Use this command to remove library data from memory. The next time you run a simulation, the .DEL LIB command removes the .LIB call command with the same library number and entry name from memory. You can then use a .LIB command to replace the deleted library. In this way, .DEL LIB helps you avoid name conflicts.

You can use the .DEL LIB command with the .ALTER command.

Example 1

Example 1 calculates a DC transfer function for a CMOS inverter using these steps:

1. HSPICE simulates the device by using the NORMAL inverter model from the MOS.LIB library.
2. Using the .ALTER block and the .LIB command, HSPICE substitutes a faster CMOS inverter, FAST for NORMAL.
3. HSPICE then resimulates the circuit.

4. Using the second .ALTER block, HSPICE executes DC transfer analysis simulations at three different temperatures and with an n-channel width of 100 mm, instead of 15 mm.
5. HSPICE also runs a transient analysis in the second .ALTER block and uses a .MEASURE command to measure the rise time of the inverter.

```
FILE1: ALTER1 TEST CMOS INVERTER
   .OPTION ACCT LIST
   .TEMP 125
   .PARAM WVAL=15U VDD=5
   *
   .OP
   .DC VIN 0 5 0.1
   .PRINT DC V(3) V(2)
   *
   VDD 1 0 VDD
   VIN 2 0
   *
   M1 3 2 1 1 P 6U 15U
   M2 3 2 0 0 N 6U W=WVAL
   *
.LIB 'MOS.LIB' NORMAL
.ALTER
   .DEL LIB 'MOS.LIB' NORMAL $removes LIB from memory
.DEL LIB 'MOS.LIB' NORMAL     $removes normal library from memory
.OPTION BRIEF=1               $suppress printing of details
.LIB 'MOS.LIB' FAST           $get fast model library
.OPTION BRIEF=0               $resume normal printing
.ALTER
   .OPTION NOMOD OPTS         $suppress printing model
                              $parameters and print the
                              $option summary
   .TEMP -50 0 50             $run with different temperatures
   .PARAM WVAL=100U VDD=5.5   $change the parameters using
   VDD 1 0 5.5                $VDD 1 0 5.5 to change the power
                              $supply VDD value doesn't work
   VIN 2 0 PWL 0NS 0 2NS 5 4NS 0 5NS 5
                              $change the input source
   .OP VOL                    $node voltage table of
                              $operating points
   .TRAN 1NS 5NS              $run with transient also
   M2 3 2 0 0 N 6U WVAL       $change channel width
   .MEAS SW2 TRIG V(3) VAL=2.5 RISE=1 TARG V(3)
   + VAL=VDD CROSS=2          $measure output
   *
.END
```

Example 2

In this example, the .ALTER block adds a resistor and capacitor network to the circuit. The network connects to the output of the inverter and HSPICE simulates a DC small-signal transfer function.

```
FILE2: ALTER2.SP CMOS INVERTER USING SUBCIRCUIT
.OPTION LIST ACCT
.MACRO INV 1 2 3
M1 3 2 1 1 P 6U 15U
M2 3 2 0 0 N 6U 8U
.LIB 'MOS.LIB' NORMAL
.EOM INV
XINV 1 2 3 INV
VDD 1 0 5
VIN 2 0
.DC VIN 0 5 0. 1
.PRINT V(3) V(2)
.ALTER
.DEL LIB 'MOS.LIB' NORMAL
.TF V(3) VIN         $DC small-signal transfer
      $function
*
.MACRO INV 1 2 3               $change data within
      $subcircuit def
M1 4 2 1 1 P 100U 100U         $change channel length,width,also
                               $topology
M2 4 2 0 0 N 6U   8U           $change topology
R4 4 3 100                     $add the new element
C3 3 0 10P                     $add the new element
.LIB 'MOS.LIB' SLOW            $set slow model library
$.INC 'MOS2.DAT'               $not allowed to be used
                               $inside subcircuit, allowed
                               $outside subcircuit
.EOM INV
.END
```

See Also

.ALTER
.LIB

.DESIGN_EXPLORATION

Creates an Exploration Block to extract the parameters suitable for exploration from a netlist.

Syntax

```
.Design_Exploration
        Options
          Parameter Parameter_Name = value
          Parameter Parameter_Name = expression
            .Data BlockName
            Index    Name   Name, …
             …
            .EndData
.End_Design_Exploration
```

Arguments

If you want to explore only certain cells or subcircuits use:

Option	Description
`Option Explore_only Subckts= SubcktList`	This command is executed hierarchically—the specified subcircuits and all instantiated subcircuits and elements underneath are affected. Thus, if an inverter with name `INV1` is placed in a digital control block called `DIGITAL` and in an analog block `ANALOG`, and `Option Explore_only Subckts = ANALOG`, then the perturbations only affect the `INV1` in the analog block. You must create a new inverter `INV1analog`, with the new device sizes.
`Option Do_not_explore Subckts= SubcktList`	Excludes listed subcircuits.

The Export and non-export modes of exploration are distinguished by setting either:

Option	Description
`Option Export=yes`	Exports extraction data and runs one simulation with the original netlist
`Option Export=no`	(Default) Runs a simulation with Exploration data

The perturbation types are selected by setting either:

Option	Description
`Option Exploration_method= external Block_name= Block_name`	The Block_name is the same as the name specified in the `.DATA` block; HSPICE will sweep the row content with the *`EXCommand`* (see Executing Exploration in HSPICE).
`Option Ignore_exploration= yes|no`	(Default=no) HSPICE ignores the content in the design_exploration block, when `Ignore_exploration=yes`.
`Option Secondary_param= yes|no`	(Default = no) If `Secondary_param= yes`, HSPICE exports the MOSFET secondary instance parameters to a **.mex* file (created when `option export=yes`), and also permits the secondary parameters to be imported as a column header in the .DATA block (`option export=no`).

Description

Use the command to create an exploration block to extract prearrangers from a netlist to explore in the early stages of designing integrated circuits in CMOS technology. Exploration is currently supported for:

- Independent sources: DC value
- MOS devices: W, L, M, dtemp
- Resistors: R or W, L, M, dtemp
- Capacitors: C or W, L, M, dtemp

When designing circuits, the multiplicity factor `M` is always a positive integer, but the Exploration tool can request arbitrary positive values.

To preserve relationships which have been previously defined through expressions, exploration can only be applied to parameters which are defined with numerical values.

For a detailed description of the Exploration Block usage, see Exploration Block in the *HSPICE User Guide: Simulation and Analysis*.

.DISTO

Computes the distortion characteristics of the circuit in an AC analysis.

Syntax

```
.DISTO Rload [inter [skw2 [refpwr spwf]]]
```

Arguments and Parameters

The tables below describe the arguments and possible .DISTO values.

Argument	Description
Rload	Resistor element name of the output load resistor into which the output power feeds.
inter	Interval at which HSPICE prints a distortion-measure summary. Specifies a number of frequency points in the AC sweep (see the *np* parameter in the .AC command). ▪ If you omit inter or set it to zero, HSPICE does not print a summary. To print or plot the distortion measures, use the .PRINT command. ▪ If you set inter to 1 or higher, HSPICE prints a summary of the first frequency and of each subsequent inter-frequency increment. To obtain a summary printout for only the first and last frequencies, set inter equal to the total number of increments needed to reach fstop in the .AC command. For a summary printout of only the first frequency, set inter to greater than the total number of increments required to reach fstop. HSPICE prints an extensive summary from the distortion analysis for each frequency listed. Use the inter parameter in the .DISTO command to limit the amount of output generated.
skw2	Ratio of the second frequency (F2) to the nominal analysis frequency (F1) in the range 1e-3 < skw2 < 0.999. If you omit skw2, the default value is 0.9.
refpwr	Reference power level—used to compute the distortion products. If you omit refpwr, the default value is 1mW—measured in decibels magnitude (dbM). The value must be ≥ 1e-10.
spwf	Amplitude of the second frequency (F2). The value must be ≥ 1e-3. The default is 1.0.

These are the possible .DISTO values.

.DISTO Value	Description
DIM2	Intermodulation distortion, first difference. Relative magnitude and phase of the frequency component (F1 - F2).
DIM3	Intermodulation distortion, second difference. The relative magnitude and phase of the frequency component (2 · F1 - F2).
HD2	Second-order harmonic distortion. Relative magnitude and phase of the frequency component 2 · F1 (ignores F2).
HD3	Third-order harmonic distortion. Relative magnitude and phase of the frequency component 3 · F1 (ignores F2).
SIM2	Intermodulation distortion, sum. Relative magnitude and phase of the frequency component (F1 + F2).

Description

Use the .DISTO command to calculate the distortion characteristics of the circuit in an AC small-signal, sinusoidal, steady-state analysis. The program computes and reports five distortion measures at the specified load resistor. The analysis assumes that the input uses one or two signal frequencies.

- HSPICE uses the first frequency (F1, the nominal analysis frequency) to calculate harmonic distortion. The .AC command frequency-sweep sets it.
- HSPICE uses the optional second input frequency (F2) to calculate intermodulation distortion. To set it implicitly, specify the skw2 parameter, which is the F2/F1 ratio

HSPICE performs only one distortion analysis per simulation. If your design contains more than one .DISTO command, HSPICE runs only the last command. The .DISTO command calculates distortions for diodes, BJTs (levels 1, 2, 3, and 4), and MOSFETs (Level49 and Level53, Version 3.22). You can use the .DISTO command only with the .AC command.

Example

```
.DISTO RL 2 0.95 1.0E-3 0.75
```

See Also

.AC

.DOUT

Specifies the expected final state of an output signal.

Syntax

```
.DOUT nd VTH ( time state [time state])
.DOUT nd VLO VHI ( time state [time state])
```

The first syntax specifies a single threshold voltage, VTH. A voltage level above VTH is high; any level below VTH is low.

The second syntax defines a threshold for both a logic high (VHI) and low (VLO).

Note:

If you specify VTH, VLO, and VHI in the same command, then only VTH is processed and VLO and VHI are ignored.

Arguments and Parameters

Argument	Description
nd	Node name.
time	Absolute timepoint.
state	Expected condition of the nd node at the specified *time*: ■ 0: Expect ZERO,LOW. ■ 1: Expect ONE,HIGH. ■ Else: Do not care.
VTH	Single voltage threshold.
VLO	Voltage of the logic-low state.
VHI	Voltage of the logic-high state.

For both syntax cases, the *time*, *state* pair describes the expected output. During simulation, the simulated results are compared against the expected output vector.

Legal values for *state* are:

.DOUT State Value	Description
0	Expect ZERO
1	Expect ONE
X, x	Do not care
U, u	Do not care
Z, z	Expect HIGH IMPEDANCE (do not care)

Description

Use `.DOUT` to specify the expected final state of an output signal. During simulation, HSPICE compares simulation results with the expected output. If the states are different, an error report results.

Example

```
.PARAM VTH=3.0
.DOUT node1 VTH(0.0n 0 1.0n 1
+ 2.0n X 3.0n U 4.0n Z 5.0n 0)
```

The `.PARAM` command in this example sets the `VTH` variable value to 3. The `.DOUT` command, operating on the `node1` node, uses `VTH` as its threshold voltage.

When `node1` is above 3V, it is a logic 1; otherwise, it is a logic 0.

- At 0ns, the expected state of node1 is logic-low.
- At 1ns, the expected state is logic-high.
- At 2ns, 3ns, and 4ns, the expected state is “do not care.”
- At 5ns, the expected state is again logic low.

See Also

.MEASURE (or) .MEAS
.PARAM
.PRINT
.PROBE
.STIM

.EBD

Invokes IBIS Electronic Board Description (EBD) functionality.

Syntax

```
.EBD ebdname
+ file = 'filename'
+ component = 'compname:reference_designator'
+ {component = 'compname:reference_designator'...}
+ {usemap = package_value}
```

Arguments

Argument	Description
compname	Name after the .IBIS command that describes a component.
reference_designator	Reference designator that maps the component.
package_value	Value=0,1, 2,or 3 sets the package value (the same as option 'package' of .IBIS) of all components in [Reference Designator Map]. Default=0.

Description

Enter the .EBD command to use the IBIS EBD feature. HSPICE uses the EBD file when simulating the line connected with the reference_designator. When the keyword 'usemap' is added to the .EDB command, new components are added into the circuit according to the [Reference Designator Map]. The new component names are: 'Comp'+referenceName+'_'+ebdName

In Figure 7, CompU22_ebd and CompU23_ebd are added if U22 and U23 occur in [Reference Designator Map].

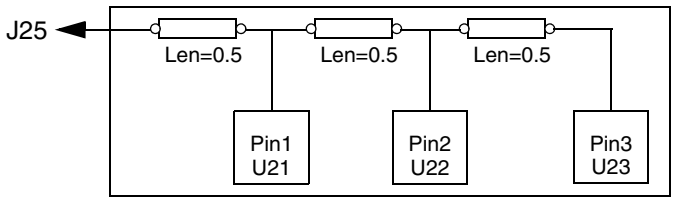

Figure 7 Circuit Connection for EBD Example

If a component is associated with both the keywords component and usemap, then the mapping relation defined by component only is used. The format of

the node name on the EBD side is *ebdName_pinName*. For example, the name **J25** is `ebd_J25`

Note:

If a component pin is not found and it is not a terminal node in the EBD path, then the name is used to designate the related node. For example, in Figure 7 on page 78, if U22_2 (here, 2 is the pin name) does not exist, then the node name will be ebd_U22_2.

If the component pin is a terminal node in the EBD path and is not found, then the node and the associated section will not be added into circuit. For example, in Figure 7, if U23_3 does not exist, then the section between Pin2 and Pin3 will be ignored and U22_2 is the terminal node.

Example

```
.ebd ebd
   + file = 'test.ebd'
   + model = '16Meg X 8 SIMM Module'
   + component = 'cmpnt:u21'
   * + usemap = 0
.ibis cmpnt
   + file = 'ebd.ibs'
   + component = 'SIMM'
   + hsp_ver=2003.09 nowarn
```

This example corresponds to the following *.ebd* file:

```
   .................
   [Begin Board Description] 16Meg X 8 SIMM Module
   .................
   [Pin List] signal_name
   J25          POWER5
   [Path Description] CAS_2
   Pin J25
   Len=0.5 L=8.35n C=3.34p R=0.01 /
   Node u21.1
   Len=0.5 L=8.35n C=3.34p R=0.01 /
   Node u22.2
   Len=0.5 L=8.35n C=3.34p R=0.01 /
   Node u23.3
```

See Also

.IBIS
.PKG

.ELSE

Precedes commands to be executed in a conditional block when preceding .IF and .ELSEIF conditions are false.

Syntax

.ELSE

Description

Use this command to precede one or more commands in a conditional block after the last .ELSEIF command, but before the .ENDIF command.

HSPICE/HSPICE RF executes these commands by default if the conditions are all false in the preceding .IF command and in all of the preceding .ELSEIF commands in the same conditional block.

For the syntax and a description of how to use the .ELSE command within the context of a conditional block, see the .IF command.

See Also

.ELSEIF
.ENDIF
.IF

.ELSEIF

Specifies conditions that determine whether HSPICE/HSPICE RF executes subsequent commands in a conditional block.

Syntax

```
.ELSEIF (condition)
```

Description

HSPICE executes the commands that follow the first .ELSEIF command only if *condition1* in the preceding .IF command is false and *condition2* in the first .ELSEIF command is true.

If *condition1* in the .IF command and *condition2* in the first .ELSEIF command are both false, then HSPICE moves on to the next .ELSEIF command if there is one.

If this second .ELSEIF condition is true, HSPICE executes the commands that follow the second .ELSEIF command, instead of the commands after the first .ELSEIF command.

HSPICE ignores the commands in all false .IF and .ELSEIF commands, until it reaches the first .ELSEIF condition that is true. If no .IF or .ELSEIF condition is true, HSPICE continues to the .ELSE command.

For the syntax and a description of how to use the .ELSEIF command within the context of a conditional block, see the .IF command.

See Also

.ELSE
.ENDIF
.IF

.END

Ends a simulation run in an input netlist file.

Syntax

.END [*comment*]

Arguments

Argument	Description
comment	Can be any comment. Typically, the comment is the name of the netlist file or of the simulation run that this command terminates.

Description

An .END command must be the last command in the input netlist file. The period preceding END is required. Text that follows the .END command is regarded as a comment only. An input file that contains more than one simulation run must include an .END command for each simulation run. You can concatenate several simulations into a single file.

Example

```
MOS OUTPUT
   .OPTION NODE NOPAGE
   VDS 3 0
   VGS 2 0
   M1 1 2 0 0 MOD1 L=4U W=6U AD=10P AS=10P
   .MODEL MOD1 NMOS VTO=-2 NSUB=1.0E15 TOX=1000
   + UO=550
   VIDS 3 1
   .DC   VDS 0 10 0.5    VGS 0 5 1
   .PRINT DC I(M1) V(2)
.END MOS OUTPUT
MOS CAPS
   .OPTION SCALE=1U SCALM=1U WL ACCT
   .OP
   .TRAN .1 6
   V1 1 0 PWL 0 -1.5V 6 4.5V
   V2 2 0 1.5VOLTS
   MODN1 2 1 0 0 M 10 3
   .MODEL M NMOS VTO=1 NSUB=1E15 TOX=1000
   + UO=800 LEVEL=1 CAPOP=2
   .PRINT TRAN V(1) (0,5) LX18(M1) LX19(M1) LX20(M1)
   + (0,6E-13)
.END MOS CAPS
```

.ENDDATA

Ends a .DATA block in an HSPICE input netlist file.

Syntax

.ENDDATA

Description

Use this command to terminate a .DATA block in an HSPICE input netlist.

See Also

.DATA

.ENDIF

Ends a conditional block of commands in an HSPICE input netlist file.

Syntax

```
.ENDIF
```

Description

Use this command to terminate a conditional block of commands that begins with an `.IF` command.

For the syntax and a description of how to use the `.ENDIF` command within the context of a conditional block, see the `.IF` command.

See Also

.ELSE
.ELSEIF
.IF

.ENDL

Ends a .LIB command in an HSPICE/HSPICE RF input netlist file.

Syntax

```
.ENDL
```

Description

Use this command to terminate a .LIB command in an HSPICE input netlist.

See Also

.LIB

.ENDS

Ends a subcircuit definition (.SUBCKT) in an HSPICE input netlist file.

Syntax

```
.ENDS subckt_name
```

Arguments

Argument	Description
subckt_name	Subcircuit name definition to end a command that begins with a .SUBCKT command.

Description

Use this command to terminate a .SUBCKT command. This command must be the last for any subcircuit definition that starts with a .SUBCKT command. You can nest subcircuit references (calls) within subcircuits in HSPICE.

Note:

Using -top subck_name on the command line effectively eliminates the need for the .subckt *subckt_name* and .ends *subckt_name*

Example 1

```
.ENDS mos_circuit
```

This example terminates a subcircuit named mos_circuit.

Example 2

```
.ENDS
```

Terminates all subcircuit definitions that begin with a .SUBCKT command.

See Also

.SUBCKT

.ENV

Performs standard envelope simulation in HSPICE RF.

Syntax

```
.ENV TONES=f1 [f2...fn] NHARMS=h1[h2...hn]
+ ENV_STEP=tstep ENV_STOP=tstop
```

Arguments

Parameter	Description
TONES	Carrier frequencies, in hertz.
NHARMS	Number of harmonics.
ENV_STEP	Envelope step size, in seconds.
ENV_STOP	Envelope stop time, in seconds.

Description

Use this command to perform standard envelope simulation.

The simulation proceeds just as it does in standard transient simulation, starting at `time=0` and continuing until `time=env_stop`. An HB analysis is performed at each step in time. You can use Backward-Euler (`BE`), trapezoidal (`TRAP`), or level-2 Gear (`GEAR`) integration.

- For `BE` integration, set `.OPTION SIM_ORDER=1`.
- For `TRAP`, set `.OPTION SIM_ORDER=2` (default) `METHOD=TRAP` (default).
- For `GEAR`, set `.OPTION SIM_ORDER=2` (default) `METHOD=GEAR`.

See Also

.ENVOSC
.HB
.PRINT
.PROBE

.ENVFFT

Performs Fast Fourier Transform (FFT) on envelope output in HSPICE RF.

Syntax

```
.ENVFFT output_var NP=value FORMAT=keyword
+ WINDOW=keyword ALFA=value
```

Arguments

Parameter	Description
output_var	Any valid output variable.
NP	Number of points to use in the FFT analysis. *NP* must be a power of 2. If not a power of 2, then it is automatically adjusted to the closest higher number that is a power of 2. The default is 1024.
FORMAT	Output format: NORM= normalized magnitude UNORM=unnormalized magnitude (default)
WINDOW	Window type to use: RECT=simple rectangular truncation window (default) BART=Bartlett (triangular) window HANN=Hanning window HAMM=Hamming window BLACK=Blackman window HARRIS=Blackman-Harris window GAUSS=Gaussian window KAISER=Kaiser-Bessel window
ALFA	Controls the highest side-lobe level and bandwidth for GAUSS and KAISER windows. The default is 3.0.

Description

Use this command to perform Fast Fourier Transform (FFT) on envelope output. This command is similar to the `.FFT` command. In HSPICE RF the data being transformed is complex. You usually want to do this for a specific harmonic of a voltage, current, or power signal.

See Also

.ENV
.ENVOSC
.FFT

.ENVOSC

Performs envelope simulation for oscillator startup or shutdown in HSPICE RF.

Syntax

```
.ENVOSC TONE=f1 NHARMS=h1 ENV_STEP=tstep ENV_STOP=tstop
+ PROBENODE=n1,n2,vosc [FSPTS=num, min, max]
```

Arguments

Parameter	Description
TONES	Carrier frequencies, in hertz.
NHARMS	Number of harmonics.
ENV_STEP	Envelope step size, in seconds.
ENV_STOP	Envelope stop time, in seconds.
PROBENODE	Defines the nodes used for oscillator conditions and the initial probe voltage value.
FSPTS	Specifies the frequency search points used in the initial small-signal frequency search. Usage depends on oscillator type.

Description

Use .EVOSC to perform envelope simulation for oscillator startup or shutdown. Oscillator startup or shutdown analysis must be helped along by converting a bias source from a DC description to a PWL description that either:

- Starts at a low value that supports oscillation and ramps up to a final value (startup simulation)
- Starts at the DC value and ramps down to zero (shutdown simulation).

In addition to computing the state variables at each envelope time point, the .ENVOSC command also computes the frequency. This command is applied to high-Q oscillators that take a long time to reach steady-state. For these circuits, standard transient analysis is too costly. Low-Q oscillators, such as typical ring oscillators are more efficiently simulated with standard transient analysis.

See Also

.ENV
.ENVFFT

.EOM

Ends a .MACRO command.

Syntax

.EOM *subckt_name*

Arguments

Argument	Description
subckt_name	Subcircuit name definition to end a macro that begins with a .SUBCKT command.

Description

Use this command to terminate a .MACRO command. .EOM must be the last for any subcircuit definition that starts with a .MACRO command. You can nest subcircuit references (calls) within subcircuits.

Example 1

```
.EOM diode_circuit
```

This example terminates a subcircuit named diode_circuit.

Example 2

```
.EOM
```

If you omit the subcircuit name as in this second example, this command terminates all subcircuit definitions that begin with a .MACRO command.

See Also

.MACRO

.FFT

Calculates the Discrete Fourier Transform (DFT) value used for spectrum analysis. Numerical parameters (excluding string parameters) can be passed to the `.FFT` command.

Syntax

Syntax # 1 Alphanumeric input

```
.FFT output_var [START=value] [STOP=value]
+ NP=value [FORMAT=keyword
+ [WINDOW=keyword] [ALFA=value]
+ [FREQ=value [FMIN=value] [FMAX=value]
```

Syntax #2 Numerics and expressions

```
.FFT [output_var] [START=param_expr1] [STOP=param_expr2]
+ [NP=param_expr3] [FORMAT=keyword]
+ [WINDOW=keyword] [ALFA=param_expr4]
+ [FREQ=param_expr5] [FMIN=param_expr6] [FMAX=param_expr7]
```

Syntax # Verilog-A Blocks

```
.FFT VAblock:SigName StartIdx=n1 StartIdx=n2
   SamplePeriod=val
+ ...
```

Arguments

Note:

Because the options can be set independently and might lead to conflicts, HSPICE conducts an error-check process preceding the FFT sampling process. The need for constant option-setting is described in the Description section below.

Argument	Description
output_var	Any valid output variable, such as voltage, current, or power.
START	Start of the output variable waveform to analyze. Defaults to the START value in the .TRAN command (tstart), which defaults to 0.
FROM	An alias for START in `.FFT` commands.

Argument	Description
STOP	End of the output variable waveform to analyze. Defaults to the TSTOP value in the .TRAN command.
TO	An alias for STOP, in .FFT commands.
NP	Number of points to use in the FFT analysis. NP must be a power of 2. If NP is not a power of 2, HSPICE automatically adjusts it to the closest higher number that is a power of 2. The default is 1024.
FORMAT	Output format: ▪ NORM= normalized magnitude (default) ▪ UNORM=unnormalized magnitude
WINDOW	Window can be one of the following types: ▪ RECT=simple rectangular truncation window (default). ▪ BART=Bartlett (triangular) window. ▪ HANN=Hanning window. ▪ HAMM=Hamming window. ▪ BLACK=Blackman window. ▪ HARRIS=Blackman-Harris window. ▪ GAUSS=Gaussian window. ▪ KAISER=Kaiser-Bessel window.
ALFA	Parameter to use in GAUSS and KAISER windows to control the highest side-lobe level, bandwidth, and so on. 1.0 <= ALFA <= 20.0 The default is 3.0
FREQ	Frequency to analyze. If FREQ is non-zero, the output lists only the harmonics of this frequency, based on FMIN and FMAX. HSPICE also prints the THD for these harmonics. The default is 0.0 (Hz).
FMIN	Minimum frequency for which HSPICE prints FFT output into the listing file. THD calculations also use this frequency. T=(STOP-START) The default is 1.0/T (Hz).
FMAX	Maximum frequency for which HSPICE prints FFT output into the listing file. THD calculations also use this frequency. The default is 0.5*NP*FM IN (Hz).
VAblock	Name of the Verilog-A block.

Argument	Description
SigName	Parameter name of the series output from Verilog-A. It should have the following type definition in Verilog-A block: `(* desc="SigName" *) real SigName[n1:n2];`
StartIdx	Start index of the series for FFT.
StopIdx	End index of the series for FFT; it must be greater than StartIdx; otherwise, HSPICE uses the whole series for the FFT process.
SamplePeriod	Time interval between two samples inside the series. It must be a positive value, the default value is 1 second.

Description

Use this command to calculate the Discrete Fourier Transform (DFT) values for spectrum analysis. `.FFT` uses internal time point values to calculate these values. A DFT uses sequences of time values to determine the frequency content of analog signals in circuit simulation. You can pass numerical parameters/expressions (but no string parameters) to the `.FFT` command. Output variables for `.FFT` can be voltage, current, or power, followed by a parenthesis containing the instance name. If it is power, for example, you need to write the signal's name in the format `p(`*`instance_name`*).

You can specify only one output variable in an `.FFT` command. The following is an *incorrect* use of the command because it contains two variables in one `.FFT` command:

```
.FFT v(1) v(2) np=1024
```

For an `.FFT` analysis using a Verilog A-block, the FFT time window is:
`TimeWindow = SamplePeriod*(`*`stopidx-startidx`*`)`

A FFT process requires sampling the waveform with equally spaced time points, and the total point number must be 2^N (N: integer). Therefore, the start/stop time points, fundamental frequency, sampling rate, and total point number are not independent of each other. They need to satisfy the following relationship:

$$\frac{point\_number}{t_{stop} - t_{start}} = sample\_rate, \text{ where } point\_number = 2^N$$

$$F_{fund} = \frac{M}{t_{stop} - t_{start}}, \text{ where } M \text{ is an integer number}$$

If that relationship is compromised, conflicts between parameters may arise. To avoid such conflicts, HSPICE conducts an error check process according to the following:

Parameter	Check if input....,	Adjust if input...	Set if not input...
START	Error if < tstart (start point in `.TRAN`)	N/A	=tstart (start point in `.TRAN`
STOP	Error if > tstop (stop point in `.TRAN`)	N/A	=tstop (stop point in `.TRAN`
NP	Error if NP< 4 Error if NP > 2^{27}	Is not a power of 2; adjust to nearest power of 2, issue warning and final value	Default value (1024)
FREQ	Error if < $\frac{1}{\mathrm{STOP} - START}$	If not integer multiple of 1/(STOP-START)), adjust to nearest multiple of 1/(STOP-START), issue warning and final value	$\frac{1}{\mathrm{STOP} - START}$
SamplePeriod	Error if non-positive	Use default value: 1s	1 second
StartIdx	Error if $\geq$ StopIdx	N/A	Start index of the VA array
StopIdx	Error if $\leq$ StartIdx	N/A	Stop index of the VA array

An embedded `.FFT` command in a measure_file can be called to perform FFT measurements from previous simulation results as follows:

```
HSPICE -i *.tr0 -meas measure_file
```

Example 1

```
.FFT v(1)
.FFT v(1,2) np=1024 start=0.3m stop=0.5m freq=5.0k
+ window=kaiser alfa=2.5
.FFT I(rload) start=0m to=2.0m fmin=100k fmax=120k
+ format=unorm
.FFT par('v(1) + v(2)') from=0.2u stop=1.2u
+ window=harris
```

Example 2

```
.FFT v(1) np=1024
.FFT v(2) np=1024
```

This example generates an *.ft0* file for the FFT of v(1) and an .ft1 file for the FFT of v(2).

See Also

.TRAN
.MEASURE FFT
Spectrum Analysis

.FLAT

Provides subcircuit OP back annotation when a device is modeled as a subckt.

Syntax

```
.FLAT device_name
```

Description

When a device is modeled as a subcircuit rather than as .MODEL, use of the .FLAT command within a subcircuit allows the writing of a results file with proper values for the device. Back-annotation is done by retrieving results from the *input.op0* (for DC) and *input.op1* (for transient) results files. Using the .FLAT command avoids the following problem:

In the subcircuit modeling of the device M0...

```
.subckt nmossub D G S B l=1u w=1u
M0 D int_G S B nmos4 l=l w=w
R1 int_G G 1K
.ends nmossub
```

...the *input.op#* file that is now produced has the following section for M0.

```
"X1^m0" "mosfet" (
2.38656e-08
-2.80717e-25
-3.17776e-11
```

When back-annotation is attempted once again, it fails since the annotation code is still looking for the results saved at the top level. In the example below, the M1 device within the subcircuit will have its results written as X1 and not hierarchically as X1^M1. The other devices will still have their results written hierarchically.

Example

```
.subckt nmossub D G S B l=l w=w
M1 D_int G_int S_int B nch l=l w=w
M2 D_int G_int S_int B nch l=l w=w
RD D D_int 100
RG G G_int 10
RS S S_int 400
.flat M1
.ends nmossub

X1 1 2 0 0 nmossub
```

.FOUR

Performs a Fourier analysis as part of the transient analysis.

Syntax

```
.FOUR freq ov1 [ov2 ov3 ...]
```

Arguments

Argument	Description
freq	Fundamental frequency.
ov1 ...	Output variables to analyze.

Description

Use this command to perform a Fourier analysis as part of the transient analysis. You can use this command in HSPICE to perform the Fourier analysis over the interval (tstop-fperiod, tstop), where:

- tstop is the final time, specified for the transient analysis.
- fperiod is a fundamental frequency period (`freq` parameter).

HSPICE performs Fourier analysis on 501 points of transient analysis data on the last 1/f time period, where f is the fundamental Fourier frequency. HSPICE interpolates transient data to fit on 501 points, running from (tstop-1/f) to tstop.

To calculate the phase, the normalized component and the Fourier component, HSPICE uses 10 frequency bins. The Fourier analysis determines the DC component and the first nine AC components. For improved accuracy, the `.FOUR` command can use non-linear, instead of linear interpolation.

You can use a `.FOUR` command only with a `.TRAN` command.

Example

```
.FOUR 100K V(5)
```

See Also

.TRAN
.FFT

.FSOPTIONS

Sets various options for the HSPICE Field Solver.

Syntax

```
.FSOPTIONS name [ACCURACY=LOW|MEDIUM|HIGH]
+ [GRIDFACTOR=val] [PRINTDATA=YES|NO]
+ [COMPUTE_GO=YES|NO] [COMPUTE_GD=YES|NO]
+ [COMPUTE_RO=YES|NO] [COMPUTE_RS=YES|NO]
+ [COMPUTE_RS=YES|NO|DIRECT|ITER]
+ [COMPUTE_TABLE=FREQENCY_SWEEP]
+ [SCALE_RS] [ROUGHNESS]
```

Arguments

Note:

The forms of the following arguments are interchangeable:

```
COMPUTE_GO : COMPUTEGO
COMPUTE_GD : COMPUTEGD
COMPUTE_RO : COMPUTERO
COMPUTE_RS : COMPUTERS
COMPUTE_TABLE : COMPUTETABLE
```

Argument	Description
name	Option name.
ACCURACY	Solver accuracy is one of the following: ▪ LOW ▪ MEDIUM ▪ HIGH
GRIDFACTOR	Multiplication factor (integer) to determine the final number of segments used to define the shape. If you set COMPUTE_RS=yes, the field solver does not use this parameter to compute Ro and Rs values.
PRINTDATA	Prints output matrixes to a file.
COMPUTE_GO	Computes the static conductance matrix.
COMPUTE_GD	Computes the dielectric loss matrix.

Argument	Description
COMPUTE_RO	Computes the DC resistance matrix.
COMPUTE_RS	Activates and chooses filament solver to compute R_o and R_s. The solver computes the skin-effect resistance matrix. ▪ YES: activate filament solver with direct matrix solver ▪ NO: (Default) Does not perform filament solver ▪ DIRECT: Activate filament solver with direct matrix solver (same as “YES”) ▪ ITER: Activates filament solver with iterative matrix solver
COMPUTE_TABLE	Specifies a type of frequency sweep for extracting RLGC Tabular Model. You can specify either LIN, DEC, OCT, POI. Specify the nsteps, start, and stop values using the following syntax for each type of sweep: ▪ `LIN` *`nsteps start stop`* ▪ `DEC` *`nsteps start stop`* ▪ `OCT` *`nsteps start stop`* ▪ `POI` *`nsteps freq_values`*
SCALE_RS	Scaling factor to the skin-effect (Rs) matrix.
ROUGHNESS	Value of the height of the RMS surface roughness.

Description

Use the `.FSOPTIONS` command to set various options for the field solver. The following rules apply to the field solver when specifying options with the `.FSOPTIONS` command:

- The field solver always computes the L and C matrixes.
- If `COMPUTE_RS=YES`, the field solver starts and calculates L_o, R_o, and R_s.
- For each accuracy mode, the field solver uses either the predefined number of segments or the number of segments that you specified. It then multiplies this number times the `GRIDFACTOR` to obtain the final number of segments.

Because a wide range of applications are available, the predefined accuracy level might not be accurate enough for some applications. If you need a higher accuracy than the value that the `HIGH` option sets, then increase either the `GRIDFACTOR` value or the `N`, `NH`, or `NW` values to increase the mesh density. `NW` and `NH` quantities are used for rectangles and `N` is used for circles, polygons

and strips. See the .SHAPE commands in this chapter for the complete syntax for each shape.

See the *HSPICE User Guide: Signal Integrity* for more information on Extracting Transmission Line Parameters (Field Solver).

Example

```
// LU solver
*.fsoptions printem printdata=yes compute_rs=direct
compute_gd=yes
// GMRES solver
.fsoptions printem printdata=yes compute_rs=iter compute_gd=yes
```

See Also

.LAYERSTACK
.MATERIAL
.SHAPE

.GLOBAL

Globally assigns a node name.

Syntax

```
.GLOBAL node1 node2 node3 ...
```

Arguments

Argument	Description
node1, node2...	Name of a global nodes, such as supply and clock names; overrides local subcircuit definitions.

Description

Use this command to globally assign a node name in HSPICE. This means that all references to a global node name, used at any level of the hierarchy in the circuit, connect to the same node.

The most common use of a `.GLOBAL` command is if your netlist file includes subcircuits. This command assigns a common node name to subcircuit nodes. Another common use of `.GLOBAL` commands is to assign power supply connections of all subcircuits. For example, `.GLOBAL VCC` connects all subcircuits with the internal node name VCC.

Typically, in a subcircuit, the node name consists of the circuit number concatenated to the node name. When you use a `.GLOBAL` command, HSPICE does not concatenate the node name with the circuit number and assigns only the global name. You can then exclude the power node name in the subcircuit or macro call.

Example

This example shows global definitions for `VDD` and `input_sig` nodes.

```
.GLOBAL VDD input_sig
```

.HB

Invokes the single and multitone harmonic balance algorithm for periodic steady state analysis.

Syntax

Syntax # 1 without SS_TONE

```
.HB TONES=F1 [F2 ... FN] [SUBHARMS=SH]
+ [NHARMS=H1, H2 ... HN] [INTMODMAX=n]
+ [SWEEP parameter_sweep]
```

Syntax#2 with SS_TONE

```
.HB TONES=F1 [F2 ... FN] [SUBHARMS=SH]
+ [NHARMS=H1, H2 ... HN] [INTMODMAX=n]
+ [SS_TONE=n] [SWEEP parameter_sweep]
```

Arguments

Argument	Description
TONES	Fundamental frequencies.
SUBHARMS	Subharmonics in the analysis spectrum. The minimum non-DC frequency in the analysis spectrum is f/subharms, where f is the frequency of oscillation.
NHARMS	Number of harmonics to use for each tone. Must have the same number of entries as TONES. You must specify NHARMS, INTMODMAX or both.
INTMODMAX	Maximum intermodulation product order that you can specify in the analysis spectrum. You must specify NHARMS, INTMODMAX or both.

Argument	Description
SWEEP	Type of sweep. You can sweep up to three variables. You can specify either LIN, DEC, OCT, POI, SWEEPBLOCK, DATA, OPTIMIZE or MONTE. Specify the nsteps, start, and stop times using the following syntax for each type of sweep: ▪ LIN *nsteps start stop* ▪ DEC *nsteps start stop* ▪ OCT *nsteps start stop* ▪ POI *nsteps freq_values* ▪ SWEEPBLOCK *nsteps freq1 freq2 ... freqn* ▪ DATA=*dataname* ▪ OPTIMIZE=OPT*xxx* ▪ MONTE=*val*
SS_TONE	Small-signal tone number for HBLIN analysis. The value must be an integer number. The default value is 0, indicating that no small signal tone is specified.

Description

Use this command to invoke the single and multitone harmonic balance algorithm for periodic steady state analysis.

The `NHARMS` and `INTMODMAX` input parameters define the spectrum.

- If `INTMODMAX=N`, the spectrum consists of all f=a*f_1 + b*f_2 + ... + n*fn frequencies so that f>=0 and |a|+|b|+...+|n|<=N. The a,b,...,n coefficients are integers with absolute value <=N.
- If `INTMODMAX` is not specified, HSPICE RF defaults it to the largest value in the `NHARMS` list.
- If entries in the `NHARMS` list are > `INTMODMAX`, HSPICE RF adds the m*f_k frequencies to the spectrum, where f_k is the corresponding tone, and m is a value <= the `NHARMS` entry.

For detailed discussion of HBLIN analysis, see Frequency Translation S-Parameter (HBLIN) Extraction in the *HSPICE User Guide: RF Analysis*.

Example 1

The resulting HB analysis spectrum={dc, f_1, f_2}.

```
.hb tones=f1, f2 intmodmax=1
```

Example 2

The HB analysis spectrum={dc, f_1, f_2, f_1+f_2, f_1-f_2, 2*f_1, 2*f_2}.

```
.hb tones=f1, f2 intmodmax=2
```

Example 3

The resulting HB analysis spectrum={dc, f_1, f_2, f_1+f_2, f_1-f_2, 2*f_1, 2*f_2, 2*f_1+f_2, 2*f_1-f_2, 2*f_2+f_1, 2*f_2-f_1, 3*f_1, 3*f_2}.

```
.hb tones=f1, f2 intmodmax=3
```

Example 4

The resulting HB analysis spectrum={dc, f_1, f_2, f_1+f_2, f_1-f_2, 2*f_1,2*f_2}.

```
.hb tones=f1, f2 nharms=2,2
```

Example 5

The resulting HB analysis spectrum={dc, f_1, f_2, f_1+f_2, f_1-f_2, 2*f_1, 2*f_2, 2*f_1-f_2, 2*f_1+f_2, 2*f_2-f_1, 2*f_2+f_1}.

```
hb tones=f1, f2 nharms=2,2 intmodmax=3
```

Example 6

The resulting HB analysis spectrum={dc, f_1, f_2, f_1+f_2, f_1-f_2, 2*f_1, 2*f_2, 2*f_1-f_2, 2*f_1+f_2, 2*f_2-f_1, 2*f_2+f_1, 3*f_1, 3*f_2, 4*f_1, 4*f_2, 5*f_1, 5*f_2}.

```
.hb tones=f1, f2 nharms=5,5 intmodmax=3
```

See Also

.ENV
.HBAC
.HBLIN
.HBNOISE
.HBOSC
.OPTION HBCONTINUE
.OPTION HBJREUSE
.OPTION HBJREUSETOL
.OPTION HBACKRYLOVDIM
.OPTION HBKRYLOVTOL
.OPTION HBLINESEARCHFAC
.OPTION HBMAXITER
.OPTION HBSOLVER
.OPTION HBTOL
.OPTION LOADHB
.OPTION SAVEHB
.OPTION TRANFORHB

.PRINT
.PROBE

.HBAC

Performs harmonic-balance–based periodic AC analysis on circuits operating in a large-signal periodic steady state.

Syntax

```
.HBAC frequency_sweep
```

Arguments

Argument	Description
frequency_sweep	Frequency sweep range for the input signal (also refer to as the input frequency band (IFB) or fin). You can specify LIN, DEC, OCT, POI, SWEEPBLOCK, or DATA. Specify the nsteps, start and stop times using the following syntax for each type of sweep: ■ LIN *nsteps start stop* ■ DEC *nsteps start stop* ■ OCT *nsteps start stop* ■ POI *nsteps freq_values* ■ SWEEPBLOCK *nsteps freq1 freq2 ... freqn* ■ DATA=*dataname*

Description

Use this command to invoke a harmonic balance-based periodic AC analysis to analyze small-signal perturbations on circuits operating in a large-signal periodic steady state.

See Also

.HB
.HBNOISE
.HBOSC
.OPTION HBACTOL
.OPTION HBACKRYLOVDIM
.PRINT
.PROBE

.HBLIN

Extracts frequency translation S-parameters and noise figures.

Syntax

Without `SS_TONE`

```
.HBLIN frequency_sweep
+ [NOISECALC=1|0|yes|no] [FILENAME=file_name]
+ [DATAFORMAT=ri|ma|db]
+ [MIXEDMODE2PORT=dd|cc|cd|dc|sd|sc|cs|ds]
```

With `SS_TONE`

```
.HBLIN [NOISECALC=1|0|yes|no] [FILENAME=file_name]
+ [DATAFORMAT=ri|ma|db]
+ [MIXEDMODE2PORT=dd|cc|cd|dc|sd|sc|cs|ds]
```

Arguments

Parameter	Description
frequency_sweep	Frequency sweep range for the input signal (also referred to as the input frequency band (IFB) or fin). You can specify LIN, DEC, OCT, POI, SWEEPBLOCK, or DATA. Specify the nsteps, start, and stop times using the following syntax for each type of sweep: ▪ LIN *nsteps start stop* ▪ DEC *nsteps start stop* ▪ OCT *nsteps start stop* ▪ POI *nsteps freq_values* ▪ SWEEPBLOCK *nsteps freq1 freq2 ... freqn* ▪ DATA=*dataname*
NOISECALC	Enables calculating the noise figure. The default is no (0).
FILENAME	Output file name for the extracted S-parameters or the object name after the -o command-line option. The default is the netlist file name.

Parameter	Description
DATAFORMAT	Format of the output data file. ■ dataformat=RI, real-imaginary. This is the default for .sc#/citi file. ■ dataformat=MA, magnitude-phase. This is the default format for Touchstone file. ■ dataformat=DB, DB(magnitude)-phase.
MIXEDMODE2PORT	Mixed-mode data map of output mixed mode S-parameter matrix. The availability and default value for this keyword depends on the first two port (P element) configuration as follows: ■ case 1: p1=p2=single-ended (standard-mode P element) available: ss default: ss ■ case 2: p1=p2=balanced (mixed-mode P element) available: dd, cd, dc, cc default: dd ■ case 3: p1=balanced p2=single-ended available: ds, cs default: ds ■ case 4: p1=single p2=balanced available: sd, sc default: sd

Description

Use this command in HSPICE RF to extract frequency translation S-parameters and noise figures.

See Also

.HB
.HBAC
.PRINT
.PROBE

.HBLSP

Performs periodically driven nonlinear circuit analyses for power-dependent S parameters.

Syntax

```
.HBLSP NHARMS=nh [POWERUNIT=dbm|watt]
+ [SSPCALC=1|0|YES|NO] [NOISECALC=1|0|YES|NO]
+ [FILENAME=file_name] [DATAFORMAT=ri|ma|db]
+ FREQSWEEP freq_sweep POWERSWEEP power_sweep
```

Arguments

Parameter	Description
NHARMS	Number of harmonics in the HB analysis triggered by the `.HBLSP` command.
POWERUNIT	Power unit. Default is watt.
SSPCALC	Extract small-signal S-parameters. Default is 0 (NO).
NOISECALC	Perform small-signal 2-port noise analysis. Default is 0 (NO).
FILENAME	Output data *.p2d#* filename. Default is the netlist name or the object name after the -o command-line option.
DATAFORMAT	Format of the output data file. Default is ma (magnitude, angle).
FREQSWEEP	Frequency sweep specification. A sweep of type LIN, DEC, OCT, POI, or SWEEPBLOCK. Specify the *nsteps*, *start*, and *stop* times using the following syntax for each type of sweep: ■ LIN *nsteps start stop* ■ DEC *nsteps start stop* ■ OCT *nsteps start stop* ■ POI *nsteps freq_values* ■ SWEEPBLOCK=*blockname* This keyword must appear before the POWERSWEEP keyword.

Parameter	Description
POWERSWEEP	Power sweep specification. A sweep of type LIN, DEC, OCT,POI, or SWEEPBLOCK. Specify the nsteps, start, and stop times using the following syntax for each type of sweep: ■ LIN *nsteps start stop* ■ DEC *nsteps start stop* ■ OCT *nsteps start stop* ■ POI *nsteps power_values* ■ SWEEPBLOCK=*blockname* This keyword must follow the FREQSWEEP keyword.

Description

Use this command in HSPICE RF to invoke periodically driven nonlinear circuit analyses for power-dependent S-parameters.

For details, see the *HSPICE User Guide: RF Analysis*, Large-Signal S-parameter (HBLSP) Analysis.

See Also

.HB
.PRINT
.PROBE

.HBNOISE

Performs cyclo-stationary noise analysis on circuits operating in a large-signal periodic steady state.

Syntax

```
.HBNOISE [output] [insrc] [parameter_sweep]
+ [n1, n2, ..., nk,+/-1]
+ [listfreq=(frequencies|none|all)] [listcount=val]
+ [listfloor=val] [listsources=on|off]
```

Arguments

Parameter	Description
output	Output node, pair of nodes, or 2-terminal element. HSPICE RF references equivalent noise output to this node (or pair of nodes). Specify a pair of nodes as V(n+,n-). If you specify only one node, V(n+), then HSPICE RF assumes that the second node is ground. You can also specify a 2-terminal element name that refers to an existing element in the netlist.
insrc	Input source. If this is a resistor, HSPICE RF uses it as a reference noise source to determine the noise figure. If the resistance value is 0, the result is an infinite noise figure.
parameter_sweep	Frequency sweep range for the input signal. Also referred to as the input frequency band (IFB) or *fin*). You can specify LIN, DEC, OCT, POI, SWEEPBLOCK, DATA, MONTE, or OPTIMIZE sweeps. Specify the nsteps, start, and stop frequencies using the following syntax for each type of sweep: ▪ LIN *nsteps start stop* ▪ DEC *nsteps start stop* ▪ OCT *nsteps start stop* ▪ POI *nsteps freq_values* ▪ SWEEPBLOCK *nsteps freq1 freq2 ... freqn* ▪ DATA *dataname* ▪ MONTE *niterations* ▪ OPTIMIZE *optxxx*

Parameter	Description
n1,n2,...,nk, +/-1	Index term defining the output frequency band (OFB or *fout*) at which the noise is evaluated. Generally, fout=ABS(n1*f+n2*f2+...+nk*fk+/-fin) Where: ▪ f1,f2,...,fk are the first through k-th steady-state tones determined from the harmonic balance solution ▪ n1,n2,...,nk are the associated harmonic multipliers ▪ fin is the IFB defined by *parameter_sweep*. The default index term is [1,1,...1,-1]. For a single tone analysis, the default mode is consistent with simulating a low-side, down conversion mixer where the RF signal is specified by the IFB and the noise is measured at a down-converted frequency that the OFB specifies. In general, you can use the [n1,n2,...,nk,+/-1] index term to specify an arbitrary offset. The noise figure measurement is also dependent on this index term.

listfreq	Prints the element noise value to the .lis file. You can specify at which frequencies the element noise value is printed. The frequencies must match the sweep_frequency values defined in the *parameter_sweep*, otherwise they are ignored.
	In the element noise output, the elements that contribute the largest noise are printed first. The frequency values can be specified with the NONE or ALL keyword, which either prints no frequencies or every frequency defined in *parameter_sweep*. Frequency values must be enclosed in parentheses. For example:`listfreq=(none)`
	`listfreq=(all)`
	`listfreq=(1.0G)`
	`listfreq=(1.0G, 2.0G)` The default value is NONE.

listcount	Prints the element noise value to the .lis file, which is sorted from the largest to smallest value. You do not need to print every noise element; instead, you can define `listcount` to print the number of element noise frequencies. For example, `listcount=5` means that only the top 5 noise contributors are printed. The default value is 1.
listfloor	Prints the element noise value to the *.lis* file and defines a minimum meaningful noise value (in $V/Hz^{1/2}$ units). Only those elements with noise values larger than `listfloor` are printed. The default value is 1.0e-14 $V/Hz^{1/2}$.

Parameter	Description
listsources	Prints the element noise value to the *.lis* file when the element has multiple noise sources, such as a FET, which contains the thermal, shot, and 1/f noise sources. You can specify either ON or OFF: ON Prints the contribution from each noise source and OFF does not. The default value is OFF.

Description

Use this command to invoke cyclo-stationary noise analysis on circuits operating in a large-signal periodic steady state.

See Also

.HB
.HBAC
.HBOSC
.PRINT
.PROBE

.HBOSC

Performs oscillator analysis on autonomous (oscillator) circuits. The input syntax for HBOSC analysis supports two different formats, depending on whether the PROBENODE location is specified using a circuit element (current source) or using the HBOSC PROBENODE parameters:

Syntax

Syntax #1

```
.HBOSC TONE=F1 NHARMS=H1
+ PROBENODE=N1,N2,VP
+[FSPTS=NUM, MIN, MAX] [STABILITY=(-2|-1|0|1|2)]
+[SWEEP PARAMETER_SWEEP] [SUBHARMS=I]
```

Syntax #2 (Uses current source to set PROBENODE)

```
ISRC N1 N2 HBOSCVPROBE=VP
.HBOSC TONE=F1 NHARMS=H1
+[FSPTS=NUM, MIN, MAX] [STABILITY=(-2|-1|0|1|2)]
+[SWEEP PARAMETER_SWEEP] [SUBHARMS=I]
```

Arguments

Parameter	Description
TONE	Approximate value for oscillation frequency (Hz). The search for an exact oscillation frequency begins from this value unless you specify an FSPTS range or transient initialization.
NHARMS	Number of harmonics to use for oscillator HB analysis.
PROBENODE	Circuit nodes that are probed for oscillation conditions. ▪ N1 and N2 are the positive and negative nodes for a voltage probe inserted in the circuit to search for oscillation conditions. ▪ VP is the initial probe voltage value (suggested: 1/2 the supply voltage). The phase of the probe voltage is forced to zero; all other phases are relative to the probe phase. HSPICE RF uses this probe to calculate small-signal admittance for the initial frequency estimates. It should be connected near the “heart” of the oscillator (near resonators, inside the ring of a ring oscillator, and so on). Note: The PROBENODE pins and approximate voltage value can also be set by using a zero amp current source that uses the HBOSCVPROBE keyword.

Parameter	Description
HBOSCVPROBE=VP	Sets PROBENODE with a current source. If a current source with HBOSCVPROBE is used, the PROBENODE syntax is not necessary.
FSPTS	Frequency search points that HSPICE RF uses in its initial small-signal frequency search to find an oscillation frequency. Optional, but recommended for high-Q and most LC oscillators. ▪ NUM is an integer. ▪ MIN and MAX are frequency values in units of Hz. If the FSPTS analysis finds an approximate oscillation frequency, the TONE parameter is ignored. An option for FSPTS

Parameter	Description
STABILITY	When used with FSPTS, activates the additional oscillator stability analyses depending on the following values: ■ 0: A single point oscillator frequency-search stability analysis is performed. The FSPTS search is executed, and the first successful linear oscillation frequency value found is used as the starting point for the two-tier Newton nonlinear oscillator analysis. The probenode vp value specified is used as the starting amplitude for the Newton solver. ■ 1: (default) A single point oscillator frequency-search stability analysis, plus an estimate of oscillator amplitude, is performed. The FSPTS search is executed, and the first successful linear oscillation frequency value found is used as the starting point for the two-tier Newton nonlinear oscillator analysis. An additional analysis for automatically estimating the probenode amplitude is also performed, and this value is used as the starting amplitude for the two-tier Newton solver. ■ –1: A single point oscillator frequency-search stability analysis, plus an estimate of oscillator amplitude, is performed. The FSPTS search is executed, and the first successful linear oscillation frequency value found is accurately computed and reported. An additional analysis for automatically estimating the probenode amplitude is also performed, and this value is also reported. The analysis aborts without attempting the two-tier Newton nonlinear oscillator analysis. By using STABILITY=–1, a check can be made if any linear oscillation conditions are found, before attempting the nonlinear oscillator analysis. ■ 2: A multipoint frequency-search stability analysis is performed. The FSPTS search is executed, and all successful linear oscillation frequency values found over the entire FSPTS search range are reported. For each potential oscillation frequency found, an additional analysis for estimating the probenode amplitude is also performed. All frequency and amplitude values are reported. The frequency value that has the largest predicted amplitude is used as the starting point for the two-tier Newton nonlinear oscillator analysis. ■ –2: A multipoint frequency-search stability analysis is performed. The FSPTS search is executed, and all successful linear oscillation frequency values found over the entire FSPTS search range are reported. For each potential oscillation frequency found, an additional analysis for estimating the probenode amplitude is also performed. All frequency and amplitude values are reported. The analysis aborts without attempting the two-tier Newton nonlinear oscillator analysis. By using STABILITY=–2, a check can be made if any linear oscillation conditions are found, before attempting the nonlinear oscillator analysis.

Parameter	Description
SWEEP	Type of sweep. You can sweep up to three variables. You can specify either LIN, DEC, OCT, POI, SWEEPBLOCK, DATA, OPTIMIZE, or MONTE. Specify the nsteps, start, and stop frequencies using the following syntax for each type of sweep: ■ LIN *nsteps start stop* ■ DEC *nsteps start stop* ■ OCT *nsteps start stop* ■ POI *nsteps freq_values* ■ SWEEPBLOCK *nsteps freq1 freq2 ... freqn* ■ DATA=*dataname* ■ OPTIMIZE=OPT*xxx* ■ MONTE=*val*
SUBHARMS	Subharmonics in the analysis spectrum. The minimum non-DC frequency in the analysis spectrum is f/subharms, where f is the frequency of oscillation. Use this option if your oscillator circuit includes a divider or prescaler that result in frequency terms that are subharmonics of the fundamental oscillation frequency

Description

Use this command to invoke oscillator analysis on autonomous (oscillator) circuits.

Example 1

This example performs an oscillator analysis, searching for frequencies in the vicinity of 900 MHz. This example uses nine harmonics with the probe inserted between the `gate` and `gnd` nodes. The probe voltage estimate is `0.65` V.

```
.HBOSC tone=900MEG nharms=9 probenode=gate,gnd,0.65
```

Example 2

This example performs an oscillator analysis, searching for frequencies in the vicinity of 2.4 GHz. This example uses `11` harmonics with the probe inserted between the `drainP` and `drainN` nodes. The probe voltage estimate is `1.0` V.

```
.HBOSC tone=2400MEG nharms=11
+ probenode=drainP,drainN,1.0 fspts=20,2100MEG,2700MEG
```

Example 3

Another means to define the probenode information is through a zero-current source. The following two methods define an equivalent `.HBOSC` command:

- Method 1:

```
.HBOSC tone = 2.4G nharms = 10
+ probenode = drainP, drainN, 1.0
+ fspts = 20, 2.1G, 2.7G
```

- Method 2:

```
ISRC drainP drainN 0 HBOSCVPROBE = 1.0
.HBOSC tone = 2.4G nharms = 10
+ fspts = 20, 2.1G, 2.7G
```

In method 2, the PROBENODE information is defined by a current source in the circuit. Only one such current source is needed, and its current must be 0.0 with the HBOSC PROBENODE voltage defined through its HBOSCVPROBE property.

See Also

.HB
.OPTION HBFREQABSTOL
.OPTION HBFREQRELTOL
.OPTION HBMAXOSCITER
.OPTION HBPROBETOL
.OPTION HBTRANFREQSEARCH
.OPTION HBTRANINIT
.OPTION HBTRANPTS
.OPTION HBTRANSTEP
.PRINT
.PROBE

.HBXF

Calculates transfer from the given source in the circuit to the designated output.

Syntax

`.HBXF` *`out_var freq_sweep`*

Arguments

Parameter	Description
out_var	Specify `i(2_`*`port_elem`*`)` or `V(`*`n1`*`<,`*`n2`*`>)`
freq_sweep	A sweep of type LIN, DEC, OCT, POI, or SWEEPBLOCK. Specify nsteps, start/stop times the syntax below for each type of sweep: ▪ LIN *nsteps start stop* ▪ DEC *nsteps start stop* ▪ OCT *nsteps start stop* ▪ POI *nsteps freq_values* ▪ SWEEPBLOCK = *BlockName* Specify the frequency sweep range for the output signal. HSPICE RF determines the offset frequency in the input sidebands; for example, f1 = abs(fout - k*f0) s.t. f1<=f0/2 The f0 is the steady-state fundamental tone and f1 is the input frequency.

Description

Calculates the transfer function from the given source in the circuit to the designated output.

Example

Here, trans-impedance from `isrc` to `v(1)` is calculated based on HB analysis.

```
.hb tones=1e9 nharms=4
.hbxf  v(1) lin 10 1e8 1.2e8
.print hbxf tfv(isrc)  tfi(n3)
```

See Also

.HB
.HBAC
.HBNOISE
.HBOSC
.SNXF

.HDL

Specifies the Verilog-A source name and path.

Syntax

```
.HDL "file_name" [module_name] [module_alias]
```

Arguments

Argument	Description
file_name	Verilog-A or CML file.
module_name	Optional module name. If a module is specified, then only that module is loaded from the specified Verilog-A or CML file. If the module is not found or if the module specification is not uniquely case-insensitive inside, then an error is generated. (HSPICE only).
module_alias	If specified (in addition to a module name), then that module is loaded into the system using the alias in place of the module name defined in the Verilog-A source file. Thereafter, any reference to the module is made using its alias. The system behaves as if the module had the alias as its module name. A module might be loaded with any number of aliases in addition to being loaded without an alias. This argument is useful when loading modules of the same name from different files. See Example 4 below. (HSPICE only)

Description

Use .HDL commands to specify the Verilog-A or compiled model library (CML) source name and path within a netlist. The Verilog-A file is assumed to have a **.va* extension only when a prefix is provided. You can also use .HDL commands in .ALTER blocks to vary simulation behavior. For example, to compare multiple variations of Verilog-A modules.

In .MODEL commands you must add the Verilog-A type of model cards. Every Verilog-A module can have one or more associated model cards. The type of model cards should be the same as the Verilog-A module name. Verilog-A module names cannot conflict with HSPICE built-in device keywords. If a conflict occurs, HSPICE issues a warning message and the Verilog-A module definition is ignored.

The module_name and module_alias arguments can be specified without quotes or with single or double quotes. Any tokens after the module alias are ignored.

The same Verilog-A case insensitivity rules used for module and parameter names apply to both the module_name and module_alias arguments, and the same module override logic applies.

File Loading Considerations

These restrictions and issues must be considered when loading Verilog-A modules:

- All Verilog-A modules are loaded into the system prior to any device instantiation. You can place an .HDL command anywhere in the top-level circuit.
- An .HDL command is illegal inside a .SUBCKT or IF-ELSEIF-ELSE block; otherwise, the simulation exits with an error message.
- When a module to be loaded has the same name as a previously-loaded module or the names differ in case only, the latter one is ignored and the simulator issues a warning message.
- If a Verilog-A module file is not found or the CML file has an incompatible version, the simulation exits and an error message is issued.

Example 1

```
.HDL "/myhome/Verilog_A_lib/res.va"
```

This example loads the res.va Verilog-A model file from the directory /myhome/Verilog_A_lib.

Example 2

```
.HDL "va_models"
```

This example loads the va_models.va Verilog-A model file (not va_model file) from the current working directory.

Example 3

```
* simple .alter test
.hdl amp_one.va
v1 1 0 10
x1 1 0 va_amp
.tran 10n 100n
.alter alter1
.hdl amp_two.va
.end
```

This example loads the module called va_amp from the amp_one.va file for the first simulation run. For the second run, HSPICE loads the va_amp module from the amp_two.va file.

Example 4

The module_alias argument is useful when loading modules of the same name from different files. For example, if you have a module res in two libraries, such as 'fast.va' and 'slow.va', then you can write,

```
.hdl 'fast.va' 'res' 'fast_res'
.hdl 'slow.va' 'res' 'slow_res'
...
x1 1 2 fast_res r=1
x2 2 0 slow_res r =1
...
```

See Also

.ALTER
.MODEL

.IBIS

Provides IBIS functionality by specifying an IBIS file and component and optional keywords.

Syntax

```
.IBIS 'ibis_name'
+ file='ibis_file_name'
+ component='component_name' [time_control=0|1]
+ [mod_sel='sel1=mod1,sel2=mod2,...']
+ [package=0|1|2|3] [pkgfile='pkg_file_name']
+ [typ={typ|min|max}]
+ [nowarn]
+ ...
```

Arguments

Keyword	Description
ibis_name	Instance name of this ibis command.
file	Name of ibis (**ibs*) file.
component or cname	Component name.
time_control	Invokes an HSPICE time-control algorithm to achieve greater accuracy for high speed digital signal buffers: ■ 0 (default): Time step algorithm will not take effect. ■ 1: Launches the time-step algorithm.
mod_sel	Assigns special model for model selector, here model selector can be used for series model. If model selector is used for a pin of a component, but `mod_sel` is not set in the `.ibis` command, then the first model under the corresponding [Model Selector] will be selected as default.

Keyword	Description
package	When package equals: ■ 0, then the package is not added into the component. ■ 1, then RLC of [Package] (in the *.ibs* file) is added. ■ 2, then RLC of [Pin] (in the *.ibs* file) is added. ■ 3 (default), and if [Package Model] is defined, set package with a package model. If the [Package Model] is not defined, set the package with [Pin]. If the package information is not set in [Pin], set the package with [Package] as a default. You can define the [Package Model] in an IBIS file specified by the *file* keyword or a PKG file specified by the ***pkgfile*** keyword. The pkgfile keyword is useful only when package =3
typ	The value of the typ signifies a column in the *IBIS* file from which the current simulation extracts data. The default is typ=typ. If min or max data are not available, typ data are used instead.
nowarn	The nowarn keyword suppresses warning messages from the IBIS parser.

Note:

There are many other option keywords which are the same as for the B-element (I/O buffer). They are: typ, interpo, ramp_rwf, ramp_fwf, rwf_tune, fwf_tune, pd_scal, pu_scal, pc_scal, gc_scal, rwf_scal, fwf_scal, hsp_ver, c_com_pd, c_com_pu, c_com_pc, c_com_gc. For details, see Specifying Common Keywords in Chapter 4 of the *HSPICE User Guide*: *Signal Integrity*. If such keywords are set, they are applied on all buffers of the component.

Description

The general syntax above shows the `.IBIS` command when used with a component. The optional keywords are in square brackets.

Example

```
.ibis cmpnt
+ file = 'ebd.ibs'
+ component = 'SIMM'
+ hsp_ver=2002.4 nowarn package=2
```

This example corresponds to the following *ebd.ibs* file:

```
[Component]   SIMM
[Manufacturer]   TEST
[Package]
R_pkg    200m    NA   NA
L_pkg    7.0nH    NA    NA
C_pkg    1.5pF    NA    NA
|
[Pin]   signal_name   model_name   R_pin   L_pin   C_pin
|
1    ND1    ECL    40.0m   2n   0.4p
2    ND2    NMOS    50.0m   3n   0.5p
.................
```

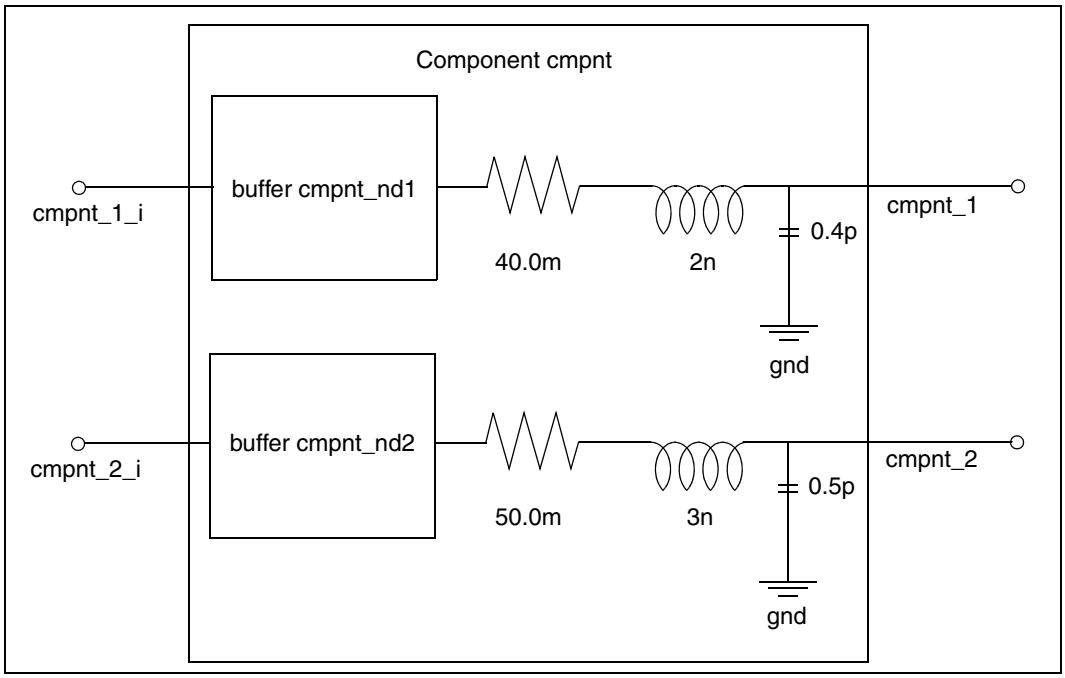

Figure 8 Equivalent Circuit for IBIS Component Example

Example

```
.IBIS cmpt1
+ file='example.ibs'
+ component='EXAMPLE'
+ mod_sel = 'DQ = DQ_FULL'
```

In the following example, the model DQ_FULL will be used for all pins that use the model name DQ.The corresponding IBIS file, *example.ibs*, contains the following [Model Selector] section:

```
|**********************MODEL SELECTOR************************
|
[Model Selector] DQ
|
DQ_FULL              Full-Strength IO Driver
DQ_HALF              54% Reduced Drive Strength IO Driver
*
```

See Also

.EBD
.PKG

.IC

Sets transient initial conditions in HSPICE.

Syntax

```
.IC V(node1)=val1 V(node2)=val2 ...
```

Arguments

Argument	Description
val1 ...	Specifies voltages. The significance of these voltages depends on whether you specify the UIC parameter in the .TRAN command.
node1 ...	Node numbers or names can include full paths or circuit numbers.

Description

Use the `.IC` command or the `.DCVOLT` command to set transient initial conditions in HSPICE. How it initializes depends on whether the `.TRAN` analysis command includes the UIC parameter. This command is less preferred compared to using the `.NODESET` command in many cases.

When using the `.IC` command, forcing circuits are connected to the `.IC` nodes for the duration of DC convergence. After DC convergence is obtained, the forcing circuits are removed for all further analysis. The DC operating point for each .IC'd node should be very close to the voltage specified in the `.IC` command. If a node is not, then that node has a DC conductance to ground comparable to GMAX. This is almost certainly an error condition. In the rare case that it is not, GMAX can be increased to prevent appreciable current division. Example: `.OPTION GMAX=1000`

Note:

In nearly all applications, `.NODESET` should be used to ensure a true DC operating point is obtained. Intentionally floating (or very high impedance) nodes should be set to a known good voltage using `.IC`.

If you do not specify the UIC parameter in the .TRAN command, HSPICE computes the DC operating point solution before the transient analysis. The node voltages that you specify in the `.IC` command are fixed to determine the DC operating point. They are used only in the first iteration to set an initial guess for the DC operating point analysis. The `.IC` command is equivalent to specifying the IC parameter on each element command, but is more convenient.

If you specify the UIC parameter in the .TRAN command, HSPICE does not calculate the initial DC operating point, but directly enters transient analysis. When you use .TRAN UIC, the .TRAN node values (at time zero) are determined by searching for the first value found in this order: from .IC value, then IC parameter on an element command, then .NODESET value, otherwise use a voltage of zero.

Note that forcing a node value of the dc operating point may not satisfy KVL and KCL. In this event you may likely see activity during the initial part of the simulation.This may happen if UIC is used and some node values left unspecified, when too many (conflicting) .IC values are specified, or when node values are forced and the topology changes. In this event you may likely see activity during the initial part of the simulation. Forcing a node voltage applies a fixed equivalent voltage source during DC analysis and transient analysis removes the voltage sources to calculate the second and later time points.

Therefore, to correct DC convergence problems use .NODESETs (without .TRAN UIC) liberally (when a good guess can be provided) and use .ICs sparingly (when the exact node voltage is known).

In addition, you can use wildcards in the .IC command. See Using Wildcards on Node Names in the *HSPICE User Guide: Simulation and Analysis*.

Example

```
.IC V(11)=5 V(4)=-5 V(2)=2.2
```

See Also

.DCVOLT
.TRAN
.OPTION DCIC
.NODESET
.OPTION GMAX

.ICM

Automatically creates port names that reference the pin name of an ICM model and generate a series of element nodes on the pin.

Syntax

```
.ICM icmname
+ file='icmfilename'
+ model='icmmodelname'
```

Arguments

Argument	Description
icmname	`.ICM` command card name.
icmfilename	Name of an *.*icm* file that contains an ICM model.
icmmodelname	Working model in an *.icm* file.
nodemapname	Name of the [ICM node map] keyword in an .icm file.
pinmapname	Name of the [ICM pin map] keyword in an .icm file.
pinname	Name of the first column of entries of the [ICM node map] or [ICM pin map].
sidename	Name of the side subparameter

Description

Use this command to automatically create port names that reference the pin name of an ICM model and generate a series of element (W/S/RLGCK) nodes on the pin when one of the following conditions occur:

- If the model is described using [Nodal Path Description] ''icmname'_'nodemapname'_'sidename'_'pinname'
- If the model is described using [Tree Path Description] 'icmname'_'pinmapname'_'sidename'_'pinname'

Note:

If a side subparameter is not used in an ICM file, then 'sidename'_ (above) should be removed.

Example 1

```
.ICM icm1
+ file='test1.icm'
+ model='FourLineModel1'
```

Example 2

The following example shows how to reference a pin of the ICM model in a HSPICE netlist.

```
icm1_NodeMap1_SideName1_pin1, icm1_NodeMap2_SideName2_pin1,
icm1_NodeMap2_SideName2_pin2, ...
```

.IF

Specifies conditions that determine whether HSPICE executes subsequent commands in conditional block.

Syntax

```
.IF (condition1)...
+ [.ELSEIF (condition2)... ]
+ [.ELSE ... ]
.ENDIF
```

Arguments

Argument	Description
condition1	Condition that must be true before HSPICE executes the commands that follow the .IF command.
condition2	Condition that must be true before HSPICE executes the commands that follow the .ELSEIF command. HSPICE executes the commands that follow *condition2* only if *condition1* is false and *condition2* is true.

Description

HSPICE executes the commands that follow the first .ELSEIF command only if *condition1* in the preceding .IF command is false and *condition2* in the first .ELSEIF command is true.

If *condition1* in the .IF command and *condition2* in the first .ELSEIF command are both false, then HSPICE moves on to the next .ELSEIF command if there is one. If this second .ELSEIF condition is true, HSPICE executes the commands that follow the second .ELSEIF command, instead of the commands after the first .ELSEIF command.

HSPICE ignores the commands in all false .IF and .ELSEIF commands, until it reaches the first .ELSEIF condition that is true. If no .IF or .ELSEIF condition is true, HSPICE continues to the .ELSE command.

.ELSE precedes one or more commands in a conditional block after the last .ELSEIF command, but before the .ENDIF command. HSPICE executes these commands by default if the conditions in the preceding .IF command and in all of the preceding .ELSEIF commands in the same conditional block all false.

The .ENDIF command ends a conditional block of commands that begins with an .IF command.

Example

```
.IF (a==b)
.INCLUDE /myhome/subcircuits/diode_circuit1
...
.ELSEIF (a==c)
.INCLUDE /myhome/subcircuits/diode_circuit2
...
.ELSE
.INCLUDE /myhome/subcircuits/diode_circuit3
...
.ENDIF
```

See Also

.ELSE
.ELSEIF
.ENDIF

.INCLUDE

Includes another netlist as a subcircuit of the current netlist.

Syntax

```
.INCLUDE 'file_path file_name'
```

Arguments

Argument	Description
file_path	Path name of a file for computer operating systems that support tree-structured directories. An include file can contain nested .INCLUDE calls to itself or to another include file. If you use a relative path in a nested .INCLUDE call, the path starts from the directory of the parent .INCLUDE file, not from the current working directory. If the path starts from the current working directory, HSPICE can also find the .INCLUDE file, but prints a warning.
file_name	Name of a file to include in the data file. The file path, plus the file name, can be up to 16 characters long. You can use any valid file name for the computer's operating system.

Description

Use this command to include another netlist in the current netlist. You can include a netlist as a subcircuit in one or more other netlists. You must enclose the file path and file name in single or double quotation marks. Otherwise, an error message is generated.

Example

```
.INCLUDE `/myhome/subcircuits/diode_circuit´
```

.LAYERSTACK

Defines a stack of dielectric or metal layers.

Syntax

```
.LAYERSTACK sname [BACKGROUND=mname]
+ [LAYER=(mname,thickness) ...]
```

Arguments

Argument	Description
sname	Layer stack name.
mname	Material name.
BACKGROUND	Background dielectric material name. By default, the field solver assumes AIR for the background.
thickness	Layer thickness.

Description

Use this command to define a stack of dielectric or metal layers. You must associate each transmission line system with *only* one layer stack. However, you can associate a single-layer stack with many transmission line systems.

In the layer stack:

- Layers are listed from bottom to top.
- Metal layers (ground planes) can be located only at the bottom, only at the top, or both at the top and bottom.
- Layers are stacked in the y-direction; the bottom of a layer stack is at y=0.
- All conductors must be located above y=0.
- Background material must be dielectric.

The following limiting cases apply to the `.LAYERSTACK` command:

- Free space without ground:

  ```
  .LAYERSTACK mystack
  ```

- Free space with a (bottom) ground plane where a predefined metal name = perfect electrical conductor (PEC):

  ```
  .LAYERSTACK halfSpace PEC 0.1mm
  ```

See Also

.FSOPTIONS
.MATERIAL
.SHAPE

.LIB

Creates and reads from libraries of commonly used commands, device models, subcircuit analyses, and commands.

Syntax

Use the following syntax for library calls:

```
.LIB '[file_path] file_name' entry_name
```

Use the following syntax to define library files:

```
.LIB entryname1
. $ ANY VALID SET OF HSPICE STATEMENTS
.ENDL entry_name1
.LIB entry_name2
.
. $ ANY VALID SET OF HSPICE STATEMENTS
.ENDL entry_name2
.LIB entry_name3
.
. $ ANY VALID SET OF HSPICE STATEMENTS
.ENDL entry_name3
```

Arguments

Argument	Description
file_path	Path to a file. Used where a computer supports tree-structured directories. When the LIB file (or alias) is in the same directory where you run HSPICE RF you do not need to specify a directory path; the netlist runs on any machine. Use "../" syntax in the file_path to designate the parent directory of the current directory.
entry_name	Entry name for the section of the library file to include. The first character of an entry_name cannot be an integer. If more than one entry with the same name is encountered in a file, only the first one is loaded.
file_name	Name of a file to include in the data file. The combination of filepath plus file_name can be up to 256 characters long, structured as any filename that is valid for the computer's operating system. Enclose the file path and file name in single or double quotation marks. Use "../" syntax in the filename to designate the parent directory of the current directory.

Description

Use the .LIB call command to read from libraries of commonly used commands, device models, subcircuit analyses, and commands (library calls) in library files. Note that as HSPICE RF encounters each .LIB call name in the main data file, it reads the corresponding entry from the designated library file, until it finds an .ENDL command.

You can also place a .LIB call command in an .ALTER block.

To build libraries (library file definition), use the .LIB command in a library file. For each macro in a library, use a library definition command (.LIB *entry_name*) and an .ENDL command. The .LIB command begins the library macro and the .ENDL command ends the library macro. The text after a library file entry name must consist of HSPICE RF commands. Library calls can call other libraries (nested library calls) if they are different files. You can nest library calls to any depth. Use nesting with the .ALTER command to create a sequence of model runs. Each run can consist of similar components by using different model parameters without duplicating the entire input file.

The simulator uses the .LIB command and the .INCLUDE command to access the models and skew parameters. The library contains parameters that modify .MODEL commands.

You must enclose the file path and file name in single or double quotation marks. Otherwise, an error message is generated.

Example 1

```
* Library call
.LIB 'MODELS' cmos1
```

Example 2

```
.LIB MOS7
$ Any valid set of HSPICE RF commands
.
.
.
.ENDL MOS7
```

Example 3

This is an example of *illegal* nested .LIB commands for the file3 library.

```
.LIB MOS7
...
.LIB 'file3' MOS7 $ This call is illegal in MOS7 library
...
.ENDL
```

Example 4

```
.LIB TT
$TYPICAL P-CHANNEL AND N-CHANNEL CMOS LIBRARY
$ PROCESS: 1.0U CMOS, FAB7
$ following distributions are 3 sigma ABSOLUTE GAUSSIAN
.PARAM TOX=AGAUSS(200,20,3)    $ 200 angstrom +/- 20a
+ XL=AGAUSS(0.1u,0.13u,3)    $ polysilicon CD
+ DELVTON=AGAUSS(0.0,.2V,3)    $ n-ch threshold change
+ DELVTOP=AGAUSS(0.0,.15V,3)
   $ p-ch threshold change
.INC '/usr/meta/lib/cmos1_mod.dat'
   $ model include file
.ENDL TT
.LIB FF
$HIGH GAIN P-CH AND N-CH CMOS LIBRARY 3SIGMA VALUES
.PARAM TOX=220 XL=-0.03 DELVTON=-.2V
+ DELVTOP=-0.15V
.INC '/usr/meta/lib/cmos1_mod.dat'
   $ model include file
.ENDL FF
```

Example 4 is a .LIB call command of model skew parameters and features both worst-case and statistical distribution data. The statistical distribution median value is the default for all non-Monte Carlo analyses. The model is in the /usr/meta/lib/cmos1_mod.dat include file.

```
.MODEL NCH NMOS LEVEL=2 XL=XL TOX=TOX
+ DELVTO=DELVTON .....
.MODEL PCH PMOS LEVEL=2 XL=XL TOX=TOX
+ DELVTO=DELVTOP .....
```

The .MODEL keyword (left side) equates to the skew parameter (right side). A .MODEL keyword can be the same as a skew parameter.

See Also

.ALTER
.ENDL
.INCLUDE

.LIN

Extracts noise and linear transfer parameters for a general multiport network.

Syntax

Multiport Syntax

```
.LIN [sparcalc=[1|0] [modelname = modelname]]
+ [filename = filename] [format=selem|citi|touchstone]
+ [noisecalc=[2|1|0] [gdcalc=[1|0]]
+ [mixedmode2port=dd|dc|ds|cd|cc|cs|sd|sc|ss]>
+ [dataformat=ri|ma|db]
```

Two-Port Syntax

```
.LIN [sparcalc=1|0 [modelname = modelname]]
+ [filename = filename] [format=selem|citi|touchstone]
+ [noisecalc=1|0] [gdcalc=1|0]
+ [mixedmode2port=dd|dc|ds|cd|cc|cs|sd|sc|ss]
+ [dataformat=ri|ma|db]
+ [listfreq=(frequencies|none|all)]
+ [listcount=num] [listfloor=val] [listsources=1|0|on|off]
```

Arguments

Argument	Description
sparcalc	If 1, extract S parameters (default).
modelname	Model name to be listed in the .MODEL command in the .sc# model output file.
filename	Output file name (The default is netlist name).
format	Output file format: ■ selem is for S-element .sc# format, which you can include in the netlist. ■ citi is CITIfile format. ■ touchstone is TOUCHSTONE file format.

Argument	Description
noisecalc	Specifies level of N-port noise wave correlation matrix extraction. If 2, extract N noise parameters (perform multiport noise analysis). If 1, extract noise parameters (perform 2-port noise analysis). The default is 0.
gdcalc	If 1, extract group delay (perform group delay analysis). The default is 0.
mixedmode2port	The mixedmode2port keyword describes the mixed-mode data map of output mixed mode S-parameter matrix. The availability and default value for this keyword depends on the first two port (P-element) configuration as follows: ■ case 1: p1=p2=single (standard mode P element) available: ss default: ss ■ case 2: p1=p2=balanced (mixed mode P element) available: dd, cd, dc, cc default: dd ■ case 3: p1=balanced p2=single available: ds, cs default: ds ■ case 4: p1=single p2=balanced available: sd, sc default: sd
dataformat	The dataformat keyword describe the data format output to the *.sc#/touchstone/citi* file. ■ dataformat=RI, real-imaginary. This is the default for the .sc#/citi file. ■ dataformat=MA, magnitude-phase. This is the default format for touchstone file. ■ dataformat=DB, DB(magnitude)-phase. HSPICE uses six digits for both frequency and S-parameters in HSPICE generated data files (.sc#/touchstone/citifile). The number of digits for noise parameters are five in .sc# and Touchstone files and six in CITIfiles.

Argument	Description

listfreq= (none|all|freq1req2....)	Dumps the element noise figure value to the *.lis* file. You can specify which frequencies the element phase noise value dumps. The frequencies must match the sweep_frequency values defined in the parameter_sweep, otherwise they are ignored. In the element phase noise output, the elements that contribute the largest phase noise are dumped first. The frequency values can be specified with the NONE or ALL keyword, which either dumps no frequencies or every frequency defined in the parameter_sweep.
	▪ ALL: output all of the frequency points (default, if LIST* is required.)
	▪ NONE - do not output any of the frequency points
	▪ freq1 freq2...: output the information on the specified frequency points
	Frequency values must be enclosed in parentheses. For example:`listfreq=(none)`
	`listfreq=(all)`
	`listfreq=(1.0G)`
	`listfreq=(1.0G, 2.0G)`

listcount=*num*	Outputs the first few noise elements that make the biggest contribution to NF. The number is specified by *num*. The default is to output all of the noise element contribution to NF. The NF contribution is calculated with the source impedance equal to the Zo of the first port.
listfloor=*val*	Lists elements whose noise contribution to NF (in dB) are higher than value specified in dB to *.lis* file. Default is inf.
listsources= [1|0|yes|no]	Defines whether or not to output the contribution of each noise source of each noise element. Default is no/0

Description

Use this command to extract noise and linear transfer parameters for a general multiport network.

When used with P- (port) element(s) and `.AC` commands, `.LIN` makes available a broad set of linear port-wise measurements:

- standard and mixed-mode multiport S- (scattering) parameters
- standard and mixed-mode multiport Y/Z parameters

- standard mode multiport H-parameter
- standard mode two-port noise parameters
- standard and mixed-mode group delays
- standard mode stability factors
- standard mode gain factors
- standard mode matching coefficients

The `.LIN` command computes the S-(scattering), Y-(admittance), Z-(impedance) parameters directly, and H-(hybrid) parameters directly based on the location of the port (P) elements in your circuit, and the specified values for their reference impedances. The `.LIN` command also supports mixed-mode transfer parameters calculation and group delay analysis when used together with mixed-mode P elements.

To calculate the insertion and return loss for the high speed differential signal on my PCB board you can use the `.LIN` command with a port (P) element at input and output, where Port=1 defines the input and Port=2 defines the output. The return loss in dB is `|S111(DB)|` and the insertion loss in dB is `|S21(DB)|`.

By default, the `.LIN` command creates a .sc# file with the same base name as your netlist. This file contains S-parameter, noise parameter, and group delay data as a function of the frequency. You can use this file as model data for the S-element. Noise contributor tables are generated for every frequency point and every circuit device. The last four arguments allow users to better control the output information. If the `LIST*` arguments are not set, .LIN 2port noise analysis will output to *.lis* file with the older format. If any of the `LIST*` arguments is set, the output information follows the syntax noted in the arguments section.

Example

```
.LIN sparcalc=1 modelname=my_custom_model
+ filename=mydesign format=touchstone noisecalc=1
+ gdcalc=1 dataformat=ri
```

This example extracts linear transfer parameters for a general multiport network, performs a 2-port noise analysis and a group-delay analysis for a model named `my_custom_model`. The output is in the `mydesign` Touchstone format output file. The data format in the Touchstone file is real-imaginary.

See also

Linear Network Parameter Analysis

.LOAD

Uses the operating point information of a file previously created with a .SAVE command.

Syntax

.LOAD [FILE=*load_file*] [RUN=PREVIOUS|CURRENT]

Arguments

Argument	Description
load_file	Name of the file in which .SAVE saved an operating point for the circuit under simulation.The format of the file name is *design*.ic#. Default is *design*.ic0, where *design* is the root name of the design.
RUN	The format of file name is *design*.ic#. Used only outside of .ALTER commands in a netlist that contains .ALTER commands. ▪ PREVIOUS: Each .ALTER uses the saved operating point from the previous .ALTER run in the current simulation run. ▪ CURRENT: Each .ALTER uses the saved operating point from the corresponding .ALTER run in the previous simulation run.

Description

Use this command to input the contents of a file that you stored using the .SAVE command. Files stored with the .SAVE command contain operating point information for the point in the analysis at which you executed .SAVE.

Do not use the .LOAD command for concatenated netlist files.

Example 1

```
.SAVE FILE=design.ic0
.LOAD FILE=design.ic0
                $load--design.ic0 save--design.ic0
.alter
...             $load--none   save--design.ic1
.alter
...             $load--none   save--design.ic2
.end
```

This example loads a file name *design.ic0*, which you previously saved using a .SAVE command.

Example 2

```
.SAVE FILE=design.ic
.LOAD FILE=design.ic RUN=PREVIOUS
               $load--none   save--design.ic0
.alter
...            $load--design.ic0 save--design.ic1
.alter
...            $load--design.ic1 save--design.ic2
.end
```

Example 3

```
.SAVE FILE=design.ic
.LOAD FILE=design.ic RUN=CURRENT
              $load--design.ic0 save--design.ic0
.alter
...           $load--design.ic1 save--design.ic1
.alter
...           $load--design.ic2 save--design.ic2
.end
```

See Also

.ALTER
.SAVE

.LPRINT

Produces output in VCD file format from transient analysis in HSPICE RF. (Valid only for HSPICE RF.)

Syntax

```
.LPRINT (v1,v2) output_varable_list
```

Arguments

Argument	Description
v1, v2	Threshold values for digital output. Values less than v1 are output as digital 0. Values greater than 1 are output as digital 1.
output_varable_list	Output variables to `.PRINT`. These are variables from a DC, AC, TRAN, or NOISE analysis).

Description

Use this command to produce output in VCD file format from transient analysis.

Example

In this example, the `.LPRINT` command sets threshold values to 0.5 and 4.5, and the voltage level at voltage source VIN.

```
.OPTION VCD
.LPRINT (0.5,4.5) v(VIN)
```

See Also

.PRINT

.MACRO

Replicates output commands within subcircuit (subckt) definitions.

Syntax

```
.MACRO subckt_name n1 [n2 n3 ...] [parnam=val]
.EOM
```

Arguments

Argument	Description
subckt_nam	Reference name for the subcircuit model call.
n1 ...	Node numbers for external reference; cannot be the ground node (zero). Any element nodes that are in the subcircuit, but are not in this list strictly local with three exceptions: ▪ Ground node (zero). ▪ Nodes assigned using BULK=node in MOSFET or BJT models. ▪ Nodes assigned using the .GLOBAL command.
parnam	Parameter name set to a value. Use only in the subcircuit. To override this value, assign it in the subcircuit call or set a value in a .PARAM command.
SubDefaultsList	*SubParam1=Expression* [*SubParam2=Expression*...]

Description

Use this command to define a subcircuit in your netlist. You can create a subcircuit description for a commonly used circuit and include one or more references to the subcircuit in your netlist. Use the .EOM command to terminate a .MACRO command.

Example 1

Example 1 defines two subcircuits: SUB1 and SUB2. These are resistor divider networks, whose resistance values are parameters (variables). The X1, X2, and X3 commands call these subcircuits. Because the resistor values are different in each call, these three calls produce different subcircuits.

```
*FILE SUB2.SP TEST OF SUBCIRCUITS
.OPTION LIST ACCT
   V1 1 0 1
.PARAM P5=5 P2=10
.SUBCKT SUB1 1 2 P4=4
   R1 1 0 P4
   R2 2 0 P5
   X1 1 2 SUB2 P6=7
   X2 1 2 SUB2
.ENDS
*
.MACRO SUB2 1 2 P6=11
   R1 1 2 P6
   R2 2 0 P2
.EOM
   X1 1 2 SUB1 P4=6
   X2 3 4 SUB1 P6=15
   X3 3 4 SUB2
*
.MODEL DA D CJA=CAJA CJP=CAJP VRB=-20 IS=7.62E-18
+   PHI=.5 EXA=.5 EXP=.33
.PARAM CAJA=2.535E-16 CAJP=2.53E-16
.END
```

Example 2

```
.SUBCKT Inv a y Strength=3
   Mp1 <MosPinList> pMosMod L=1.2u W='Strength * 2u'
   Mn1 <MosPinList> nMosMod L=1.2u W='Strength * 1u'
.ENDS
...
xInv0 a y0 Inv         $ Default devices: p device=6u,
         $ n device=3u
xInv1 a y1 Inv Strength=5         $ p device=10u, n device=5u
xInv2 a y2 Inv Strength=1         $ p device= 2u, n device=1u
...
```

This example implements an inverter that uses a *Strength* parameter. By default, the inverter can drive three devices. Enter a new value for the *Strength* parameter in the element line to select larger or smaller inverters for the application.

See Also

.ENDS
.EOM
.SUBCKT

.MALIAS

Assigns an alias to a diode, BJT, JFET, or MOSFET model that you defined in a .MODEL command.

Syntax

.MALIAS *model_name=alias_name1* [*alias_name2 ...*]

Arguments

Argument	Description
model_name	Model name defined in the .MODEL card
alias_name1...	Alias that an instance (element) of the model references

Description

Use this command to assign an alias (another name) to a diode, BJT, JFET, or MOSFET model that you defined in a .MODEL command.

.MALIAS differs from .ALIAS in two ways:

- A model can define the alias in an .ALIAS command, but not the alias in a .MALIAS command. The .MALIAS command applies to an element (an instance of the model), not to the model itself.
- The .ALIAS command works only if you include .ALTER in the netlist. You can use .MALIAS without .ALTER.

You can use .MALIAS to alias to a model name that you defined in a .MODEL command or to alias to a subcircuit name that you defined in a .SUBCKT command. The syntax for .MALIAS is the same in either usage.

Note:

Do not use .MALIAS in .ALTER blocks.

Example

```
*file: test malias statement
.OPTION acct tnom=50 list gmin=1e-14 post
.temp 0.0 25
.tran .1 2
vdd 2 0 pwl 0 -1 1 1
d1 2 1 zend dtemp=25
d2 1 0 zen dtemp=25
* malias statements
.malias zendef=zen zend
* model definition
.model zendef d (vj=.8 is=1e-16 ibv=1e-9 bv=6.0 rs=10
+ tt=0.11n n=1.0 eg=1.11 m=.5 cjo=1pf tref=50)
.end
```

- `zendef` is a diode model
- `zen` and `zend` are its aliases.
- The `zendef` model points to both the `zen` and `zend` aliases.

See Also

.ALIAS
.MODEL

.MATERIAL

Specifies material to be used with the HSPICE field solver.

Syntax

```
.MATERIAL mname METAL|DIELECTRIC [ER=val]
+ [UR=val] [CONDUCTIVITY=val] [LOSSTANGENT=val]
+ ROUGHNESS=val
```

Arguments

Argument	Description
mname	Material name.
METAL|DIELECTRIC	Material type: METAL or DIELECTRIC.
ER	Dielectric constant (relative permittivity).
UR	Relative permeability.
CONDUCTIVITY	Static field conductivity of conductor or lossy dielectric (S/m).
LOSSTANGENT	Alternating field loss tangent of dielectric (tan δ).
ROUGHNESS	RMS surface roughness height, used when scaling the field solver.

Description

The field solver assigns the following default values for metal: `CONDUCTIVITY=-1` (perfect conductor), `ER=1`, `UR=1`.

`PEC` (perfect electrical conductor) is a predefined metal name. You cannot redefine its default values.The field solver assigns default values for dielectrics:

- `CONDUCTIVITY=0` (lossless dielectric)
- `LOSSTANGENT=0` (lossless dielectric)
- `ER=1`
- `UR=1`

`AIR` is a predefined dielectric name. You cannot redefine its default values. Because the field solver does not currently support magnetic materials, it ignores `UR` values.

See Also

.LAYERSTACK
.FSOPTIONS

.MEASURE (or) .MEAS

Modifies information to define the results of successive simulations.

Syntax

See the links below for the various syntaxes.

Description

Use this command to modify information and to define the results of successive HSPICE simulations. The `.MEASURE` command prints user-defined electrical specifications of a circuit. Optimization uses `.MEASURE` commands extensively. You can shorten the command name to `.MEAS`. The specifications include:

- Propagation
- Delay
- RIse time
- Fall time
- Peak-to-peak voltage
- Minimum and maximum voltage over a specified period
- Other user-defined variables

You can also use `.MEASURE` with either the error function (`ERRfun`) or `GOAL` parameter to optimize circuit component values, and to curve-fit measured data to model parameters.

The `.MEASURE` command can use several different formats, depending on the application. You can use it for DC sweep, and AC or transient analyses.

Note:

If a `.measure` command uses the result of previous `.meas` command, then the calculation starts when the previous result is found. Until the previous result is found, it outputs zero.

See Also

.MEASURE (Rise, Fall, and Delay Measurements)
.MEASURE (FIND and WHEN)
.MEASURE (Equation Evaluation/ Arithmetic Expression)
.MEASURE (AVG, EM_AVG, INTEG, MIN, MAX, PP, and RMS)
.MEASURE (Integral Function)
.MEASURE (Derivative Function)
.MEASURE (Error Function)

.MEASURE (Pushout Bisection)
.MEASURE(ACMATCH)
.MEASURE(DCMATCH)
.MEASURE FFT
.AC
.DC
.DCMATCH
.DOUT
.OPTION NCWARN
.OPTION MEASFAIL
.OPTION MEASFILE
.OPTION MEASOUT
.PRINT
.PROBE
.STIM
.TRAN

.MEASURE (Rise, Fall, and Delay Measurements)

Measures independent-variable differentials such as rise time, fall time, and slew rate.

Syntax

```
.MEASURE [DC|AC|TRAN] result TRIG ... TARG ...
+ [GOAL=val] [MINVAL=val] [WEIGHT=val]
```

The input syntax for delay, rise time, and fall time in HSPICE RF is:

```
.MEASURE [TRAN] varname TRIG_SPEC TARG_SPEC
```

In this syntax, *varname* is the user-defined variable name for the measurement (the time difference between TRIG and TARG events). The input syntax for *TRIG_SPEC* and *TARG_SPEC* is:

```
TRIG var VAL=val [TD=time] [CROSS=c|LAST]
+ [RISE=r|LAST] [FALL=f|LAST][TRIG AT=time]
TARG var VAL=val [TD=time] [CROSS=c|LAST]
+ [RISE=r|LAST] [FALL=f|LAST] [REVERSE][TARG AT=time]
```

Arguments

Argument	Description
DC | AC | TRAN	Analysis type of the measurement. If you omit this parameter, HSPICE uses the last analysis mode that you requested.
result	Name associated with the measured value in the HSPICE output, can be up to 16 characters long. This example measures the independent variable, beginning at the trigger and ending at the target: ▪ Transient analysis measures time. ▪ AC analysis measures frequency. ▪ DC analysis measures the DC sweep variable. If simulation reaches the target before the trigger activates, the resulting value is negative. Do not use DC, TRAN, or AC as the *result* name.
TRIG...	Beginning of trigger specifications.
TARG ...	Beginning of the target specification.

Argument	Description
GOAL	Desired measure value in ERR calculation for optimization. To calculate the error, the simulation uses the equation: $ERRfun = (GOAL - result) / GOAL$.
MINVAL	If the absolute value of GOAL is less than MINVAL, the MINVAL replaces the GOAL value in the denominator of the ERRfun expression. Used only in ERR calculation for optimization. The default is 1.0e-12.
WEIGHT	Multiplies the calculated error by the weight value. Used only in ERR calculation for optimization. The default is 1.0.

The following are parameters for the TRIG and TARG subcommands.

TRIG/TARG Parameter	Description
trig_val	Value of *trig_var*, which increments the counter by one for crossings, rises, or falls.
trig_var	Specifies the name of the output variable that determines the logical beginning of a measurement. If HSPICE reaches the target before the trigger activates, .MEASURE reports a negative value.
REVERSE	Allows trigger time to come before target time when added to the command.
TD	Amount of simulation time that must elapse before HSPICE enables the measurement. Simulation counts the number of crossings, rises, or falls only after the *time_delay* value. Default trigger delay is zero.
AT=val	Special case for trigger specification. *val* is: ▪ Time for TRAN analysis. ▪ Frequency for AC analysis. ▪ Parameter for DC analysis. ▪ SweepValue from .DC mismatch analysis. The trigger determines where measurement takes place.

Description

Use the Rise, Fall, and Delay form of the `.MEASURE` command to measure independent-variable (time, frequency, or any parameter or temperature)

differentials such as rise time, fall time, slew rate, or any measurement that requires determining independent variable values. This format specifies TRIG and TARG subcommands. These two commands specify the beginning and end of a voltage or current amplitude measurement.

Example 1

```
* Example of rise/fall/delay measurement
.MEASURE TRAN tdlay TRIG V(1) VAL=2.5 TD=10n
+ RISE=2 TARG V(2) VAL=2.5 FALL=2
```

This example measures the propagation delay between nodes 1 and 2 for a transient analysis. HSPICE measures the delay from the second rising edge of the voltage at node 1 to the second falling edge of node 2. The measurement begins when the second rising voltage at node 1 is 2.5 V and ends when the second falling voltage at node 2 is 2.5 V. The TD=10n parameter counts the crossings after 10 ns has elapsed. HSPICE prints results as tdlay=<value>.

Example 2

```
.MEASURE TRAN riset TRIG I(Q1) VAL=0.5m RISE=3
+ TARG I(Q1) VAL=4.5m RISE=3
* Rise/fall/delay measure with TRIG and TARG specs
.MEASURE pwidth TRIG AT=10n TARG V(IN) VAL=2.5
+ CROSS=3
```

In Example 2, TRIG AT=10n starts measuring time at t=10 ns in the transient analysis. The TARG parameters terminate time measurement when V(IN) = 2.5 V on the third crossing. pwidth is the printed output variable.

If you use the .TRAN analysis command with a .MEASURE command, do not use a non-zero start time in the .TRAN command to avoid incorrect .MEASURE results.

Example 3

```
.MEAS TRAN TDEL12 TRIG V(signal1) VAL='VDD/2'
+ RISE=10 TARG V(signal2) VAL='VDD/2' RISE=1 TD=TRIG
```

Example 3 shows a target that is delayed until the trigger time before the target counts the edges.

.MEASURE (FIND and WHEN)

Measures independent and dependent variables (as well as derivatives of dependent variables if a specific event occurs).

Syntax

```
.MEASURE [DC|AC|TRAN] result WHEN out_var=val [TD=val]
+ [FROM=val] [TO=val]
+ [RISE=r|LAST][FALL=f|LAST][CROSS=c|LAST][REVERSE]
+ [GOAL=val] [MINVAL=val] [WEIGHT=val]

.MEASURE [DC|AC|TRAN] result
+ WHEN out_var1=out_var2 [TD=val] [RISE=r|LAST]
+ [FALL=f|LAST] [CROSS=c|LAST] [GOAL=val] [MINVAL=val]
+ [WEIGHT=val]

.MEASURE [DC|AC|TRAN] result FIND out_var1
+ WHEN out_var2=val [TD=val] [FROM=val] [TO=val]
+ [RISE=r|LAST][FALL=f|LAST] [CROSS=c|LAST] [REVERSE]
+ [GOAL=val][MINVAL=val] [WEIGHT=val]

.MEASURE [DC|AC|TRAN] result FIND out_var1
+ WHEN out_var2=out_var3 [TD=val]
+ [RISE=r|LAST] [FALL=f|LAST] [REVERSE]
+ [CROSS=c|LAST] [GOAL=val] [MINVAL=val] [WEIGHT=val]

.MEASURE [DC|AC|TRAN] result FIND out_var1
+ AT=val [FROM=val] [TO=val] [GOAL=val][MINVAL=val]
+ [WEIGHT=val]
```

Arguments

Argument	Description
DC | AC | TRAN	Analysis type of the measurement. If you omit this parameter, HSPICE uses the last analysis mode that you requested.
result	Name of a measured value in the HSPICE output.
WHEN	WHEN function.
out_var(1,2,3)	Variables that establish conditions to start a measurement.

Argument	Description
TD	Time at which measurement starts.
FROM... TO...	Allows adding multiple trigger conditions to some WHEN measurements.

CROSS=c
RISE=r
FALL=f
	Numbers indicate which CROSS, FALL, or RISE event to measure. For example:
	`.meas tran tdlay trig v(1) val=1.5 td=10n`
	`+ rise=2 targ v(2) val=1.5 fall=2`
	In this example, rise=2 specifies the measure of the v(1) voltage only on the first two rising edges of the waveform. The value of these first two rising edges is 1. However, trig v(1) val=1.5 indicates to trigger when the voltage on the rising edge voltage is 1.5, which never occurs on these first two rising edges. So the v(1) voltage measurement never finds a trigger.
	RISE=r, the WHEN condition is met and measurement occurs after the designated signal has risen *r* rise times.
	FALL =f, measurement occurs when the designated signal has fallen *f* fall times.
	A crossing is either a rise or a fall so for CROSS=*c*, measurement occurs when the designated signal has achieved a total of *c* crossing times as a result of either rising or falling.
	For TARG, the LAST keyword specifies the last event.

LAST	HSPICE measures when the last CROSS, FALL, or RISE event occurs. ▪ CROSS=LAST, measurement occurs the last time the WHEN condition is true for a rising or falling signal. ▪ FALL=LAST, measurement occurs the last time the WHEN condition is true for a falling signal. ▪ RISE=LAST, measurement occurs the last time the WHEN condition is true for a rising signal. LAST is a reserved word; you cannot use it as a parameter name in the above .MEASURE commands.
REVERSE	Allows FALL to precede RISE in specified .MEAS commands, when declared.

Argument	Description
GOAL	Desired .MEASURE value. Optimization uses this value in ERR calculation. The following equation calculates the error: $ERRfun = (GOAL - result)/\ GOAL$ In HSPICE RF output you cannot apply .MEASURE to waveforms generated from another .MEASURE command in a parameter sweep.
MINVAL	If the absolute value of GOAL is less than MINVAL, then MINVAL replaces the GOAL value in the denominator of the ERRfun expression. Used only in ERR calculation for optimization. The default is 1.0e-12.
WEIGHT	Calculated error multiplied by the weight value. Used only in ERR calculation for optimization. The default is 1.0.
FIND	FIND function.
AT=val	Special case for trigger specification. *val* is: ▪ Time for TRAN analysis. ▪ Frequency for AC analysis. ▪ Parameter for DC analysis. ▪ SweepValue from .DC mismatch analysis. The trigger determines where measurement takes place.

Description

The `FIND` and `WHEN` functions of the `.MEASURE` command measure:

- Any independent variables (time, frequency, parameter).
- Any dependent variables (voltage or current for example).
- A derivative of a dependent variable if a specific event occurs.

Example1

Calculating Voltage

```
* MEASURE statement using FIND/WHEN

.MEAS TRAN TRT FIND PAR('V(3)-V(4)')
+ WHEN V(1)=PAR('V(2)/2') RISE=LAST
.MEAS STIME WHEN V(4)=2.5 CROSS=3
```

In this example, the first measurement, TRT, calculates the difference between V(3) and V(4) when V(1) is half the voltage of V(2) at the last rise event.

The second measurement, STIME, finds the time when V(4) is 2.5V at the third rise-fall event. A CROSS event is a rising or falling edge.

Example 2

Using a DC Sweep Variable

```
* sweep measure
v0 1 0 3
r0 1 0 x
.dc x 1 5 1
.meas res find par(x) when i(r0)=2
.end
```

By adding `par()` to the sweep variable it can be used in a `.MEASURE` command.

.MEASURE (Equation Evaluation/ Arithmetic Expression)

Evaluates an equation that is a function of the results of previous .MEASURE commands.

Syntax

```
.MEASURE [DC|TRAN|AC] result PARAM='equation'
+ [GOAL=val] [MINVAL=val]
.MEASURE TRAN varname PARAM="expression"
```

Arguments

Argument	Description
DC | AC | TRAN	Analysis type of the measurement. If you omit this parameter, HSPICE uses the last analysis mode that you requested.
result	Name of a measured value in the HSPICE output.
PARAM='equation'	Equation wrapped in single quotes, a function of the results of previous .MEASURE commands.
GOAL	Desired .MEASURE value. In HSPICE RF output you cannot apply .MEASURE to waveforms generated from another .MEASURE command in a parameter sweep.
MINVAL	If the absolute value of GOAL is less than MINVAL, then MINVAL replaces the GOAL value in the denominator of the ERRfun expression. Used only in ERR calculation for optimization. The default is 1.0e-12.
TRAN	Transient analysis results.
varname	Name of variable to be used in evaluation.
Param="expression"	Arithmetic expression that uses results from other prior .MEASURE commands.

Description

Use the Equation Evaluation form of the .MEASURE command to evaluate an equation that is a function of the results of previous .MEASURE commands. The equation must not be a function of node voltages or branch currents.

The *expression* option is an arithmetic expression that uses results from other prior .MEASURE commands.

Expressions used in arithmetic expression must not be a function of node voltages or branch currents. Expressions used in all other .MEASURE commands can contain either node voltages or branch currents, but must not use results from other .MEASURE commands.

Example

```
.MEAS TRAN V3MAX MAX V(3) FROM 0NS TO 100NS
.MEAS TRAN V2MIN MIN V(2) FROM 0NS TO 100NS
.MEAS VARG PARAM='(V2MIN + V3MAX)/2'
```

The first two measurements, V3MAX and V2MIN, set up the variables for the third .MEASURE command.

- V3MAX is the maximum voltage of V(3) between 0ns and 100ns of the simulation.
- V2MIN is the minimum voltage of V(2) during that same interval.
- VARG is the mathematical average of the V3MAX and V2MIN measurements.

.MEASURE (AVG, EM_AVG, INTEG, MIN, MAX, PP, and RMS)

Reports statistical functions of the output variable.

Syntax

```
.MEASURE [DC|AC|TRAN] result func out_var
+ [FROM=val] [TO=val] [GOAL=val] MINVAL=val] [WEIGHT=val]
```

Arguments

Argument	Description
DC|AC|TRAN	Analysis type for the measurement. If you omit this parameter, HSPICE defaults to the last analysis mode that you requested.
result	Name of the measured value in the output, can be up to 16 characters long. The value is a function of the variable (*out_var*) and *func*.
func	Indicates one of the following measure function types: ▪ AVG (average): Calculates the area under the out_var, divided by the periods of interest. ▪ INTEG (Integral function): Reports the integral of an output variable over a specified period. ▪ MIN (minimum): Reports the minimum value of the out_var over the specified interval. ▪ MAX (maximum): Reports the maximum value of the out_var over the specified interval. ▪ PP (peak-to-peak): Reports the maximum value, minus the minimum value of the out_var over the specified interval. ▪ RMS (root mean squared): Calculates the square root of the area under the out_var2 curve, divided by the period of interest. ▪ EM_AVG:Calculates the average electromigration current. For a symmetric bipolar waveform, the current is: I_avg (0, T/2) - R*Iavg (T/2, T), where R is the recovery factor specified using .option em_recovery. Wildcards are also suppoted during this measurement. See Example 5 (Wildcards) below.
out_var	Name of any output variable whose function (*func*) the simulation measures.
FROM	Initial value for the INTEG calculation.

Argument	Description
TO	End of the INTEG calculation.
GOAL	.MEASURE value. Optimization uses this value for ERR calculation. This equation calculates the error: $ERRfun = (GOAL - result) / GOAL$ In HSPICE RF simulation output you cannot apply .MEASURE to waveforms generated from another .MEASURE command in a parameter sweep.
WEIGHT	Calculated error multiplied by the weight value. Used only in ERR calculation for optimization. The default is 1.0.

Description

Average (AVG), EM_AVG, RMS, MIN, MAX, and peak-to-peak (PP) measurement modes report statistical functions of the output variable, rather than analysis values. Wildcards are supported for the From-To functions for AVG, EM_AVG, RMS, MIN, MAX and PP measurement (unlike other measurement functions).

AVG, RMS, and INTEG have no meaning in a DC data sweep so if you use them, HSPICE issues a warning message.

Example 1 (AVG)

```
.MEAS TRAN avgval AVG V(10) FROM=10ns TO=55ns
```

This example calculates the average nodal voltage value for node 10 during the transient sweep from the time 10ns to 55ns. It prints out the result as avgval.

Example 2 (MAX)

```
.MEAS TRAN MAXVAL MAX V(1,2) FROM=15ns  TO=100ns
```

This example finds the maximum voltage difference between nodes 1 and 2 for the time period from 15 ns to 100 ns.

Example 3 (MIN)

The first command finds the minimum voltage difference between nodes 1 and 2 over the time period 15 ns to 100 ns. The second command measures the peak to peak current through transistor M1 from 10ns to 100ns.

```
.MEAS TRAN MINVAL MIN V(1,2) FROM=15ns TO=100ns
.MEAS TRAN P2PVAL PP I(M1)  FROM=10ns TO=100ns
```

Example 4 (EM_AVG)

In this example, the coefficient value is set by `.option recovery=`*`value`*. The electromagnetic migration average is measured from 5 ns to 10.2 ns.

```
.option em_recovery=0.2
.measure tran vout_1 EM_AVG v(5) from=5ns to=10.2ns
```

These commands measure the result parameter currents called `em1` and `em2` over the ranges specified.

```
.measure tran em1 em_avg i(rload) from=1n to=3.5n
.measure tran em2 em_avg i(rload) from=4n to=9n
```

Example 5 (Wildcards)

This example does the following operations (using the wildcard (`*`):

- Finds the average of all the positive currents (Ipos_avg) from 5ns to 50ns.
- Finds the average (absolute value) of all the negative currents (Ineg_avg) from 5ns to 50ns.
- Performs the operation "`Ipos_avg - R*Ineg_avg`"
  Where `R` is a user-provided coefficient following use of `.option em_recovery=`*`value`*.

```
.MEASURE TRAN EM_AVG I(OUT) FROM=5N TO=50N
```

where: `(OUT)` is the node at which the measurement is taken.

Example 6 (RMS)

In this example, the `.MEASURE` command calculates the `RMS` voltage of the `OUT` node from 0ns to 10ns. It then labels the result `RMSVAL`.

```
.MEAS TRAN RMSVAL RMS V(OUT) FROM=0NS TO=10NS
```

Example 7 (MAX)

In this example, the `.MEASURE` command finds the maximum current of the VDD voltage supply between 10ns and 200ns in the simulation. The result is called `MAXCUR`.

```
.MEAS MAXCUR MAX I(VDD) FROM=10NS TO=200NS
```

Example 8 (PP)

In this example, the `.MEASURE` command uses the ratio of `V(OUT)` and `V(IN)` to find the peak-to-peak value in the interval of 0ns to 200ns.

```
.MEAS P2P PP PAR('V(OUT)/V(IN)') FROM=0NS TO=200NS
```

Example 9 (AVG and RMS)

The `P-element`, `P(instance_name)` and `POWER` keywords can be used in `AVG` and `RMS .MEASURE` commands. For example:

```
.meas tran AVG_Sckt AVG p(x1)   $ Measure the AVG power of instance
                                  "X1"
.meas tran RMS_Ckt_Pwr RMS power  $ Measure the RMS power of the
                                     total circuit
```

See Also

.OPTION EM_RECOVERY

.MEASURE (Integral Function)

Reports the real time integration (instantaneous time integral) of an output variable over a specified period.

Syntax

```
.MEASURE [DC|AC|TRAN] result INTEG[RAL] out_var
+ [FROM=val] [TO=val] [GOAL=val]
+ [MINVAL=val] [WEIGHT=val]
```

Arguments

Argument	Description
DC | AC | TRAN	Analysis type of the measurement. If you omit this parameter, HSPICE uses the last analysis mode that you requested.
result	Name of a measured value in the HSPICE output.
INTEG	Integral function to find an output variable over a specified period.
outvar	Name of any output variable whose function the simulation measures.
FROM	Initial value for the *func* calculation. For transient analysis, this value is in units of time.
TO	End of the *func* calculation.
GOAL	Desired .MEASURE value. In HSPICE RF output you cannot apply .MEASURE to waveforms generated from another .MEASURE command in a parameter sweep.
MINVAL	If the absolute value of GOAL is less than MINVAL, then MINVAL replaces the GOAL value in the denominator of the ERRfun expression. Used only in ERR calculation for optimization. The default is 1.0e-12.

Description

The INTEGRAL function reports the integral of an output variable over a specified period. The INTEGRAL function uses the same syntax as the AVG (average), RMS, MIN, MAX and peak-to-peak (PP) measurement modes.

Examples

This example calculates the integral of I(cload) from 10ns to 100ns.

```
.MEAS TRAN charge INTEG I(cload) FROM=10ns TO=100ns
```

The following .MEASURE command calculates the integral of I(R1) from 50ns to 200 ns.

```
.MEASURE TRAN integ_i INTEGRAL I(r1) FROM=50ns TO=200ns
```

.MEASURE (Derivative Function)

Provides the derivative of an output or sweep variable.

Syntax

```
.MEASURE [DC|AC|TRAN result DERIV[ATIVE] out_var
+ [FROM=val] [TO=val] AT=val [GOAL=val] [MINVAL=val]
+ [WEIGHT=val]

.MEASURE [DC|AC|TRAN] result DERIV<ATIVE> out_var
+ [FROM=val TO=val] WHEN var2=val [RISE=r|LAST]
+ [FALL=f|LAST] [CROSS=c|LAST] [TD=tdval]
+ [GOAL=goalval] [MINVAL=minval] [WEIGHT=val]

.MEASURE [DC|AC|TRAN] result DERIV[ATIVE] out_var
+ [FROM=val] [TO=val] WHEN var2=var3 [RISE=r|LAST]
+ [FALL=f|LAST] [CROSS=c|LAST] [TD=tdval]
+ [GOAL=val] [MINVAL=val] [WEIGHT=val]
```

Arguments

Argument	Description
DC | AC | TRAN	Analysis type of the measurement. If you omit this parameter, HSPICE uses the last analysis mode that you requested.
result	Name of the measured value in the output.
DERIVATIVE	Derivative function.
out_var	Variable for which HSPICE finds the derivative.
FROM=*val* TO=*val*	Specifies a range to measure, such as time window.
var(2,3)	Variables establish that conditions to start a measurement.
AT=*val*	Value of *out_var* at which the derivative is found.

Argument	Description
GOAL	Specifies the desired .MEASURE value. Optimization uses this value for ERR calculation. This equation calculates the error: $ERRfun = (GOAL - result)/\ GOAL$ In HSPICE RF output you cannot apply .MEASURE to waveforms generated from another .MEASURE command in a parameter sweep.
MINVAL	If the absolute value of GOAL is less than MINVAL, MINVAL replaces the GOAL value in the denominator of the ERRfun expression. Used only in ERR calculation for optimization. The default is 1.0e-12.
WEIGHT	Calculates the error between result and GOAL by multiplied by the weight value. Used only in ERR calculation for optimization. The default is 1.0.
WHEN	WHEN function.
RISE=r FALL=f CROSS=c	Numbers indicate which occurrence of a CROSS, FALL, or RISE event starts a measurement. ▪ For RISE=r when the designated signal has risen r rise times, the WHEN condition is met and measurement starts. ▪ For FALL=f, measurement starts when the designated signal has fallen f fall times. ▪ A crossing is either a rise or a fall so for CROSS=c, measurement starts when the designated signal has achieved a total of c crossing times as a result of either rising or falling.
LAST	Last CROSS, FALL, or RISE event. ▪ CROSS=LAST, measures the last time the WHEN condition is true for a rising or falling signal. ▪ FALL=LAST, measures the last time WHEN is true for a falling signal. ▪ RISE=LAST, measures the last time WHEN is true for a rising signal. LAST is a reserved word; do not use it as a parameter name in the above .MEASURE commands.
TD	Time when measurement starts.

Description

The `DERIV` function provides the derivative of:

- An output variable at a specified time or frequency.
- Any sweep variable, depending on the type of analysis.
- A specified output variable when some specific event occurs.

Example 1

```
.MEAS TRAN slew rate DERIV V(out) AT=25ns
```

This example calculates the derivative of V(out) at 25 ns.

Example 2

```
.MEAS TRAN slew DERIV v(1) WHEN v(1)='0.90*vdd'
```

This example calculates the derivative of v(1) when v(1) is equal to 0.9*vdd.

Example 3

```
.MEAS AC delay DERIV 'VP(output)/360.0' AT=10khz
```

This example calculates the derivative of VP(output)/360.0 when the frequency is 10 kHz.

Example 4

```
.MEAS DC result find v(in) when derive v(out)= …
```

This example measures the derivative of a nodal waveform.

Example 5

```
.MEAS DC result derive v(out) …
```

If you plot "`result`" from the command above you will get the dV(out)/dTemperature vs Temperature plot.

Example 6

The following example measures and finds when the maximum derivative of a signal occurs.

```
.probe dt=deriv("v(out)")
.meas m0 max par(dt)
.meas m1 when par(dt)=m0
```

The example shows (1) a probe of the derivative of the signal, (2) the maximum value of the derivative, and (3) when the maximum value of the derivative occurred.

.MEASURE (Error Function)

Reports the relative difference between two output variables.

Syntax

```
.MEASURE DC|AC|TRAN> result
+ ERRfun meas_var calc_var
+ [MINVAL=val] [IGNOR|YMIN=val]
+ [YMAX=val] [WEIGHT=val] [FROM=val] [TO=val]
```

Arguments

Argument	Description
DC|AC|TRAN	Analysis type for the measurement. If you omit this parameter, HSPICE defaults to the last analysis mode requested.
result	Name of the measured result in the output.
ERRfun	Error function to use: ERR, ERR1, ERR2, or ERR3.
meas_var	Name of any output variable or parameter in the data command. `M` denotes the *meas_var* in the error equation.
calc_var	Name of the simulated output variable or parameter in the .MEASURE command to compare with *meas_var*. `C` is the *calc_var* in the error equation.
MINVAL	If the absolute value of *meas_var* is less than *MINVAL*, *MINVAL* replaces the meas_var value in the denominator of the *ERRfun* expression. Used only in ERR calculation for optimization. Default: 1.0e-12.
IGNOR|YMIN	If the absolute value of *meas_var* is less than the *IGNOR* value, then the *ERRfun* calculation does not consider this point. Default: 1.0e-15.
YMAX	If the absolute value of *meas_var* is greater than the *YMAX* value, then the *ERRfun* calculation does not consider this point. Default: 1.0e+15.
WEIGHT	Calculates error multiplied weight value. Used only in ERR calculation for optimization. The default is 1.0.

Argument	Description
FROM	**Specifies the b**eginning of the ERRfun calculation. For transient analysis, the *FROM* value is in units of time. Defaults to the first value of the sweep variable.
TO	End of the *ERRfun* calculation. Default is last value of the sweep variable.

Description

The relative error function reports the relative difference between two output variables. You can use this format in optimization and curve-fitting of measured data. The relative error format specifies the variable to measure and calculate from the `.PARAM` variable. To calculate the relative error between the two, HSPICE uses the `ERR`, `ERR1`, `ERR2`, or `ERR3` functions. With this format you can specify a group of parameters to vary to match the calculated value and the measured data.

Examples

```
.measure ac comp1 err1 par(s11m) s11(m)
.measure tran re1 err1 par(out2) v(out) from=1u to=2u
```

.MEASURE PHASENOISE

Enables measurement of phase noise at various frequency points in HSPICE RF.

Syntax

Find-When ... Phase Noise

```
.MEASURE PHASENOISE result FIND phnoise At = IFB_value
.MEASURE PHASENOISE result WHEN phnoise=value
```

RMS, average, min, max, and peak-to-peak Phase Noise

```
.MEASURE PHASENOISE result func phnoise
+ [FROM = IFB1] [TO = IFB2]
```

Integral Evaluation of Phase Noise

```
.MEASURE PHASENOISE result INTEGRAL phnoise
+ [FROM = IFB1] [TO = IFB2]
```

Derivative Evaluation of Phase noise

```
.MEASURE PHASENOISE result DERIV[ATIVE] phnoise AT = IFB1
```

Amplitude modulation noise

```
.MEASURE phasenoise result AM[NOISE] phnoise
+ [FROM = IFB1] [TO = IFB2]
```

Phase modulation noise

```
.MEASURE phasenoise result PM[NOISE] phnoise
+ [FROM = IFB1] [TO = IFB2]
```

Arguments

Parameter	Description
FIND	Selects the FIND function
result	Name of the measured result in the output.
phnoise	.MEASURE PHASENOISE value for phase noise
WHEN	Selects the WHEN function

Parameter	Description
IFB_value	Input frequency band point value
func	Indicates one of the measure command types: ■ AVG (average): Calculates the phase noise over the frequency range. ■ MAX (maximum): Reports the maximum value of the phase noise over the specified frequency range. ■ MIN (minimum): Reports the minimum value of the phase noise over the specified frequency range. ■ PP (peak-to-peak): Reports the maximum value, minus the minimum value of the phase noise over the specified frequency range. ■ RMS (root mean squared): Calculates the square root of the phase noise over the specified frequency range.
FROM...TO	Optional range for input frequency bands (IFB)
INTEGRAL	Integrates the phase noise value from the first to the second IFB frequency points
DERIVATIVE	Finds the derivative of the phase noise at the first IFB frequency point
PM[NOISE]	Measures the phase modulation noise from the specified first to the second IFB frequency points (when `.OPTION PHASENOISEAMPM=1`)
AM[NOISE]	Measures the amplitude modulation noise from the specified first to the second IFB frequency points (when `.OPTION PHASENOISEAMPM=1`)

Description

The `.MEASURE PHASENOISE` syntax supports yielding the following phase noise instances in dbc/Hz:

- Yields the phase noise using FIND or WHEN functions: at a specified input frequency band (FIND), or phase noise found at a specified input frequency point (WHEN).
- Yields the average, RMS, minimum, maximum, or peak-to-peak value of the phase noise from frequency `IFB1` to frequency `IFB2`, where the value of `func` can be RMS, AVG, MIN, MAX or PP. If `FROM` and `TO` are not specified, the value will be calculated over the frequency range specified in the `.PHASENOISE` command.

- Integrates the phase noise value from the IFB1 frequency to the IFB2 frequency.
- Finds the derivative of phase noise at the IFB1 frequency point.

Note:

The .MEASURE PHASENOISE command cannot contain an expression that uses a phase noise variable as an argument. You also cannot use .MEASURE PHASENOISE for error measurement and expression evaluation of PHASENOISE.

The HSPICE RF optimization flow can read the measured data from a .MEASURE PHASENOISE analysis. This flow can be combined in the HSPICE RF optimization routine with a .MEASURE HBTR analysis.

Examples

The FIND keyword yields the result of a variable value at a specific input frequency band (IFB) point.

```
.MEASURE PHASENOISE np1 find PHNOISE at=100K
```

The WHEN keyword yields the input frequency point at a specific phase noise value.

```
.MEASURE PHASENOISE fcorn1 when PHNOISE=-120
```

The following sample command find functions such as the RMS, AVG, MIN, MAX, or PP over the frequency range.

```
.measure PHASENOISE rn1 RMS phnoise
.measure PHASENOISE agn1 AVG phnoise from=100k to=10meg
.measure PHASENOISE nmin MIN phnoise
```

The INTEGRAL command integrates the phase noise across the two specified Input frequency band points.

```
.measure PHASENOISE inns1 INTEGRAL phnoise
.measure PHASENOISE rns1 INTEGRAL phnoise from=50k to 500k
```

These DERIV sample commands find the derivative of the phase noise at one input frequency band point.

```
.measure PHASENOISE dnf1 DERIVATIVE phnoise at=100k
.measure PHASENOISE fdn1 DERIVATIVE phnoise at=10meg
```

These `AM/PM` sample commands find the amplitude modulation (AM) and phase modulation (PM) noise across the specified input frequency range.

```
.measure PHASENOISE amp1 AM phnoise from=100k to 400k
.measure PHASENOISE pmp1 PM phnoise from=10meg to=30meg
```

See Also

.PHASENOISE
.MEASURE PTDNOISE
.MEASURE (FIND and WHEN)
.MEASURE (AVG, EM_AVG, INTEG, MIN, MAX, PP, and RMS)
.MEASURE (Integral Function)
.MEASURE (Derivative Function)
Measuring Phase Noise with .MEASURE PHASENOISE
.HB
.OPTION PHNOISEAMPM

.MEASURE PTDNOISE

Allows for the measurement of these integnoise, time-point, tdelta-value, slewrate, and strobed jitter parameters in HSPICE RF.

Syntax

```
.MEASURE PTDNOISE meas_name STROBEJITTER onoise freq_sweep
```

Arguments

Parameter	Units	Description
strobed jitter	sec	Calculated from the noise voltage (integrated over the frequency range specified by *frequency_range*), divided by the slewrate at the same node(s), at the time point specified by *time_value*.
While only STROBEJITTER can be specified, all of the parameters listed below are also output to the **.msnptn#* file.		
integptdnoise	V	Total ptd voltage noise in $V^2/(Hz)$ (integrated over a frequency range specified by *frequency_range*) at the time point specified by *time_value*. The value is stated as a voltage (V).
timepoint	sec	Time point at which the ptdnoise and slewrate are calculated.
tdelta-value	sec	TDELTA value used to calculate slewrate.
slewrate	v/sec	Output signal slewrate at the time point specified by *time_value*.

Description

Use to obtain strobed jitter parameters in large signal periodic time dependent noise analysis. For more information, see the *HSPICE User Guide: RF Analysis* section on Periodic Time-Dependent Noise Analysis (.PTDNOISE).

See Also

.PTDNOISE
.MEASURE Syntax and File Format

.MEASURE (Pushout Bisection)

Specifies a maximum allowed pushout time to control the distance from failure in bisection analysis.

Syntax

```
.MEASURE TRAN result MeasureClause
+ pushout=time [lower|upper]
-or-
.MEASURE TRAN result MeasureClause
+ pushout_per=percentage [lower|upper]
```

Arguments

Argument	Description
result	Name associated with the measured value in the HSPICE output, can be up to 16 characters long.
pushout=time	Appropriate time to obtain the pushout result (an absolute time).
pushout_per= percentage	Relative error. If you specify a 0.1 relative error, the T_lower or T_upper and T_pushout have more than a 10% difference in value. This occurrence causes the iteration to stop and output the optimized parameter.
lower|upper	Parameter boundary values for pushout comparison. These arguments are optional. If the parameter is defined as `.PARAM ParamName= OPTxxx(Initial, min, max)`, the “lower” means the lower bound “min”, and the “upper” means the upper bound “max”. The default is lower.

Description

Pushout is used only in bisection analysis. In Pushout Bisection, instead of finding the last point just before failure you specify a maximum allowed pushout time to control the distance from failure.

To limit the range you can add both absolute and relative pushout together.

For example:

```
.Measure Tran pushout When v(D_Output)='vih/2'
+ rise=1 pushout=20p,50p pushout_per=0.1
```

The final measure result for the preceding example should be in the range of:

```
'goldmeas+ min(50p, max(0.10 * goldmeas, 20p))'
```

...or the final measure result should satisfy,

```
| measresult-goldmeas | < Min[ pushout_max,
      Max( pushout_per*goldmeas, pushout_min ) ]
```

Example 1

```
.Param DelayTime=Opt1 ( 0.0n, 0.0n , 5.0n )
.Tran 1n 8n Sweep Optimize=Opt1  Result=setup_prop  + Model=OptMod
.Measure Tran setup_prop Trig v(data)
+ Val='v(Vdd) 2' fall=1 Targ v(D_Output)
+ Val='v(Vdd)' rise=1 pushout=1.5n lower
```

In this example, the parameter to be optimized is `Delaytime` and the evaluation goal is `setup_prop`. The `Pushout=1.5 lower` means that the `setup_prop` of the final solution is not `1.5n` far from the `setup_prop` of the lower bound of the parameter (`0.0n`).

Example 2

```
.Measure Tran setup_prop Trig v(data)
+ Val='v(Vdd)/2' fall=1 Targ v(D_Output)
+ Val='v(Vdd)' rise=1 pushout_per=0.1 lower
```

In this example, the differences between the `setup_prop` of the final solution and that of the lower bound of the parameter (`0.0n`) is not more than 10%.

.MEASURE(ACMATCH)

Introduces special keywords to access results for ACMatch analysis in HSPICE.

Syntax

```
.MEASURE DC result [MAX][ACM_Total|ACM_Global|
+ ACM_Global(par)|ACM_Local|ACM_Local(dev)]
```

Arguments

Argument	Description
results	Name associated with the measured values in the HSPICE output, can be up to 16 characters long.
MAX	Sample function; Instead of “MAX” other functions can be used which select one out of multiple results.
ACM_Total	Output sigma due to global, local, and spatial variations.
ACM_Global	Output sigma due to global variations.
ACM_Global(par)	Contribution of parameter (*par*) to output sigma due to global variations.
ACM_Local	Output sigma due to local variations.
ACM_Local(dev)	Contribution of device (*dev*) to output sigma due to local variations.

Description

ACMatch analysis saves results using `.MEASURE` commands, with AC type (M,P,R,I) for an output variable, as specified on the `.ACMatch` command. If you specify multiple output variables the command issues a result for the last one only. You must specify an AC sweep to produce these kinds of outputs; a single point sweep is sufficient. ACMatch uses the special keywords shown above to access the results from the different variation types. For usable keywords with the `.PROBE` command, see Output from .PROBE and .MEASURE Commands for ACMatch in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.AC
.MEASURE(ACMATCH)
.PROBE

.MEASURE(DCMATCH)

Introduces special keywords to access the different types of results for DCMatch analysis in HSPICE.

Syntax

```
.MEASURE DC result [MAX] [DCM_Total | DCM_global |
+ DCM_Global(par) | DCM_Local | DCM_Local(dev) |
+ DCM_Spatial | DCM_Spatial(par)]
```

Arguments

Argument	Description
result	Name associated with the measured values in the HSPICE output, can be up to 16 characters long.
MAX	Sample function. Instead of “MAX,” other functions can be used which select one out of multiple results.
DCM_Total	Output sigma due to global, local, and spatial variations.
DCM_Global	Output sigma due to global variations.
DCM_Global(par)	Contribution of parameter (*par*) to output sigma due to global variations.
DCM_Local	Output sigma due to local variations.
DCM_Local(dev)	Contribution of device (*dev*) to output sigma due to local variations.
DCM_Spatial	Output sigma due to local variations.
DCM_Spatial(par)	Contribution of parameter (*par*) to output sigma due to spatial variations.

Description

DCMatch analysis uses special keywords to access the different types of results. You can save the different results produced by a DCMatch analysis using the `.MEASURE` command for the output variable specified on the `.DCMatch` command. For keywords to be used with the `.PROBE` command, see Syntax for .PROBE Command for DCMatch in the *HSPICE User Guide: Simulation and Analysis*. If you specify multiple output variables, the command produces a result for the last one only. You must specify a DC sweep to produce these kinds of outputs; a single point sweep is sufficient.

Example

In this example, the result `systoffset` reports the systematic offset of the amplifier; the result `matchoffset` reports the variation due to mismatch; and the result `maxoffset` reports the maximum (3-sigma) offset of the amplifier.

```
.MEAS DC systoffset avg V(inp,inn)
.MEAS DC matchoffset avg DCm_local
.MEAS DC maxoffset
param='abs(systoffset)+3.0*matchoffset'
```

See Also

.DC
.PROBE

.MEASURE FFT

Specifies measurement of FFT results.

Syntax

Syntax #1

```
.MEASURE FFT result
+ Find [vm|vp|vr|vi|vdb|im|ip|ir|ii|idb](signal) AT=freq
```

Syntax #2

```
.MEASURE FFT result THD signal_name [nbharm=num]
```

Syntax #3

```
.MEASURE FFT result[SNR|SNDR|ENOB] signal_name
+ [nbharm=num|maxfreq=val]
```

Syntax #4

```
.MEASURE FFT result SFDR signal_name
+ [minfreq=val][maxfreq=val]
```

Arguments

Argument	Description
result	Name associated with the measured values in the FFT output, can be up to 16 characters long.
Find	FiND function.
At	Value of the *frequency* at which the component frequency and signal are found
frequency component and signal	Can be any of the following: vm|vp|vr|vi|vdb|im|ip|ir|ii|idb
freq	Specified frequency
THD	Total harmonic distortion
signal_name	User-supplied name of signal

Argument	Description
nbharm	Harmonic up to which to carry out the measurement. Default: highest harmonic in FFT result.
maxfreq	Higher limit of frequency range to carrying out the measurement. Default: maximum frequency in an FFT result.
minfreq	Lower limit of frequency range to calculate SFDR.
SNR	Signal to noise ratio.
SNDR	Signal to noise-plus-distortion ratio.
ENOB	Effective number of bits.
SFDR	Spurious free dynamic range.

Description

Four syntaxes are provided for finding measurements of several types for FFT results. See examples below for sample usage.

- Syntax #1: Measures a frequency component at certain frequency.
- Syntax #2: Measures THD of a signal spectrum up to a specified harmonic; Default: `nbharm`=maximum harmonic in FFT result
- Syntax # 3: Measures SNR/SNDR/ENOB of a signal up to a specified frequency; Defaults: `nbharm`=maximum harmonic in FFT result; `maxfreq`=maximum frequency in FFT result
- Syntax # 4: Measures SFDR of a signal from `minfreq` to `maxfreq`; searches the frequency component with maximum magnitude from `minfreq` to `maxfreq`.

An embedded `.MEASURE FFT` command in a measure file can be called to perform FFT measurements from previous simulation results as follows:

```
HSPICE -i *.tr0 -meas measure_file
```

Examples

1. Measuring frequency component at certain frequency.

```
.meas FFT v12 Find vm(1,2)at=20k
```

2. Measuring THD of a signal spectrum up to a specified harmonic

```
.meas FFT thd56 THD V(node5, node6) nbharm=10
```

3. Measuring SNR/SNDR/ENOB of a signal up to a specified frequency

```
.meas FFT snr12 SNDR V(node1, node2) maxfreq=1G
```

4. Measuring SFDR of a signal from `minfreq` to `maxfreq` and searching the frequency component with maximum magnitude from `minfreq` to `maxfreq`.

```
.meas FFT sfdr9 SFDR V(node9)
```

See Also

.FFT
Spectrum Analysis

.MODEL

Includes an instance of a predefined HSPICE model in an input netlist.

Syntax

Passive and active device model syntax

```
.MODEL mname type [level=num]
+ [pname1=val1 pname2=val2 ...]
```

See specific element type for supported model parameter information.

Optimization model syntax

```
.MODEL mname OPT [METHOD][close=num] [max] [cut=num]
+ [difsiz=num] [grad=num] [parmin=num] [relin=num]
+ [relout=num] [itrop=num] [DYNACC=0|1]
```

The following syntax is used for a Monte Carlo analysis:

```
.MODEL mname ModelType ([level=val]
+ [keyword1=val1][keyword2=val2]
+ [keyword3=val3][lot/n /distribution value]
+ [DEV/n /distribution value]...)
```

See Chapter 20, Monte Carlo Analysis Using the Variation Block Flow in the *HSPICE User Guide: Simulation and Analysis* for more information.

The following syntax is used for model reliability analysis

```
.model mname mosra
+ level|mosralevel value
+ [relmodelparam]
```

Arguments

Argument	Description
mname	Model name reference. Elements must use this name to refer to the model. If model names contain periods (.), the automatic model selector might fail. When used with .MOSRA it is the user defined MOSFET reliability model name

Argument	Description
type	Model type. Must be one of the following.: ■ AMP—operational amplifier model ■ C—capacitor model ■ CORE—magnetic core model ■ D—diode model ■ L—inductor model or magnetic core mutual inductor model ■ NJF—n-channel JFET model ■ NMOS—n-channel MOSFET model ■ NPN—npn BJT model ■ OPT—optimization model ■ PJF—p-channel JFET model ■ PLOTQ—plot model for the .GRAPH command ■ PMOS—p-channel MOSFET model ■ PNP—pnp BJT model ■ R—resistor model ■ U—lossy transmission line model (lumped) ■ W—lossy transmission line model ■ S—S-parameter
level	Model level. ■ For optimization model, LEVEL=1 specifies the Modified Levenberg-Marquardt method. Use this setting with multiple optimization parameters and goals. ■ Only Level=1 is available in HSPICE. Additional levels are available in HSPICE RF. See below This argument is ignored when METHOD has been specified. ■ LEVEL=2 (HSPICE RF) specifies the BISECTION method in HSPICE RF. You would use this setting with one optimization parameter. ■ LEVEL=3 (HSPICE RF) specifies the PASSFAIL method. You would use this setting with two optimization parameters. ■ For transistors, diodes, and some passive element models, see the HSPICE Reference Manual, Elements and Device Models. ■ For MOSFET Models, see the HSPICE Reference Manual: MOSFET Models. ■ To use custom MOSRA models and for discussion of LEVEL values, refer to the HSPICE Application Note: Unified Custom Reliability Modeling API (MOSRA API). Contact your Synopsys technical support team for more information.

Argument	Description
pname1 ...	Parameter name. Assign a model parameter name (*pname1*) from the parameter names for the appropriate model type. Each model section provides default values. For legibility, enclose the parameter assignment list in parentheses and use either blanks or commas to separate each assignment. Use a plus sign (+) to start a continuation line.
METHOD	Specifies an optimization method. ▪ METHOD=BISECTION specifies the Bisection method. ▪ METHOD=PASSFAIL specifies the passfail method. This argument supersedes LEVEL when present.
close	(Optimization) Initial estimate of how close parameter initial value estimates are to the solution. The close argument multiplies changes in new parameter estimates. If you use a large close value, the optimizer takes large steps toward the solution. For a small value, the optimizer takes smaller steps toward the solution. You can use a smaller value for close parameter estimates and a larger value for rough initial guesses. The default is 1.0. ▪ If close is greater than 100, the steepest descent in the Levenburg-Marquardt algorithm dominates. ▪ If close is less than 1, the Gauss-Newton method dominates. For more details, see L. Spruiell, “Optimization Error Surfaces,” *Meta-Software Journal*, Volume 1, Number 4, December 1994.
max	(Optimization) Upper limit on close. Use values > 100. The default is 6.0e+5.
cut	(Optimization) Modifies close, depending on how successful iterations are toward the solution. If the last iteration succeeds, descent toward the close solution decreases by the cut value. That is, close=close / cut If the last iteration was not a successful descent to the solution, close increases by cut squared. That is, close=close * cut * cut. Cut drives close up or down, depending on the relative success in finding the solution. The cut value must be > 1. The default is 2.0.
difsiz	(Optimization) Increment change in a parameter value for gradient calculations ($\Delta x = DIFSIZ \cdot MAX(x, 0.1)$). If you specify delta in a .PARAM command, then Δx=delta. The default is 1e-3.

Argument	Description
grad	(Optimization) Represents possible convergence if the gradient of the RESULTS function is less than GRAD. Most applications use values of 1e-6 to 1e-5. Too large a value can stop the optimizer before finding the best solution. Too small a value requires more iterations. The default is 1.0e-6.
parmin	(Optimization) Allows better control of incremental parameter changes during error calculations. The default is 0.1. This produces more control over the trade-off between simulation time and optimization result accuracy. To calculate parameter increments, HSPICE uses the relationship: $\Delta par\_val = \Delta IFSIZ \cdot MAX(par\_val, PARMIN)$
relin	(Optimization) Relative input parameter (delta_par_val / MAX(par_val,1e-6)) for convergence. If all optimizing input parameters vary by no more than RELIN between iterations, the solution converges. RELIN is a relative variance test so a value of 0.001 implies that optimizing parameters vary by less than 0.1%from one iteration to the next. The default is 0.001.
relout	(Optimization) Relative tolerance to finish optimization. For relout=0.001, if the relative difference in the RESULTS functions from one iteration to the next, is less than 0.001, then optimization is finished. The default is 0.001.
itropt	(Optimization) Maximum number of iterations. Typically, you need no more than 20-40 iterations to find a solution. Too many iterations can imply that the relin, grad, or relout values are too small. The default is 20.
DYNACC	(Optimization) Dynamic accuracy tolerance setting to accelerate bisection simulation. The default is 0. When DYNACC=1, if HSPICE is in accuracy mode, it uses reduced accuracy simulations to narrow the bisection window, then switches to the original accuracy algorithm to refine the solution. This method reduces simulation time by doing the majority of simulations at lower accuracy, which run faster by taking fewer time steps.
keyword	(Monte Carlo) Model parameter keyword.

Argument	Description
distribution	(Monte Carlo) The distribution function name, which must be specified as GAUSS, AGAUSS, LIMIT, UNIF, or AUNIF. If you do not set the distribution function, the default distribution function is used. The default distribution function is uniform distribution.
DEV	(Monte Carlo) DEV tolerance, which is independent (each device varies independently).
LOT	(Monte Carlo) The LOT tolerance, which requires all devices that refer to the same model use the same adjustments to the model parameter.
LOT/n DEV/n	(Monte Carlo) Which of ten random number generators numbered 0 through 9 are used to calculate parameter value deviations. This correlates deviations between parameters in the same model as well as between models. The generators for DEV and LOT tolerances are distinct: Ten generators exist for both DEV tracking and LOT tracking. N must be an integer 0 to 9.
relmodelparam	Model parameter for HCI and NBTI, when doing a reliability MOSFET device analysis.

Description

Use this command to include an instance (element) of a predefined HSPICE model in your input netlist.

For each optimization within a data file, specify a `.MODEL` command. HSPICE can then execute more than one optimization per simulation run. The `.MODEL` optimization command defines:

- Convergence criteria
- Number of iterations
- Derivative methods

Example 1

```
.MODEL MOD1 NPN BF=50 IS=1E-13 VBF=50 AREA=2 PJ=3 N=1.05
```

Example 2

Example 2 shows the addition of the `DYNACC=1` option in an optimization model card to invoke bisection speedup.

```
.MODEL optmod OPT METHOD=BISECTION  ITROPT=20 dynacc=1 relout=1e20
```

Example 3

In this example, a .MODEL command used for a Monte Carlo analysis.

```
.model m1 nmos level=6 bulk=2 vt=0.7 dev/2 0.1
+ tox=520 lot/gauss 0.3 a1=.5 a2=1.5 cdb=10e-16
+ csb=10e-16 tcv=.0024
```

Example 4

In this example, transistors M1 through M3 have the same random vto model parameter for each of the five Monte Carlo runs through the use of the LOT construct.

```
...
.model mname nmos level=53 vto=0.4 LOT/agauss 0.1 version=3.22
M1 11 21 31 41 mname W=20u L=0.3u
M2 12 22 32 42 mname W=20u L=0.3u
M3 13 23 33 43 mname W=20u L=0.3u
...
.dc v1 0 vdd 0.1 sweep monte=5
.end
```

Example 5

In this example, transistors M1 through M3 have different values of the vto model parameter for each of the Monte Carlo runs through the use of the DEV construct.

```
...
.model mname nmos level=54 vto=0.4 DEV/agauss 0.1
M1 11 21 31 41 mname W=20u L=0.3u
M2 12 22 32 42 mname W=20u L=0.3u
M3 13 23 33 43 mname W=20u L=0.3u
...
.dc v1 0 vdd 0.1 sweep monte=5
.end
```

Example 6

This example establishes a MOS reliability model card.

```
.model NCH_RA mosra
+ level=1
+ a_hci=1e-2
+ n_hci=1
```

.MOSRA

Starts HSPICE HCI and/or NBTI reliability analysis for HSPICE.

Syntax

```
.MOSRA RelTotalTime=time_value
+ [RelStartTime=time_value] [DEC=value] [LIN=value]
+ [RelStep=time_value] [RelMode=0|1|2] SimMode=[0|1|3]
+ [AgingStart=time_value] [AgingStop=time_value]
+ [AgingPeriod=time_value] [AgingWidth=time_value]
+ [AgingInst="inst_name"]
+ [HciThreshold=value] [NbtiThreshold=value]
+ [Integmod=0|1|2] [Xpolatemod=0|1|2]
+ [Tsample1=value] [Tsample2=value]
```

Argument	Description
RelTotalTime	Final reliability test time to use in post-stress simulation phase. Required argument.
RelStartTime	Time point of the first post-stress simulation. Default is 0.
DEC	Specifies number of post-stress time points simulated per decade.
LIN	Linear post-stress time points from RelStartTme to RelTotalTime.
RelStep	Post-stress simulation phase on time= RelStep, 2* RelStep, 3* RelStep, … until it achieves the RelTotalTime; the default is equal to RelTotalTime. Value is ignored if DEC or LIN value is set.
RelMode	Simulation accounts for both HCI and NBTI effects or either one of them. If RelMode in the `.MOSRA` command is not set or set to 0, then the RelMode inside individual MOSRA models takes precedence for that MOSRA model only; the rest of the MOSRA models take the RelMode value from the `.MOSRA` command. If any other value except 0,1, or 2 is set, a warning message is issued, and RelMode is set to default value 0. The RelMode parameter is also available in the MOSRA model card with additional restrictions. ▪ 0: both HCI and NBTI, Default. ▪ 1: HCI only ▪ 2: NBTI only

Argument	Description
SimMode	■ 0: Select pre-stress simulation only ■ 1: Select post-stress simulation only ■ 2: Select both pre- and post-stress simulation If set to 1, HSPICE reads in the **.radeg0* file for post-stress simulation (only) and uses it to update the input for reliability analysis; the transient output is in a **.tr1* waveform file. The **.radeg* file should be in the same directory as the input netlist. When SimMode =1, the netlist stimuli could be different from the SimMode=0 netlist that generated the **.radeg* file.
AgingStart	Optionally defines time when HSPICE starts stress effect calculation during transient simulation. Default is 0.0.
AgingStop	Optionally defines time when HSPICE stops stress effect calculation during transient simulation. Default is tstop in `.TRAN` command.
AgingPeriod	Stress period. Scales the total degradation over time.
AgingWidth	The AgingWidth (circuit time “on”) argument works with the AgingPeriod argument. For example: if you specify AgingPeriod=1.0s and AgingWidth=0.5s, then the circuit is turned on for 0.5s, and turned off for 0.5s. (The period is 1.0s.)
AgingInst	Selects MOSFET devices to which HSPICE applies HCI and/or NBTI analysis. The default is all MOSFET devices with reliability model appended. The name must be surrounded by quotes. Multiple names are allowed, and wildcards are supported.
HciThreshold	Optionally, used in post-stress simulation. HCI effect is accounted for in a particular transistor, based on the specified HCI threshold value. Default is 0.0
NbtiThreshold	Optionally, used in post-stress simulation. NBTI effect is accounted for in a particular transistor, based on this threshold value. Default is 0.0
Integmod	The flag is used to select the integration method and function. ■ 0 (default): call the MOSRAAPI integration function ■ 1: True derivation and integration method ■ 2: Linearization and integration method (support non-constant n coefficient)

Argument	Description
Xpolatemod	The flag is used to select the extrapolation method and function. ■ 0 (default): call the MOSRAAPI extrapolation function ■ 1: Linearization extrapolation method (support non-constant n coefficient) ■ 2: Aeff/Neff extraction and extrapolation method
Tsample1	First simulation time point of stress_total sampling for Aeff/Neff extraction
Tsample2	Second simulation time point of stress_total sampling for Aeff/Neff extraction

Description

Use the `.MOSRA` command to initiate HCI and NBTI analysis. This is a two-phase simulation, the fresh simulation phase and the post stress simulation phase. During the fresh simulation phase, HSPICE computes the electron age/stress of selected MOS transistors in the circuit based on circuit behavior and the HSPICE built-in stress model including HCI and/or NBTI effect. During the post stress simulation phase, HSPICE simulates the degradation effect on circuit performance, based on the stress information produced during the fresh simulation phase. If you specify either DEC or LIN, the RelStep value is ignored.

For a full description refer to the *HSPICE User Guide: Simulation and Analysis*: MOSFET Model Reliability Analysis (MOSRA).

Example

```
.mosra reltotaltime=6.3e+8 relstep=6.3e+7
+ agingstart=5n agingstop=100n
+ hcithreshold=0 nbtithreshold=0
+ aginginst="x1.*"
```

See also

.APPENDMODEL
.MODEL

.MOSRAPRINT

Provides .PRINT/.PROBE usage to the .MOSRA command.

Syntax

```
.MOSRAPRINT output_name output_type(element_name, vds=exp1,
   vgs=exp2, vbs=exp3)
```

Argument	Description
output_nam	User-defined output variable; this output_name@element_name is used as the as output variable name in the output file.
output_type	One of the following output variable types: vth, gm, gds, or ids.
element_name	Output variable name in the output file.

Definition

This command provides measurement functionality to the .MOSRA command. This syntax prints the Ids value of the MOSFET at the final reliability test time. There is no order requirement for vds, vgs, and vbs. You can use the wildcards '?' and '*' in element_name.

The output file format is the same as the measurement file format. The extension file name for this file is **.ra#*.

Example

The following syntax prints the ids value of the MOSFET m1, when vds = 5 vgs=5, vbs=0, at the reltime point.

```
.MOSRA reltotaltime=5e+7 relstep=1e+7
.MOSRAPRINT ids(m1, vds=5, vgs=5, vbs=0)
```

See Also

.MOSRA

.NODESET

Initializes specified nodal voltages for DC operating point analysis and corrects convergence problems in DC analysis.

Syntax

```
.NODESET V(node1)=val1 V(node2)=val2 ...
-or-
.NODESET node1 val1 node2 val2
```

Arguments

Argument	Description
node1 ...	Node numbers or names can include full paths or circuit numbers.
val1	Voltages.

Description

Use the .NODESET command to set a seed value for the iterative DC convergence algorithm for all specified nodal voltages. Use this to correct convergence problems in DC analysis. How it behaves depends on whether the .TRAN analysis command includes the UIC parameter.

Forcing circuits are connected to the .NODESET nodes for the first iteration of DC convergence. To increase the number of held iterations, see .OPTION DCHOLD. The forcing circuits are then removed and Newton Raphson iterations continued until DC convergence is obtained. The .NODESET nodes can move to their true DC operating points. For this reason, .NODESET should be used to provide initial guesses to either speed up convergence, aid non-convergence, or to set the preferred DC state of multistable nodes. If the DC operating voltage of a .NODESET node is appreciably different than the voltage in the .NODESET command you should investigate the circuit to determine why. It is a likely error condition.

Note:

In nearly all applications you should use .NODESET to ensure a true DC operating point. Set intentionally floating (or very high impedance) nodes to a known good voltage using .IC.

If you do not specify the UIC parameter in the .TRAN command then use .NODESET to set seed values for an initial guess for DC operating point

analysis. If the node value is close to the DC solution then you will enhance convergence of the simulation.

If you specify the UIC parameter in the .TRAN command, HSPICE does not calculate the initial DC operating point, but directly enters transient analysis. When you use .TRAN UIC, the .TRAN node values (at time zero) are determined by searching for the first value found in this order: from .IC value, then IC parameter on an element command, then .NODESET value, otherwise use a voltage of zero.

Note that forcing a node value of the DC operating point might not satisfy KVL and KCL. In this event you might see activity during the initial part of the simulation. This might happen if you use UIC and do not specify some node values, when you specify too many conflicting .IC values, or when you force node values and topology changes. Forcing a node voltage applies a fixed equivalent voltage source during DC analysis and transient analysis removes the voltage sources to calculate the second and later time points.

Therefore to correct DC convergence problems use .NODESETs (without .TRAN UIC) liberally (when a good guess can be provided) and use .ICs sparingly (when the exact node voltage is known).

In addition, you can use wildcards in the .NODESET command. See Using Wildcards on Node Names in the *HSPICE User Guide: Simulation and Analysis*.

Example

```
.NODESET V(5:SETX)=3.5V V(X1.X2.VINT)=1V
.NODESET V(12)=4.5 V(4)=2.23
.NODESET 12 4.5 4 2.23 1 1
```

See Also

.DC
.IC
.OPTION DCHOLD
.TRAN

.NOISE

Controls the noise analysis of the circuit.

Syntax

```
.NOISE v(out) vin interval
+ [listckt=[1|0]]
+ [listfreq=(frequencies|none|all)]
+ [listcount=num] [listfloor=val]
+ [listsources=1|0|yes|no]]
```

Arguments

Argument	Description
v(out)	Nodal voltage or branch current output variable. Defines the node or branch at which HSPICE sums the noise.
vin	Independent voltage source to use as the noise input reference
interval | inter	Interval at which HSPICE prints a noise analysis summary. *inter* specifies how many frequency points to summarize in the AC sweep. If you omit *inter* or set it to zero, HSPICE or HSPICE RF does not print a summary. If inter is equal to or greater than one, HSPICE prints summary for the first frequency, and once for each subsequent increment of the inter frequency. The noise report is sorted according to the contribution of each node to the overall noise level. If any of the LIST* arguments below are specified, the output information will follow the format required by LIST*, and interval does not influence the output information for later sweeps.
listckt= [1|0]	■ 1: The contribution of each subcircuit is listed in the *.lis* file and you can view the subcircuit noise contribution curve in CosmosScope. ■ 0: (default) HSPICE does not list the noise contribution of any subcircuits.

Argument	Description

listfreq= (none|all|freq1 freq2....)	Dumps the element noise figure value to the *.lis* file. You can specify which frequencies the element phase noise value dumps. The frequencies must match the sweep_frequency values defined in the parameter_sweep, otherwise they are ignored. In the element phase noise output, the elements that contribute the largest phase noise are dumped first. The frequency values can be specified with the NONE or ALL keyword, which either dumps no frequencies or every frequency defined in the parameter_sweep.
	■ ALL: output all of the frequency points (default, if LIST* is required.)
	■ NONE: do not output any of the frequency points
	■ freq1 freq2...: output the information on the specified frequency points
	Frequency values must be enclosed in parentheses. For example:
	`listfreq=(none)`
	`listfreq=(all)`
	`listfreq=(1.0G)`
	`listfreq=(1.0G, 2.0G)`

listcount=*num*	Outputs the first few noise elements that make the biggest contribution to NF. The number is specified by *num*. The default is to output all of the noise element contribution to NF. The NF contribution is calculated with the source impedance equal to the Zo of the first port.
listfloor=*val*	Contribution to the output noise power greater than the value specified by LISTFLOOR. Default is to output all the noise elements. The unit of LISTFLOOR is V^2/hz
listsources= [1|0|yes|no]	Defines whether or not to output the contribution of each noise source of each noise element. Default is no/0.

Description

Use this command and `.AC` commands to control the noise analysis of the circuit. You can use this command only with an `.AC` command. Noise contributor tables are generated for every frequency point and every circuit device. The last four arguments allow users to better control the output information.

Example 1

This example sums the output noise voltage at the node 5 by using the voltage source VIN as the noise input reference and prints a noise analysis summary every 10 frequency points.

```
.NOISE V(5) VIN 10
```

Example 2

This example sums the output noise current at the r2 branch by using the voltage source VIN as the noise input reference and prints a noise analysis summary every 5 frequency points.

```
.NOISE I(r2) VIN 5
```

Example 3

The following example shows the list subcircuit option turned on and sample results:

```
.NOISE listckt=1

**********************************************************************
 **** subcircuit squared noise voltages (sq v/hz)
 x1    total  1.90546e-20
 x7    total  7.14403e-19
 x1.x3    total  1.90546e-20
**********************************************************************
```

See Also

.AC

.OP

Calculates the DC operating point of the circuit; saves circuit voltages at multiple timesteps.

Syntax

```
.OP format time format time... [interpolation]
...
.op voltage time1 time2 ...
```

Arguments

Argument	Description
format	Any of the following keywords. Only the first letter is required. The default is ALL ▪ ALL: Full operating point, including voltage, currents, conductances, and capacitances. This parameter outputs voltage/current for the specified time. ▪ BRIEF: One-line summary of each element's voltage, current, and power. Current is stated in milliamperes and power in milliwatts. ▪ CURRENT: Voltage table with a brief summary of element currents and power. ▪ DEBUG: Usually invoked only if a simulation does not converge. Debug prints the non-convergent nodes with the new voltage, old voltage, and the tolerance (degree of non-convergence). It also prints the non-convergent elements with their tolerance values. ▪ NONE: Inhibits node and element printouts, but performs additional analysis that you specify. ▪ VOLTAGE: Voltage table only. The preceding keywords are mutually-exclusive; use only one at a time.
time	Time at which HSPICE prints the report. Place this parameter directly after `ALL`, `VOLTAGE`, `CURRENT`, or `DEBUG`. HSPICE RF returns node voltages only if time (t) is 0.
interpolation	Interpolation method for `.OP` time points during transient analysis or no interpolation. Only the first character is required; that is, typing **i** has the same effect as typing **interpolation**. Default is not active. If you specify *interpolation*, all of the time points in the `.OP` command (except time=0) use the interpolation method to calculate the OP value during the transient analysis. If you use this keyword, it must be at the end of the `.OP` command. HSPICE ignores any word after this keyword.

Description

Use this command to calculate the DC operating point of the circuit. You can also use the .OP command to produce an operating point during a transient analysis. You can include only one .OP command in a simulation.

If an analysis requires calculating an operating point you do not need to specify the .OP command; HSPICE calculates an operating point. If you use a .OP command and if you include the UIC parameter in a .TRAN analysis command, then simulation omits the time=0 operating point analysis and issues a warning in the output listing.

Use .OP to output circuit node voltages at different timesteps to **.ic0* files. You can replace use of the .SAVE command to save node voltages. The **.ic0* files are identical to those created by the .SAVE command. (Remove.SAVE commands to avoid conflict with the .OP command used to save node voltages.)

Example 1

```
.OP .5NS CUR 10NS VOL 17.5NS 20NS 25NS
```

This example calculates:

- Operating point at .05ns.
- Currents at 10 ns for the transient analysis.
- Voltages at 17.5 ns, 20 ns and 25 ns for the transient analysis.

Example 2

```
.OP
```

This example calculates a complete DC operating point solution.

See Also

.TRAN

.OPTION

Modifies various aspects of an HSPICE simulation; individual options for HSPICE and HSPICE RF commands are described in Chapter 3, HSPICE and RF Netlist Simulation Control Options.

Syntax

```
.OPTION opt1 [opt2 opt3 ...]
```

Arguments

Argument	Description
opt1 ...	Input control options. Many options are in the form *opt*=*x*, where *opt* is the option name and `x` is the value assigned to that option.

Description

Use this command to modify various aspects of an HSPICE simulation, including:

- output types
- accuracy
- speed
- convergence

You can set any number of options in one `.OPTION` command, and you can include any number of `.OPTION` commands in an input netlist file. Most options default to `0` (OFF) when you do not assign a value by using either `.OPTION` *`opt=val`* or the option with no assignment: `.OPTION` *`opt`*.

To reset options, set them to `0` (`.OPTION` *`opt`*`=0`). To redefine an option, enter a new `.OPTION` command; HSPICE uses the last definition.

You can use the following types of options with this command. For detailed information on individual options, see Chapter 3, HSPICE and RF Netlist Simulation Control Options.

- General Control Options
- Input/Output Controls
- Model and Variation Definition
- Analysis

For instructions on how to use options that are relevant to a specific simulation type, see the appropriate analysis chapters in the *HSPICE User Guide: Simulation and Analysis* for:

- Initializing DC/Operating Point Analysis
- Pole/Zero Analysis
- Spectrum Analysis
- Transient Analysis
- AC Small-Signal and Noise Analysis
- Linear Network Parameter Analysis
- Timing Analysis Using Bisection
- Analyzing Variability and Using the Variation Block
- Monte Carlo Analysis Using the Variation Block Flow
- Mismatch Analyses
- Optimization
- RC Reduction and Post-Layout Simulation
- MOSFET Model Reliability Analysis (MOSRA)

Example

```
.OPTION BRIEF $ Sets BRIEF to 1 (turns it on)
* Netlist, models,
...
.OPTION BRIEF=0 $ Turns BRIEF off
```

This example sets the `BRIEF` option to `1` to suppress a printout. It then resets `BRIEF` to `0` later in the input file to resume the printout.

.PARAM

Defines parameters in HSPICE and HSPICE RF.

Syntax

Simple parameter assignment:

```
.PARAM ParamName=RealNumber
```

Algebraic parameter assignments:

```
.PARAM ParamName='AlgebraicExpression'
.PARAM ParamName1=ParamName2
```

User-defined functions:

```
.PARAM ParamName(pv1[pv2])='Expression'
```

predefined analysis functions:

```
.PARAM FunctionName=Value
```

Optimized parameter assignment:

```
.PARAM parameter=OPTxxx (initial_guess, low_limit,
+ upper_limit)
.PARAM parameter=OPTxxx (initial_guess, low_limit,
+ upper_limit, delta)
.PARAM paramname=str('string')
```

Arguments

Argument	Description
parameter	Parameter to vary. ■ Initial value estimate. ■ Lower limit. ■ Upper limit. If the optimizer does not find the best solution within these constraints, it attempts to find the best solution without constraints.
OPTxxx	Optimization parameter reference name. The associated optimization analysis references this name. Must agree with the *OPTxxx* name in the analysis command associated with an OPTIMIZE keyname.

Argument	Description
delta	The final parameter value is the initial guess ± (*n*· *delta*). If you do not specify *delta*, the final parameter value is between *low_limit* and *upper_limit*. For example, you can use this parameter to optimize transistor drawn widths and lengths, which must be quantized.

Description

Use this command to define parameters. Parameters in HSPICE are names that have associated numeric values.

A parameter definition always uses the last value found in the input netlist (subject to global parameter rules).

Use any of the following methods to define parameters:

- A simple parameter assignment is a constant real number. The parameter keeps this value unless a later definition changes its value or an algebraic expression assigns a new value during simulation. HSPICE does not warn you if it reassigns a parameter.
- An algebraic parameter (equation) is an algebraic expression of real values, a predefined or user-defined function or circuit or model values. Enclose a complex expression in single quotes to invoke the algebraic processor, *unless* the expression begins with an alphabetic character and contains no spaces. A simple expression consists of a single parameter name. To use an algebraic expression as an output variable in a `.PRINT`, or `.PROBE` command, use the `PARAM` keyword.
- A user-defined function assignment is similar to an algebraic parameter. HSPICE extends the algebraic parameter definition to include function parameters, used in the algebraic that defines the function. You can nest user-defined functions up to three levels deep.
- A predefined analysis function. HSPICE provides several specialized analysis types, which require a way to control the analysis:
  - Temperature functions (fn)
  - Optimization guess/range

HSPICE also supports the following predefined parameter type:

- frequency
- time
- Monte Carlo functions

Note:

To print the final evaluated values of all `.PARAM` commands in the netlist, use `.OPTION LIST`. This helps you avoid seeing the same value for every time point if you run a transient analysis.

Example 1

Simple parameter assignment

```
.PARAM power_cylces=256
```

Example 2

Numerical parameter assignment

```
.PARAM TermValue=1g
   rTerm Bit0 0 TermValue
   rTerm Bit1 0 TermValue
...
```

Example 3

Parameter assignment using expressions

```
.PARAM Pi             ='355/113'
.PARAM Pi2             ='2*Pi'
.PARAM npRatio          =2.1
.PARAM nWidth          =3u
.PARAM pWidth          ='nWidth * npRatio'
Mp1   ... <pModelName> W=pWidth
Mn1   ... <nModelName> W=nWidth
...
```

Example 4

Algebraic parameter

```
.param x=cos(2)+sin(2)
```

Example 5

Algebraic expression as an output variable

```
.PRINT DC v(3) gain=PAR('v(3)/v(2)')
+ PAR('V(4)/V(2)')
```

Example 6

User-defined functions

```
.PARAM <MyFunc( x, y )>='Sqrt((x*x)+(y*y))'
.PARAM CentToFar (c)                 ='(((c*9)/5)+32)'
.PARAM F(p1,p2)                ='Log(Cos(p1)*Sin(p2))'
.PARAM SqrdProd (a,b)                ='(a*a)*(b*b)'
```

Example 7

Predefined analysis function

```
.PARAM mcVar=Agauss(1.0,0.1)
```

Example 8

```
.PARAM vtx=OPT1(.7,.3,1.0) uox=OPT1(650,400,900)
```

In this example, `uox` and `vtx` are the variable model parameters, which optimize a model for a selected set of electrical specifications.

The estimated initial value for the `vtx` parameter is 0.7 volts. You can vary this value within the limits of 0.3 and 1.0 volts for the optimization procedure. The optimization parameter reference name (OPT1) references the associated optimization analysis command (not shown).

Example 9

```
.PARAM fltmod=str('bpfmodel')
s1 n1 n2 n3 n_ref fqmodel=fltmod zo=50 fbase=25e6 fmax=1e9
```

This example shows how you can define and use string parameters.

See Also

.OPTION LIST

.PAT

Specifies predefined pattern names to be used in a pattern source; also defines new patnames.

Syntax

```
.PAT PatName=data [RB=val][R=int]
.PAT patName=[component 1 ... component n] [RB=val]
+ [R=int]
```

Arguments

Argument	Description
data	String of 1, 0, M, or Z that represents a pattern source. The first letter must be B to represent it as a binary bit stream. This series is called b-string. A 1 represents the high voltage or current value, and a 0 is the low voltage or current value. An M represents the value that is equal to 0.5*(vhi+vlo), and a Z represents the high impedance state (only for voltage source).
PatName	Pattern name that has an associated b-string or nested structure.
component	Elements that make up a nested structure. Components can be b-strings or a patname defined in other .PAT commands.
RB=val	Starting component of a repetition. The repeat data starts from the component or bit indicated by RB. RB must be an integer. If RB is larger than the length of the NS or b-string, an error is issued. If it is less than 1, it is automatically set to 1.
R=repeat	Number of times the repeating operation is executed. With no argument, the source repeats from the beginning of the NS or b-string. If R=-1, the repeating operation continues forever. R must be an integer. If it is less than -1, it automatically set to 0.

Description

When the .PAT command is used in an input file, some patnames are predefined and can be used in a pattern source. Patnames can associate a b-string or nested structure, two different types of pattern sources. In this case, a b-string is a series of 1, 0, m, and z states. The nested structure is a combination of a b-string and another netlisted structure defined in the .PAT

command. The .PAT command can also be used to define a new patname, which can be a b-string or nested structure.

Avoid using a predefined patname to define another patname to lessen the occurrence of a circular definition for which HSPICE issues an error report.

Nested structures must use brackets "[ ]", but HSPICE does not support using multiple brackets in one command. If you need to use another nested structure as a component, define it in a new .PAT command.

Example 1

The following example shows the .PAT command used for a b-string:

```
.PAT a1=b1010 r=1 rb=1
```

Example 2

The following example shows how an existing patname is used to define a new patname:

```
.PAT a1=b1010 r=1 rb=1
.PAT a2=a1
```

Example 3

The following example shows a nested structure:

```
.PAT a1=[b1010 r=1 rb=2 b1100]
```

Example 4

The following example shows how a predefined nested structure is used as a component in a new nested structure:

```
.PAT a1=[b1010 r=1 rb=2 b1100] r=1 rb=1
.PAT a2=[a1 b0m0m] r=2 rb=1
```

.PHASENOISE

Performs phase noise analysis on autonomous (oscillator) circuits in HSPICE RF.

Syntax

```
.PHASENOISE output frequency_sweep [method=0|1|2]
+ [carrierindex=int] [listfreq=(frequencies|none|all)]
+ [listcount=val] [listfloor=val] [listsources=on|off]
+ [spurious=0|1]
```

Arguments

Parameter	Description
output	Output node, pair of nodes, or 2-terminal element. HSPICE RF references phase noise calculations to this node (or pair of nodes). Specify a pair of nodes as V(n+,n-). If you specify only one node, V(n+), then HSPICE RF assumes that the second node is ground. You can also specify a 2-terminal element.
frequency_sweep	Sweep of type LIN, OCT, DEC, POI, or SWEEPBLOCK. Specify the type, nsteps, and start and stop time for each sweep type, where: ▪ type = Frequency sweep type, such as OCT, DEC, or LIN. ▪ nsteps = Number of steps per decade or total number of steps. ▪ start = Starting frequency. ▪ stop = Ending frequency. The four parameters determine the offset frequency sweep about the carrier used for the phase noise analysis. LIN *type nsteps start stop* OCT *type nsteps start stop* DEC *type nsteps start stop*POI *type nsteps start stop* SWEEPBLOCK *freq1 freq2 ... freqn*
METHOD	▪ METHOD=0 (default) selects the Nonlinear Perturbation (NLP) algorithm, which is used for low-offset frequencies. ▪ METHOD=1 selects the Periodic AC (PAC) algorithm, which is used for high-offset frequencies. ▪ METHOD=2 selects the Broadband Phase Noise (BPN) algorithm, which you can use to span low and high offset frequencies. You can use *METHOD* to specify any single method.

Parameter	Description
carrierindex	Harmonic index of the carrier at which HSPICE RF computes the phase noise (optional). The phase noise output is normalized to this carrier harmonic. The default is 1.
listfreq	Element phase noise value written to the *.lis* file. You can specify which frequencies the element phase noise value dumps. The frequencies must match the sweep_frequency values defined in the parameter_sweep, otherwise they are ignored. In the element phase noise output, the elements that contribute the largest phase noise are dumped first. The frequency values can be specified with the NONE or ALL keyword, which either dumps no frequencies or every frequency defined in the parameter_sweep. Frequency values must be enclosed in parentheses. For example: listfreq=(none) listfreq=(all) listfreq=(1.0G) listfreq=(1.0G, 2.0G) The default value is the first frequency value.
listcount	Dumps the element phase noise value to the .lis file, which is sorted from the largest to smallest value. You do not need to dump every noise element; instead, you can define listcount to dump the number of element phase-noise frequencies. For example, listcount=5 means that only the top 5 noise contributors are dumped. The default value is 20.
listfloor	Dumps the element phase noise value to the .lis file and defines a minimum meaningful noise value (in dBc/Hz units). Only those elements with phase-noise values larger than the *listfloor* value are dumped. For example, listfloor=-200 means that all noise values below -200 (dbc/Hz) are not dumped. The default value is -300 dbc/Hz.
listsources	Writes the element phase-noise value to the *.lis* file. When the element has multiple noise sources, such as a level 54 MOSFET, which contains the thermal, shot, and 1/f noise sources. When dumping the element phase-noise value you can decide if you need to dump the contribution from each noise source. You can specify either ON or OFF: ON dumps the contribution from each noise source and OFF does not. The default value is OFF.

Parameter	Description
spurious	Additional .HBAC analysis that predicts the spurious contributions to the phase noise. Spurs result from deterministic signals present within the circuit. In most cases, the spurs are very small signals and do not interfere with the steady-state operation of the oscillator but do add energy to the output spectrum of the oscillator. The energy that the spurs adds might need to be included in jitter measurements. 0 - No spurious analysis (default) 1 - Initiates a spurious noise analysis

Description

Use this command to invoke phase noise analysis on autonomous (oscillator) circuits.

See Also

.HB
.HBAC
.HBOSC
.SN
.SNAC
.SNOSC
.PRINT
.PROBE
Identifying Phase Noise Spurious Signals

.PKG

Provides the IBIS Package Model feature; automatically creates a series of W-elements or discrete R, L and C components.

Syntax

```
.PKG pkgname
+ file = 'pkgfilename'
+ model = 'pkgmodelname'
```

Arguments

Argument	Description
pkgname	Package card name
pkgfilename	Name of a *.pkg* or *.ibs* file that contains package models.
pkgmodelname	Working model in the *.pkg* file

Description

The `.PKG` command provides the IBIS Package Model feature. It supports both sections and matrixes.

The `.PKG` command automatically creates a series of W-elements or discrete R, L and C components. The following nodes are referenced in the netlist:

- Nodes on the die side:

  *'pkgname'*_*'pinname'*_dia

- Nodes on the pin side:

  *'pkgname'*_*'pinname'*

See Example 2 for how pin1 is referenced.

- If package = 0 in the .IBIS card, then no package information is added.
- If package = 1 or 2, then the package information in the *.ibs* file is added.
- If package = 3, then the package information in the *.pkg* file is added.

Example 1

```
.pkg p_test
+ file='processor_clk_ff.ibs'
+ model='FCPGA_FF_PKG'
```

Example 2

The following example shows how pin1 is referenced:

```
p_test_pin1_dia and p_test_pin1
```

The element name becomes:

```
w_p_test_pin1_? ? or r_p_test_pin1_? ? ...
```

See Also

.EBD
.IBIS

.POWER

Prints a table containing the AVG, RMS, MAX, and MIN measurements for specified signals in HSPICE RF.

Syntax

```
.POWER signal [REF=vname FROM=start_time TO=end_time]
```

Arguments

Argument	Description
signal	Signal name.
vname	Reference name.
start_time	Start time of power analysis period. You can also use parameters to define time.
end_time	End time of power analysis period. You can also use parameters to define time.

Description

Use this command to print a table containing the AVG, RMS, MAX, and MIN measurements for every signal specified.

By default, the scope of these measurements are set from 0 to the maximum timepoint specified in the `.TRAN` command.

For additional information, see POWER Analysis in the *HSPICE User Guide: RF Analysis*.

Example 1

In this example, no simulation start and stop time is specified for the x1.in signal, so the simulation scope for this signal runs from the start (0ps) to the last `.tran` time (100ps).

```
.power x1.in
.tran 4ps 100ps
```

Example 2

You can use the `FROM` and `TO` times to specify a separate measurement start and stop time for each signal. In this example:

- The scope for simulating the x2.in signal is from 20ps to 80ps.
- The scope for simulating the x0.in signal is from 30ps to 70ps.

```
.param myendtime=80ps
.power x2.in REF=a123 from=20ps to=80ps
.power x0.in REF=abc from=30ps to='myendtime - 10ps'
```

See Also

.TRAN
.OPTION SIM_POWER_ANALYSIS
.OPTION SIM_POWER_TOP
.OPTION SIM_POWERPOST
.OPTION SIM_POWERSTART
.OPTION SIM_POWERSTOP

.POWERDC

Calculates the DC leakage current in the design hierarchy.

Syntax

```
.POWERDC keyword subckt_name1...
```

Arguments

Argument	Description
keyword	One of these keywords: ■ TOP – prints the power for top-level instances ■ ALL (default) – prints the power for all instances
subckt_name#	Prints the power of all instances in this subcircuit definition

Description

Use this command to calculate the DC leakage current in the design hierarchy.

This option prints a table containing the measurements for AVG, MAX, and MIN values for the current of every instance in the subcircuit. This table also lists the sum of the power of each port in the subcircuit.

For additional information, see POWER Analysis in the *HSPICE User Guide: RF Analysis*.

You can use the `SIM_POWERDC_HSPICE` and `SIM_POWERDC_ACCURACY` options to increase the accuracy of the `.POWERDC` command.

See Also

.OPTION SIM_POWERDC_ACCURACY
.OPTION SIM_POWERDC_HSPICE

.PRINT

Prints the values of specified output variables.

Syntax

```
.PRINT antype ov1 [ov2 ... ]
```

Arguments

Argument	Description
antype	Type of analysis for outputs. Can be one of the following types: DC, AC, TRAN, NOISE, or DISTO.
ov1 ...	Output variables to print. These are voltage, current, or element template variables from a DC, AC, TRAN, NOISE, or DISTO analysis.

Description

Use this command to print the values of specified output variables. You can include wildcards in `.PRINT` commands. You can also use the `iall` keyword in a `.PRINT` command to print all branch currents of all diode, BJT, JFET, or MOSFET elements in your circuit design. By default, the `.PRINT` command prints out simulation data at a time interval of tstep of `.TRAN` command, so the number of points for this output data reported in the **.lis* are the "# points" shown at the end of **.lis* file.

Example 1

```
* CASE 1
.print v(din) i(mxn18)
.dc vdin 0 5.0 0.05
.tran 1ns 60ns
* CASE 2
.dc vdin 0 5.0 0.05
.tran 1ns 60ns
.print v(din) i(mxn18)
* CASE 3
.dc vdin 0 5.0 0.05
.print v(din) i(mxn18)
.tran 1ns 60ns
```

- If you replace the `.PRINT` command with:

  ```
  .print TRAN v(din) i(mnx)
  ```

  then all three cases have identical *.sw0* and *.tr0* files.

- If you replace the .print command with:

```
.print DC v(din) i(mnx)
```

  then the .sw0 and .tr0 files are different.

Example 2

```
.PRINT TRAN V (4) I(VIN) PAR(`V(OUT)/V(IN)')
```

This example prints the results of a transient analysis for the nodal voltage named 4. It also prints the current through the voltage source named VIN. It also prints the ratio of the nodal voltage at the OUT and IN nodes.

Example 3

```
.PRINT AC VM(4,2) VR(7) VP(8,3) II(R1)
```

- Depending on the value of the ACOUT option, VM(4,2) prints the AC magnitude of the voltage difference, or the difference of the voltage magnitudes between nodes 4 and 2.
- VR(7) prints the real part of the AC voltage between node 7 and ground.
- Depending on the ACOUT value, VP(8,3) prints the phase of the voltage difference between nodes 8 and 3, or the difference of the phase of voltage at node 8 and voltage at node 3.
- II(R1) prints the imaginary part of the current through R1.

Example 4

```
.PRINT AC ZIN YOUT(P) S11(DB) S12(M) Z11(R)
```

This example prints:

- The magnitude of the input impedance.
- The phase of the output admittance.
- Several S and Z parameters.

This command accompanies a network analysis by using the .AC and .LIN analysis commands.

Example 5

```
.PRINT DC V(2) I(VSRC) V(23,17) I1(R1) I1(M1)
```

This example prints the DC analysis results for several different nodal voltages and currents through:

- The resistor named R1.
- The voltage source named VSRC.
- The drain-to-source current of the MOSFET named M1.

Example 6

```
.PRINT NOISE INOISE
```

This example prints the equivalent input noise.

Example 7

```
.PRINT DISTO HD3 SIM2(DB)
```

This example prints the magnitude of third-order harmonic distortion, and the dB value of the intermodulation distortion sum through the load resistor that you specify in the .DISTO command.

Example 8

```
.PRINT AC INOISE ONOISE VM(OUT) HD3
```

This command includes NOISE, DISTO, and AC output variables in the same .PRINT command in HSPICE.

Example 9

```
.PRINT pj1=par(‘p(rd) +p(rs)‘)
```

This command prints the value of pj1 with the specified function.

HSPICE ignores .PRINT command references to nonexistent netlist part names, and prints those names in a warning.

Example 10

Derivative function:

```
.PRINT der=deriv('v(NodeX)')
```

Integrate function:

```
.PRINT int=integ('v(NodeX)')
```

The parameter can be a node voltage or a reasonable expression.

Example 11

```
.param p1=3

.print par('p1')
.print p2=par("p1*5")
```

You can use `p1` and `p2` as parameters in netlist. The `p1` value is 3; the `p2` value is 15.

See Also

.AC
.DC
.DCMATCH
.DISTO
.DOUT
.MEASURE (or) .MEAS
.NOISE
.PROBE
.STIM
.TRAN

.PROBE

Saves output variables to interface and graph data files.

Syntax

```
.PROBE antype ov1 [ov2 ...]
```

Arguments

Argument	Description
antype	Type of analysis for the specified plots. Analysis types are: DC, AC, TRAN, NOISE, or DISTO for HSPICE; ENV, HB, HBAC, HBLSP, HBNOISE, HBTR, HBTRAN, HBXF, NOISE, or PHASENOISE for HSPICE RF.
ov1 ...	Output variables to plot: voltage, current, or element template (HSPICE-only variables from a DC, DCMATCH, AC, ACMATCH, TRAN, NOISE, or DISTO analysis. .PROBE can include more than one output variable. HSPICE RF analyses include: ENV, HB, HBAC, HBLSP, HBNOISE, HBTR, HBTRAN, HBXF, NOISE, or PHASENOISE analysis

Description

Use this command to save output variables to interface and graph data files. The parameter can be a node voltage or a reasonable expression. You can include wildcards in `.PROBE` commands. The `.PROBE` command outputs the signals to waveform files no matter how `.OPTION PROBE` and `.OPTION PUTMEAS` are set.

Note:

For AC analysis in HSPICE, only the magnitude is saved to the waveform file unless a complex quantity is explicitly specified.

Example 1

```
.PROBE DC V(4) V(5) V(1) beta=PAR(`I1(Q1)/I2(Q1)')
```

Example 2

```
* Derivative function
.PROBE der=deriv('v(NodeX)')
* Integrate function
.PROBE int=integ('v(NodeX)')
```

See Also

.AC
.ACMATCH
.DC
.DCMATCH
.DISTO
.DOUT
.ENV
.HB
.HBAC
.HBLSP
.HBNOISE
.HBOSC
.HBXF
.MEASURE (or) .MEAS
.NOISE
.PHASENOISE
.PRINT
.STIM
.TRAN
.OPTION PROBE
.OPTION PUTMEAS

.PROTECT or .PROT

Keeps models and cell libraries private as part of the encryption process in HSPICE.

Syntax

```
.PROTECT
```

Description

Use this command to designate the start of the file section to be encrypted when using Metaencrypt.

- Use `.UNPROTECT` to end the file section that will be encrypted.
- Any elements and models located between a `.PROTECT` and an `.UNPROTECT` command inhibit the element and model listing from the `LIST` option.
- The `.OPTION NODE` nodal cross-reference and the `.OP` operating point printout do not list any nodes that are contained between the `.PROTECT` and `.UNPROTECT` commands.

Note:

If you use `.prot/.unprot` in a library or file that is not encrypted you might get warnings that the file is encrypted and the file or library is treated as a "black box."

The .prot and .unprot commands act similar to `.option brief=1` and `.option brief=0`, respectively.

See Also

.UNPROTECT or .UNPROT
.OPTION BRIEF

.PTDNOISE

Calculates the noise spectrum and the total noise at a point in time for HSPICE RF.

Syntax

```
.PTDNOISE [output] [time_value] [time_delta]
+ [frequency_sweep]
+ [listfreq=(frequencies|none|all)] [listcount=val]
+ [listfloor=val] [listsources=on|off]
```

Arguments

Parameter	Description
output	Output node, pair of nodes, or 2-terminal element. HSPICE RF references the equivalent noise output to this node (or pair of nodes). Specify a pair of nodes as V(n+,n-). If you specify only one node V(n+), then HSPICE RF assumes the second node is ground. You can also specify a 2-terminal element name that refers to an existing element in the netlist.
time_value	Time point at which time domain noise is evaluated. Specify either a time point explicitly, such as: TIME=value, where value is either numerical or a parameter name or A .MEASURE name associated with a time domain .MEASURE command located in the netlist. PTDNOISE uses the time point generated from the .MEASURE command to evaluate the noise characteristics. This is useful if you want to evaluate noise or jitter when a signal reaches some threshold value.
time_delta	A time value used to determine the slew rate of the time-domain output signal. Specified as TDELTA=value. The signal slew rate is then determined by the output signal at TIME +/- TDELTA and dividing this difference by 2 x TDELTA. This slew rate is then used in the calculation of the strobed jitter. If this term is omitted a default value of 0.01 x the .SN period is assumed.

Parameter	Description
frequency_sweep	Frequency sweep range for the output noise spectrum. The upper and lower limits also specify the integral range in calculating the integrated noise value. Specify LIN,DEC, OCT, POI, SWEEPBLOCK, or DATA sweeps. Specify the nsteps, start, and stop frequencies using the following syntax for each type of sweep: ▪ LIN *nsteps start stop* ▪ DEC*nsteps start stop* ▪ OCT *nsteps start stop* ▪ POI *nsteps freq_values* ▪ SWEEPBLOCK *nsteps freq1 freq2 ... freqn* ▪ DATA *dataname*

listfreq	Element noise value printed to the *.lis* file. This information is only printed if a noise spectrum is requested in a PRINT or PROBE command. You can specify which frequencies the element noise is printed. The frequencies must match the sweep_frequency values defined in the *frequency_sweep*, otherwise they are ignored.
	In the element noise output, the elements that contribute the largest noise are printed first. The frequency values can be specified with the NONE or ALL keyword, which either prints no frequencies or every frequency defined in *frequency_sweep*. Frequency values must be enclosed in parentheses. For example:
	`listfreq=(none)`
	`listfreq=(all)`
	`listfreq=(1.0G)`
	`listfreq=(1.0G, 2.0G)`
	The default value is NONE.

listcount	Element noise value printed to the *.lis* file, which is sorted from the largest to smallest value. You do not need to print every noise element; instead, you can define `listcount` to print the number of element noise frequencies. For example, `listcount=5` means that only the top 5 noise contributors are printed. The default value is 1.
listfloor	Element noise value printed to the *.lis* file that defines a minimum meaningful noise value (in $V/Hz^{1/2}$ units). Only those elements with noise values larger than `listfloor` are printed. The default value is 1.0e-14 $V/Hz^{1/2}$.

Parameter	Description
listsources	Element noise value printed to the *.lis* file when the element has multiple noise sources, such as a FET, which contains the thermal, shot, and 1/f noise sources. You can specify either ON or OFF: ON prints the contribution from each noise source and OFF does not. The default value is OFF.

Description

Periodic Time-Dependent noise analysis (PTDNOISE) calculates the noise spectrum and the total noise at a point in time. Jitter in a digital threshold circuit can then be determined from the total noise and the digital signal slew rate. .MEASURE PTDNOISE allows for the measurement of these parameters: integnoise, time-point, tdelta-value, slewrate, and strobed jitter. See Periodic Time-Dependent Noise Analysis (.PTDNOISE) in the *HSPICE User Guide: RF Analysis* for details.

See Also

.HBNOISE
.SNNOISE

.PZ

Performs pole/zero analysis.

Syntax

```
.PZ output input
.PZ ov srcname
```

Arguments

Argument	Description
input	Input source; the name of any independent voltage or current source.
output	Output variables, which can be: ■ Any node voltage, V(*n*). ■ Any branch current, I(*branch_name*).
ov	Output variable: ■ a node voltage V(*n*), or a branch current I(*element*)
srcnam	Input source: ■ an independent voltage or a current source name

Description

Use to perform Pole/Zero analysis. You do not need to specify `.OP` because the simulator automatically invokes an operating point calculation. See Pole/Zero Analysis in the *HSPICE User Guide: Simulation and Analysis* for complete information about pole/zero analysis.

Example

```
.PZ   V(10)   VIN
.PZ   I(RL)   ISORC
```

- In the first pole/zero analysis, the output is the voltage for node 10 and the input is the `VIN` independent voltage source.
- In the second pole/zero analysis, the output is the branch current for the RL branch and the input is the `ISORC` independent current source.

See Also

.DC

.SAMPLE

Analyzes data sampling noise.

Syntax

```
.SAMPLE FS=freq [TOL=val] [NUMF=val]
+ [MAXFLD=val] [BETA=0|1]
```

Arguments

Argument	Description
FS=freq	Sample frequency in hertz.
TOL	Sampling-error tolerance: the ratio of the noise power (in the highest folding interval) to the noise power (in baseband). The default is 1.0e-3.
NUMF	Maximum number of frequencies that you can specify. The algorithm requires about ten times this number of internally-generated frequencies so keep this value small. The default is 100.
MAXFLD	Maximum number of folding intervals (The default is 10.0). The highest frequency (in hertz) that you can specify is: FMAX=MAXFLD · FS
BETA	Optional noise integrator (duty cycle) at the sampling node: ▪ BETA=0 no integrator ▪ BETA=1 simple integrator (default) If you clock the integrator (integrates during a fraction of the 1/FS sampling interval), then set BETA to the duty cycle of the integrator.

Description

Use this command to acquire data from analog signals. It is used with the .NOISE and .AC commands to analyze data sampling noise in HSPICE. The SAMPLE analysis performs a noise-folding analysis at the output node.

See Also

.AC
.NOISE

.SAVE

Stores the operating point of a circuit in a file that you specify in HSPICE only.

Syntax

```
.SAVE [TYPE=type_keyword] [FILE=save_file]
+ [LEVEL=level_keyword] [TIME=save_time]
```

Arguments

Argument	Description
type_keyword	Storage method for saving the operating point. The type can be one of the following. Default is NODESET. ▪ NODESET: Stores the operating point as a NODESET command. Later simulations initialize all node voltages to these values if you use the .LOAD command. If circuit conditions change incrementally, DC converges within a few iterations. ▪ IC: Stores the operating point as an IC command. Later simulations initialize node voltages to these values if the netlist includes the .LOAD commands.
save_file	Name of the file that stores DC operating point data. The file name format is <*design*>.ic#. Default is <*design*>.ic0.
level_keyword	Circuit level at which you save the operating point. The level can be one of the following. ▪ ALL (default): Saves all nodes from the top to the lowest circuit level. This option offers the greatest improvement in simulation time. ▪ TOP: Saves only nodes in the top-level design. Does not save subcircuit nodes. ▪ NONE: Does not save the operating point.
save_time	Time during transient analysis when HSPICE saves the operating point. HSPICE requires a valid transient analysis command to save a DC operating point. The default is 0.

Description

Use this command to store the operating point of a circuit in a file that you specify. For quick DC convergence in subsequent simulations, use the .LOAD command to input the contents of this file. HSPICE saves the operating point by default, even if the HSPICE input file does not contain a .SAVE command. To

not save the operating point, specify `.SAVE LEVEL=NONE`. You can save the operating point data as either an `.IC` or a `.NODESET` command. A parameter or temperature sweep saves only the first operating point.

The `.SAVE` command only saves one bias point to a file.

Note:

To save multiple node voltages at different timesteps, it is preferable to use the `.OP` command.

Example

```
.TEMP -25 0 25
.SAVE TYPE=NODESET FILE=my_design.ic0 LEVEL=ALL
+ TIME=0
```

This example saves the operating point corresponding to `.TEMP -25` to a file named *my_design.ic0*.

See Also

.IC
.LOAD
.NODESET
.OP

.SENS

Determines DC small-signal sensitivities of output variables for circuit parameters.

Syntax

`.SENS` *ov1* [*ov2* ...]

Arguments

Argument	Description
ov1 ov2 ...	Branch currents or nodal voltage for DC component-sensitivity analysis

Example

In this example, the `.SENS v(2)` command is used to find out how sensitive the voltage at node 2 is to change at any element value.

```
v1 1 0 1
r1 1 2 1k
r2 2 0 1k
.SENS v(2)
.end
```

For sensitivity analysis only one element is changed at a time while all other element values are retained at their original value. The output of the `.SENS v(2)` command appears in the list file as follows:

```
dc sensitivities of output v(2)

        element          element        element        normalized
          name             value      sensitivity     sensitivity
                                   (volts/unit)  (volts/percent)

        0:r1             1.0000k    -250.0000u       -2.5000m
        0:r2             1.0000k     250.0000u        2.5000m
        0:v1             1.0000      500.0000m        5.0000m
```

The element sensitivity column lists the absolute change in V(2) when the element value is changed by unity. As shown, an element sensitivity of -250.0000u for element r1 indicates that v(2) decreases by 250uv when R1 is increased from 1000 ohms to 1001 ohms. Similarly, an element sensitivity of 500.0000m for element v1 indicates that v(2)increases by 500mv when v1 increases by 1V.

The normalized sensitivity column lists the absolute change in v(2) when the element value is increased by 1%. As shown for element r1, the normalized sensitivity of -2.5000m indicates that v(2) decreases by 2.5mv when the value of r1 is increased by 1%.

Note:

In both columns, a negative sign indicates a decrease and a positive sign indicates an increase in the output variable (in this case, v(2)).

Description

Use this command to determine DC small-signal sensitivities of output variables for circuit parameters.

If the input file includes a `.SENS` command, HSPICE determines DC small-signal sensitivities for each specified output variable relative to every circuit parameter. The sensitivity measurement is the partial derivative of each output variable for a specified circuit element measured at the operating point and normalized to the total change in output magnitude. Therefore, the sum of the sensitivities of all elements is 100%. DC small-signal sensitivities are calculated for:

- resistors
- voltage sources
- current sources
- diodes
- BJTs (including Level 4, the VBIC95 model)
- MOSFETs (Level49 and Level53, Version=3.22).

You can perform only one `.SENS` analysis per simulation. Only the last `.SENS` command is used in case more than one is present. The others are discarded with warnings.

The amount of output generated from a `.SENS` analysis is dependent on the size of the circuit.

See Also

.DC

.SHAPE

Defines a shape to be used by the HSPICE field solver.

Syntax

```
.SHAPE sname Shape_Descriptor
```

Arguments

Argument	Description
sname	Shape name.
Shape_Descriptor	One of the following: ■ Rectangle ■ Circle ■ Strip ■ Polygon

Description

Use this command to define a shape. The field solver uses the shape to describe a cross-section of the conductor.

See Also

.SHAPE (Defining Rectangles)
.SHAPE (Defining Circles)
.SHAPE (Defining Polygons)
.SHAPE (Defining Strip Polygons)
.FSOPTIONS
.LAYERSTACK
.MATERIAL

.SHAPE (Defining Rectangles)

Defines a rectangle to be used by the HSPICE field solver.

Syntax

```
.SHAPE RECTANGLE WIDTH=val HEIGHT=val [NW=val] [NH=val]
```

Arguments

Argument	Description
WIDTH	Width of the rectangle (size in the x-direction).
HEIGHT	Height of the rectangle (size in the y-direction).
NW	Number of horizontal (x) segments in a rectangle with a specified width.
NH	Number of vertical (y) segments in a rectangle with a specified height.

Description

Use this keyword to define a rectangle. Normally, you do not need to specify the `NW` and `NH` values because the field solver automatically sets these values, depending on the accuracy mode. You can specify both values or only one of these values and let the solver determine the other.

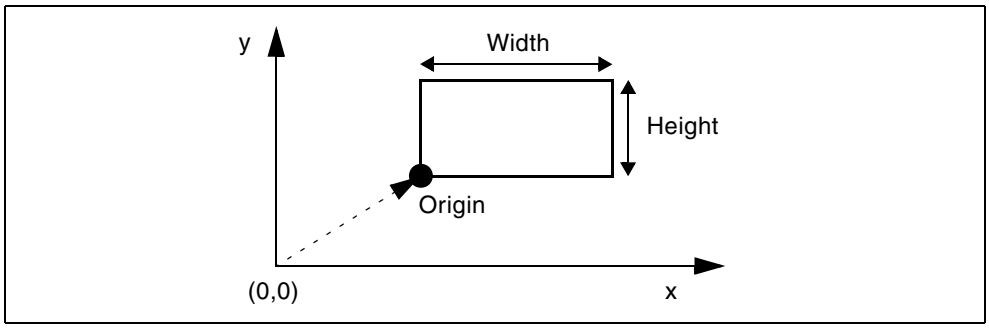

Figure 9 Coordinates of a Rectangle

.SHAPE (Defining Circles)

Defines a circle to be used by the HSPICE field solver.

Syntax

```
.SHAPE CIRCLE RADIUS=val [N=val]
```

Arguments

Argument	Description
RADIUS	Radius of the circle.
N	Number of segments to approximate a circle with a specified radius.

Description

Use this keyword to define a circle in the field solver. The field solver approximates a circle as an inscribed regular polygon with *N* edges. The more edges, the more accurate the circle approximation is.

Do not use the `CIRCLE` descriptor to model actual polygons; instead use the `POLYGON` descriptor.

Normally, you do not need to specify the N value because the field solver automatically sets this value, depending on the accuracy mode. But you can specify this value if you need to

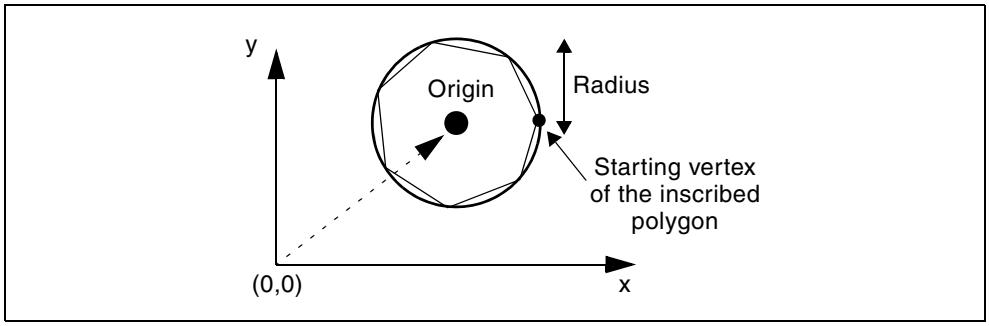

Figure 10 Coordinates of a Circle

.SHAPE (Defining Polygons)

Defines a polygon to be used by the HSPICE field solver.

Syntax

```
.SHAPE POLYGON VERTEX=(x1 y1 x2 y2 ...)
+ [N=(n1,n2,...)]
```

Arguments

Argument	Description
VERTEX	(*x*, *y*) coordinates of vertices. Listed either in clockwise or counter-clockwise direction.
N	Number of segments that define the polygon with the specified *x* and *y* coordinates. You can specify a different *N* value for each edge. If you specify only one *N* value, then the field solver uses this value for *all* edges. For example, the first value of *N, n1*, corresponds to the number of segments for the edge from (*x1 y1*) to (*x2 y2*).

Description

Use this command to define a polygon in a field solver. The specified coordinates are within the local coordinate with respect to the origin of a conductor.

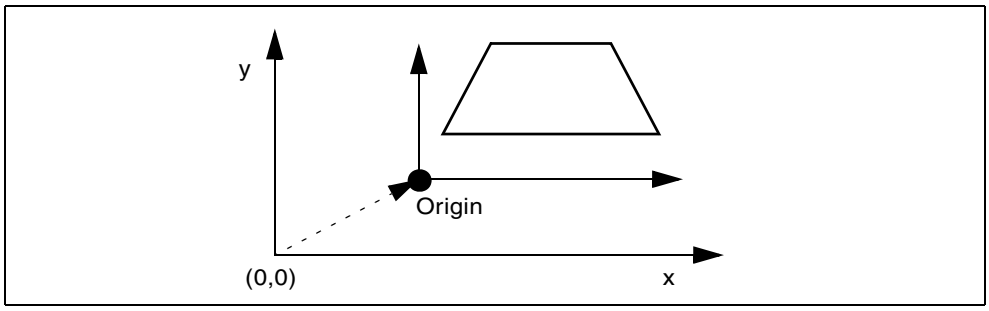

Figure 11 Coordinates of a Polygon

Example 1

The following rectangular polygon uses the default number of segments:

```
.SHAPE POLYGON VERTEX=(1 10 1 11 5 11 5 10)
```

Example 2

The following rectangular polygon uses five segments for each edge:

```
.SHAPE POLYGON VERTEX=(1 10 1 11 5 11 5 10)
+ N=5
```

Example 3

Rectangular polygon uses different number of segments for each edge:

```
.SHAPE POLYGON VERTEX=(1 10 1 11 5 11 5 10)
+ N=(5 3 5 3)
```

.SHAPE (Defining Strip Polygons)

Defines a strip polygon to be used by the HSPICE field solver.

Syntax

```
.SHAPE STRIP WIDTH=val [N=val]
```

Arguments

Argument	Description
WIDTH	Width of the strip (size in the x-direction).
N	Number of segments that define the strip shape with the specified width.

Description

Use this command to define a strip polygon in a field solver. Normally, you do not need to specify the `N` value because the field solver automatically sets this value, depending on the accuracy mode. But you can specify this value if you need to.

The field solver (filament method) does not support this shape.

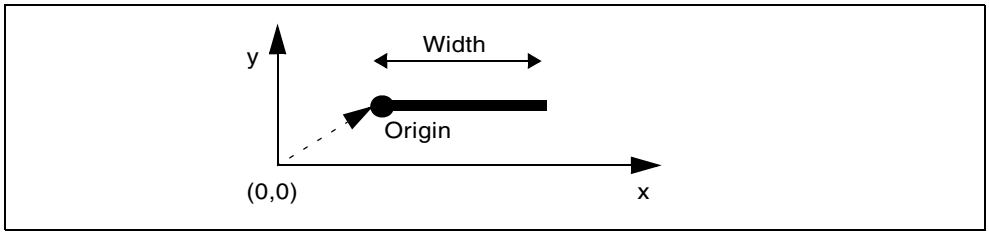

Figure 12 Coordinates of a Strip Polygon

.SN

In HSPICE RF, Shooting Newton provides two syntaxes. Syntax #1 is recommended when you are using/making Time Domain sources and measurements (for example, going from .TRAN to .SN). Syntax #2 effectively supports Frequency Domain sources and measurements (and should be used, for example, when going from .HB to .SN).

Syntax

Syntax #1

```
.SN TRES=Tr PERIOD=T [TRINIT=Ti]
+ [SWEEP parameter_sweep][MAXTRINITCYCLES=integer]
```

Syntax #2

```
.SN TONE=<F1> NHARMS=N [TRINIT=Ti]
+ [SWEEP parameter_sweep] [MAXTRINITCYCLES=integer]
```

Arguments

Parameter	Description
TRES	Time resolution to be computed for the steady-state waveforms (in seconds).
PERIOD	Expected period T (seconds) of the steady-state waveforms. Enter an approximate value when using for oscillator analysis. The period of the steady-state waveform may be entered either as PERIOD or its reciprocal, TONE.
TRINIT	Transient initialization time. If not specified, the transient initialization time will be equal to the period (for Syntax 1) or the reciprocal of the tone (for Syntax 2).
SWEEP	Parameter sweep. As in all main analyses in HSPICE RF such as .TRAN, .HB, etc., you can specify LIN, DEC, OCT, POI, SWEEPBLOCK, DATA, MONTE, or OPTIMIZE.
MAXTRINITCYCLES	SN stabilization simulation and frequency detection is stopped when the simulator detects that maxtrinitcycles have been reached in the oscnode signal, or when time=trinit, whichever comes first. Minimum cycles is 1.

Parameter	Description
TONE	Fundamental frequency (in Hz).
NHARMS	Specifies the number of high-frequency harmonic components to include in the analysis. NHARMS defaults to PERIOD/TRES rounded to nearest integer. NHARMS is required to run subsequent SNAC, SNNOISE, SNXF, and PHASENOISE analyses. When using Syntax #1, NHARMS is computed automatically as NHARMS=Round(PERIOD/TRES).

Description

Shooting-Newton adds analysis capabilities for PLL components, digital circuits/logic, such as ring oscillators, frequency dividers, phase/frequency detectors (PFDs), and for other digital logic circuits and RF components that require steady-state analysis, but operate with waveforms that are more square wave than sinusoidal. Refer to the *HSPICE User Guide: RF Analysis*, Steady-State Shooting Newton Analysis.

Options

In addition to all .TRAN options, .SN analysis supports the following options:

- `.OPTION LOADSNINIT`
- `.OPTION SAVESNINIT`
- `.OPTION SNACCURACY`
- `.OPTION SNMAXITER`

See Also

.SNAC
.SNFT
.SNNOISE
.SNOSC
.SNXF
.OPTION LOADSNINIT
.OPTION SAVESNINIT
.OPTION SNACCURACY
.OPTION SNMAXITER

.SNAC

Runs a frequency sweep across a range for the input signal based on a Shooting Newton algorithm.

Syntax

```
.SNAC frequency_sweep
```

Arguments

Parameter	Description
frequency_sweep	Frequency sweep range for the input signal (also referred to as the input frequency band (IFB) or fin). You can specify LIN, DEC, OCT, POI, or SWEEPBLOCK. Specify the nsteps, start, and stop times using the following syntax for each type of sweep: ■ LIN *nsteps start stop* ■ DEC *nsteps start stop* ■ OCT *nsteps start stop* ■ POI *nsteps freq_values* ■ SWEEPBLOCK *nsteps freq1 freq2 ... freqn*

Description

The *frequency_sweep* runs across a range for the input signal based on a Shooting Newton algorithm. For more information, see Shooting Newton AC Analysis (.SNAC) in the *HSPICE User Guide: RF Analysis*.

Example

```
VSRC node1 node2 0 SNAC 1 45
.SNAC DEC 10 1k 10K
```

See Also

.HBAC
.SN
.SNNOISE

.SNFT

Calculates the Discrete Fourier Transform (DFT) value used for Shooting Newton analysis. Numerical parameters (excluding string parameters) can be passed to the .SNFT command.

Syntax

Syntax # 1 Alphanumeric input

```
.SNFT output_var [START=value] [STOP=value]
+ [NP=value] [FORMAT=keyword]
+ [WINDOW=keyword] [ALFA=value]
+ [FREQ=value] [FMIN=value] [FMAX=value]
```

Syntax #2 Numerics and expressions

```
.SNFT output_var [START=param_expr1] [STOP=param_expr2]
+ [NP=param_expr3] [FORMAT=keyword]
+ [WINDOW=keyword] [ALFA=param_expr4]
+ [FREQ=param_expr5] [FMIN=param_expr6] [FMAX=param_expr7]
```

Arguments

Argument	Description
output_var	Any valid output variable, such as voltage, current, or power.
START	Start of the output variable waveform to analyze. Defaults to the START value in the .SN command, which defaults to 0.
FROM	Alias for START in .SNFT commands.
STOP	End of the output variable waveform to analyze. Defaults to the TSTOP value in the .SN command.
TO	Alias for STOP, in .SNFT commands.
NP	Number of points to use in the SNFT analysis. NP must be a power of 2. If NP is not a power of 2, HSPICE automatically adjusts it to the closest higher number that is a power of 2. The default is 1024.
FORMAT	Output format: ▪ NORM= normalized magnitude (default) ▪ UNORM=unnormalized magnitude

Argument	Description
WINDOW	Window type to use: ■ RECT=simple rectangular truncation window (default). ■ BART=Bartlett (triangular) window. ■ HANN=Hanning window. ■ HAMM=Hamming window. ■ BLACK=Blackman window. ■ HARRIS=Blackman-Harris window. ■ GAUSS=Gaussian window. ■ KAISER=Kaiser-Bessel window.
ALFA	Parameter to use in GAUSS and KAISER windows to control the highest side-lobe level, bandwidth, and so on. 1.0 <= ALFA <= 20.0 The default is 3.0
FREQ	Frequency to analyze. If FREQ is non-zero, the output lists only the harmonics of this frequency, based on FMIN and FMAX. HSPICE also prints the THD for these harmonics. The default is 0.0 (Hz).
FMIN	Minimum frequency for which HSPICE prints SNFT output into the listing file. THD calculations also use this frequency. T=(STOP-START) The default is 1.0/T (Hz).
FMAX	Maximum frequency for which HSPICE prints SNFT output into the listing file. THD calculations also use this frequency. The default is 0.5*NP*FM IN (Hz).

Description

Use this command to calculate the Discrete Fourier Transform (DFT) spectrum analysis values for Shooting Newton analysis. It uses internal time point values to calculate these values. A DFT uses sequences of time values to determine the frequency content of analog signals in circuit simulation. You can pass numerical parameters/expressions (but no string parameters) to the `.SNFT` command. The output goes to a file with extension *.snft#*.

You can specify only one output variable in an `.SNFT` command. The following is an incorrect use of the command because it contains two variables in one `.SNFT` command:

```
.SNFT v(1) v(2) np=1024
```

Example 1

```
.SNFT v(1)
.SNFT v(1,2) np=1024 start=0.3m stop=0.5m freq=5.0k
+ window=kaiser alfa=2.5
.SNFT I(rload) start=0m to=2.0m fmin=100k fmax=120k
+ format=unorm
.SNFT par('v(1) + v(2)') from=0.2u stop=1.2u
+ window=harris
```

The example above correctly designates the variables per `.SNFT` command.

Example 2

```
.SNFT v(1) np=1024
.SNFT v(2) np=1024
```

This example generates a .snft0 file for the SNFT of v(1) and a .snft1 file for the SNFT of v(2).

See Also

.SN

.SNNOISE

Runs a periodic, time-varying AC noise analysis based on a Shooting Newton algorithm.

Syntax

```
.SNNOISE [output] [insrc] [frequency_sweep]
+ [n1, +/-1]
+[listfreq=(frequencies|none|all)> [listcount=val]
+[listfloor=val] [listsources=on|off]
```

Arguments

Argument	Description
output	Output node, pair of nodes, or 2-terminal element that the equivalent noise output references.
insrc	Input source.
frequency_sweep	Frequency sweep range for the input signal. You can specify LIN, DEC, OCT, POI, SWEEPBLOCK, DATA, MONTE, or OPTIMIZE sweeps.
n1, +/-	Index term defining the output frequency band at which the noise is evaluated. The output frequency is computed according to fout=|n1*f1 +/- fin|, where f1 is the fundamental tone (inverse of fundamental period) and fin is from the frequency sweep.
listfreq	Prints the element noise value to the *.lis* file; the default is none.
listcount	Prints the element noise value to the *.lis* file, sorted from the largest to smallest value.
listfloor	Prints the element noise value to the *.lis* file and defines a minimum meaningful noise value. Only those elements with noise values larger than listfloor are printed. The default value is 1.0e-14 V/sqrt(Hz).
listsources	Prints the element noise value to the *.lis* file when the element has multiple noise sources. The default is off.

Description

The functionality for the .SNNOISE command to is similar to the Harmonic Balance (HBNOISE command) for periodic, time-varying AC noise analysis, but

the Shooting Newton-based algorithm completes the analysis in a much faster run time with the same result.

Example

```
.SNNOISE V(n1,n2) RIN DEC 10 1k 10k 0 -1
```

See Also

.HBNOISE
.SN
.SNAC

.SNOSC

Performs oscillator analysis on autonomous (oscillator) circuits. As with regular Shooting Newton analysis, input might be specified in terms of time or frequency values.

Syntax #1

```
.SNOSC TONE=F1 NHARMS=H1 [TRINIT=Ti] OSCNODE=N1
+[MAXTRINITCYCLES=N][SWEEP PARAMETER_SWEEP]
```

Syntax #2

```
.SNOSC TRES=Tr PERIOD=Tp [TRINIT=Tr] OSCNODE=N1
+[MAXTRINITCYCLES=I] SWEEP PARAMETER_SWEEP
```

Arguments

Parameter	Description
TONE	Approximate value for oscillation frequency (Hz). The search for an exact oscillation frequency begins from this value.
NHARMS	Number of harmonics to be used for oscillator SN analysis.
OSCNODE	Node used to probe for oscillation conditions. This node is automatically analyzed to search for periodic behavior near the TONE or PERIOD value specified.
TRINIT	Transient initialization time. If not specified, the transient initialization time is equal to the period (for Syntax 1) or the reciprocal of the tone (for Syntax 2). For oscillators we recommend specifying a transient initialization time since the default initialization time is usually too short to effectively stabilize the circuit.
MAXTRINITCYCLES	SN stabilization simulation and frequency detection is stopped when the simulator detects that MAXTRINITCYCLES have been reached in the `oscnode` signal, or when time=`trinit`, whichever comes first. Minimum cycles is 1.

Parameter	Description
TRES	Time resolution to be computed for the steady-state waveforms (in seconds). The period of the steady-state waveform may be entered either as `PERIOD` or its reciprocal, `TONE`.
PERIOD	Expected period T (seconds) of the steady-state waveforms. Enter an approximate value when using for oscillator analysis.
SWEEP	Type of sweep. You can sweep up to three variables. You can specify either LIN, DEC, OCT, POI, SWEEPBLOCK, DATA, OPTIMIZE, or MONTE. Specify the nsteps, start, and stop frequencies using the following syntax for each type of sweep: ▪ LIN *nsteps start stop* ▪ DEC *nsteps start stop* ▪ OCT *nsteps start stop* ▪ POI *nsteps freq_values* ▪ SWEEPBLOCK *nsteps freq1 freq2 ... freqn* ▪ DATA=*dataname* ▪ OPTIMIZE=OPT*xxx* ▪ MONTE=*val*

Description

Use this command to invoke oscillator analysis on autonomous (oscillator) circuits. The SNOSC command is very effective for ring oscillator circuits, and oscillators that operate with piecewise linear waveforms (HBOSC is superior for sinusoidal waveforms). As with the Harmonic Balance approach, the goal is to solve for the additional unknown oscillation frequency. This is accomplished in Shooting Newton by considering the period of the waveform as an additional unknown, and solving the boundary conditions at the waveform endpoints that coincide with steady-state operation. As with regular Shooting Newton analysis, input might be specified in terms of time or frequency values. See the examples, below.

Example 1

```
.SNOSC tone=900Meg nharms=9 trinit=10n oscnode=gate
```

Performs an oscillator analysis, searching for periodic behavior after an initial transient analysis of 10 ns. This example uses nine harmonics while searching for a oscillation at the gate node.

Example 2

```
.SNOSC tone=2400MEG nharms=11 trinit=20n oscnode=drainP
```

Performs an oscillator analysis, searching for frequencies in the vicinity of 2.4 Ghz. This example uses 11 harmonics and a search at the drainP.

Example 3

Another equivalent method to define the OSCNODE information is through a zero-current source.

```
ISRC drainP 0 SNOSCVPROBE
.SNOSC tone = 2.4 G nharms = 1 trinit=20n
```

Example 3 is identical to Example 2, except that the OSCNODE information is defined by a current source in the circuit. Only one such current source is needed and its current source must be 0.0 with the SNOSC OSCNODE identified by the SNOSCVPROBE keyword.

See Also

.HB
.OPTION HBFREQABSTOL
.OPTION HBFREQRELTOL
.OPTION HBMAXOSCITER
.OPTION HBPROBETOL
.OPTION HBTRANFREQSEARCH
.OPTION HBTRANINIT
.OPTION HBTRANPTS
.OPTION HBTRANSTEP
.PRINT
.PROBE

.SNXF

Calculates the transfer function from the given source in the circuit to the designated output.

Syntax

`.SNXF` *`out_var freq_sweep`*

Arguments

Parameter	Description
out_var	`I(2_`*`port_elem`*`)` or `V(`*`n1`*`<,`*`n2`*`>)`
freq_sweep	Sweep of type LIN, DEC, OCT, POI, or SWEEPBLOCK. Specify the nsteps, start, and stop times using the following syntax for each type of sweep: ▪ LIN *nsteps start stop* ▪ DEC *nsteps start stop* ▪ OCT *nsteps start stop* ▪ POI *nsteps freq_values* ▪ SWEEPBLOCK = *BlockName* Specify the frequency sweep range for the output signal. HSPICE RF determines the offset frequency in the input sidebands; for example, f1 = abs(fout - k*f0) s.t. f1<=f0/2 The f0 is the steady-state fundamental tone and f1 is the input frequency.

Description

Use this command in HSPICE RF to calculate the transfer function from the given source in the circuit to the designated output. The functionality for the `.SNXF` command is similar to the Harmonic Balance (`.HBXF`) command for periodic, time-varying AC noise analysis, but the Shooting Newton based algorithm completes the analysis in a much faster run time with the same result.

Example

In this example, the trans-impedance from `isrc` to `v(1)` is calculated based on the HB analysis.

```
.hb tones=1e9 nharms=4
.snxf  v(1) lin 10 1e8 1.2e8
.print snxf tfv(isrc)  tfi(n3)
```

See Also

.HB
.HBAC
.HBNOISE
.HBOSC
.PRINT
.PROBE

.STATEYE

Enables use of statistical eye diagram analysis.

Syntax

```
.STATEYE T=time_interval Trf=rise_fall_time
+ Incident_port=idx1, [idx2, … idxN]
+ Probe_port=idx1, [idx2, … idxN]
+ [Rj=Rj1, [Rj2, … RjN]]
+ [tran_init=n_periods]
+ [V_low=val] [V_high=val]
+ [T_resolution=n] [V_resolution=n]
```

Parameter	Description
T	Time of one period of the incident pulse signal (in seconds).
Trf	Single value (in seconds) to set both the rise and fall times of the incident pulse.
incident_port	Array of the index numbers of the incident port elements.
probe_port	Array of the index numbers of the probing port elements.
Rj	Array of the real numbers to specify the standard deviation of the Gaussian random jitter. The array must be in the order of the port element index. No random jitter is added by default.
V_low	Low voltage level of the incident pulse. The value is used when the voltage level is not specified in the incident port(s).
V_high	High voltage level of the incident pulse. The value is used when the voltage level is not specified in the incident port(s).
tran_init	Integer number that specifies the numbers of periods (T) that is used by the initial transient analysis to determine the pulse response of the system. Default value is 30.
T_resolution	Integer number used to specify the probability density function (PDF) image resolution of the time axis. Default value is 200.
V_resolution	Integer number used to specify the probability density function (PDF) image resolution of the voltage axis. Default value is 200.

Description

Use this command to perform statistical eye analysis to evaluate high-speed serial interfaces.

Important:

The `.STATEYE` command is invoked using the `hspicerf` executable on the command line for this release (2008.09), not `hspice`. However, running a statistical eye analysis only requires an HSPICE license token.

The statistical eye diagram is a fundamental performance metric for high-speed serial interfaces in the bit error rate (BER). When setting up a Statistical Eye Analysis, the Port element is used to designate the incident (input) and probe (output) ports for the system to be analyzed. Ports can be specified as single-ended or mixed mode. Random jitter can be applied to each incident and probe point in the system.

Each incident port acts as random bit pattern source with specified voltage magnitude. If an incident port element does not have a time domain voltage magnitude specification, the default values, V_high=1.0, V_low=-1.0 are used.

Probe ports are used as observation points where `.PRINT`, `.PROBE`, and `.MEASURE` commands can be defined.

Example

```
.STATEYE T=400p Trf=20p
+ incident_port= 1, 2
+ probe_port= 3, 4
+ Rj = 5p, 5p, 2p, 2p tran_init= 50
+ T_resolution= 300 V_resolution= 300
```

See Also

.MEASURE (or) .MEAS
.PRINT
.PROBE
Statistical Eye Analysis

.STIM

Uses the results (output) of one simulation as input stimuli in a new simulation in HSPICE.

Syntax

General Syntax:

```
.STIM [tran|ac|dc] PWL|DATA|VEC
+ [filename=output_filename ...]
```

PWL Source Syntax (Transient Analysis Only)

```
.STIM   [tran] PWL [filename=output_filename]
+   [name1=] ovar1 [node1=n+] [node2=n-]
+   [[name2=]ovar2 [node1=n+] [node2=n-] ...]
+   [from=val] [to=val] [npoints=val]
.STIM   [tran] PWL [filename=output_filename]
+   [name1=] ovar1 [node1=n+] [node2=n-]
+   [[name2=]ovar2 [node1=n+] [node2=n-] ...]
+   indepvar=[()t1 [t2 ...()]]
```

Data Card Syntax

```
.STIM   [tran|ac|dc] DATA [filename=output_filename]
+   dataname [name1=] ovar1
+   [[name2=]ovar2 ...] [from=val] [to=val]
+   [npoints=val] [indepout=val]
.STIM   [tran | ac | dc] DATA [filename=output_filename]
+   dataname [name1=] ovar1
+   [[name2=]ovar2 ...] indepvar=[()t1 [t2 ...()]]
+   [indepout=val]
```

Digital Vector File Syntax (Transient Analysis Only)

```
.STIM [tran] VEC [filename=output_filename]
+   vth=val vtl=val [voh=val] [vol=val]
+ [name1=] ovar1 [[name2=] ovar2 ...]
+ [from=val] [to=val] [npoints=val]

.STIM [tran] VEC [filename=output_filename]
+   vth=val vtl=val [voh=val] [vol=val]
+ [name1=] ovar1 [[name2=] ovar2 ...]
+ indepvar=[()t1 [t2 ...()]]
```

Arguments

PWL Source (Transient Analysis Only):

Argument	Description
tran	Transient simulation.
filename	Output file name. If you do not specify a file, HSPICE uses the input filename.
name1	PWL Source Name that you specify. The name must start with V (for a voltage source) or I (for a current source).
ovar1	Output variable that you specify. *ovar* can be: ▪ Node voltage. ▪ Element current. ▪ Parameter string. If using a parameter string you must specify *name1*. For example: v(1), i(r1), v(2,1), par('v(1)+v(2)')
node1	Positive terminal node name.
node2	Negative terminal node name.
from	Time to start output of simulation results. For transient analysis, it uses the time units that you specified. Cannot use with indepvar.
npoints	Number of output time points.
to	Time to terminate output of simulation results. For transient analysis, it uses the time units that you specified. The *from* value can be greater than the *to* value. Cannot use with indepvar.
indepvar	Dispersed (independent-variable) time points. Specify dispersed time points in increasing order. Replaces the "from" and "to" construct.

Data Card:

Argument	Description
tran | ac | dc	Simulation type: transient, AC, or DC.

Argument	Description
filename	Output file name. If you do not specify a file, HSPICE uses the input filename.
dataname	Name of the data card to generate.
from	Time to start output of simulation results. For transient analysis, it uses the time units that you specified. Cannot use with indepvar for time.
to	Time to terminate output of simulation results. For transient analysis, it uses the time units that you specified. Cannot use with indepvar for time.
name1	Name of a parameter of the data card to generate.
npoints	Number of output independent-variable points.
indepvar	Dispersed independent-variable points. Replaces the “from” and “to” constructs.
indepout	Indicates whether to generate the independent variable column. ▪ indepout, indepout=1, or on, produces the independent variable column. You can specify the independent-variables in any order. ▪ indepout= 0 or off (default) does not create an independent variable column. You can place the indepout field anywhere after the ovar1 field.
ovar1	Output variable that you specify. *ovar* can be: ▪ Node voltage. ▪ Element current. ▪ Element templates. ▪ Parameter string. You cannot use character strings as parameter values in HSPICE RF. For example: v(1), i(r1), v(2,1), par(’v(1)+v(2)’), LX1(m1), LX2(m1)

Digital Vector File (Transient Analysis Only):

Argument	Description
name1	Signal name that you specify.
filename	Output file name. If you do not specify a file, HSPICE uses the input filename.
ovar1	Output variable that you specify. *ovar* can only be a node voltage.
from	Time to start output of simulation results. For transient analysis, uses the time units that you specified.
to	Time to the terminate output of simulation results. For transient analysis, uses the specified time units.The *from* value can be greater than the *to* value.
npoints	Number of output time points.
indepvar	Specifies dispersed independent-variable points. You must specify dispersed time points in increasing order. Replaces the “from” and “to” construct.
vth	High voltage threshold.
vtl	Low voltage threshold.
voh	Logic-high voltage for each output signal.
vol	Logic-low voltage for each output signal.

Description

Use this command to reuse the results (output) of one simulation as input stimuli in a new simulation.

The `.STIM` command specifies:

- Expected stimulus (PWL Source, DATA CARD, or VEC FILE).
- Signals to transform.
- Independent variables.

One `.STIM` command produces one corresponding output file.

For additional information, see "Reusing Simulation Output as Input Stimuli" in the *HSPICE User Guide: Simulation and Analysis*.

Examples

Example 1: In this example, the `.STIM` command creates a file named "test.pwl0_tr0", having a voltage source named "v0" applied between nodes neg and 0 (ground). It has a PWL source function based on the voltage of node n0 during the time 0.0ns to 5.0ns with 10 points.

```
.stim tran pwl filename=test v0=v(n0) node1=neg
+ node2=0 from=0.0ns to=5ns npoints=10
```

Example 2: In this example the "from and to" construct is used:

```
.stim tran data filename=new PWL v(2) from=start to=end
```

Example 3: In this example, the indepvar construct replaces "from and to". (Using both constructs results in an error.)

```
.stim tran pwl filename=new v(2) indepvar=(2n 3n 4n)
```

See Also

.DOUT
.MEASURE (or) .MEAS
.PRINT
.PROBE

.SUBCKT

Defines a subcircuit in a netlist.

Syntax

```
.SUBCKT subnam n1 n2 n3 ... [parnam=val]
.ENDS
.SUBCKT SubName PinList [SubDefaultsList]
.ENDS
.SUBCKT subnam n1 n2 n3 ... [param=str('string')]
.ENDS
```

Arguments

Argument	Description
subnam	Reference name for the subcircuit model call.
n1 ...	Node numbers for external reference; cannot be the ground node (0, gnd, ground, gnd!). Any element nodes that are in the subcircuit, but are not in this list are strictly local with three exceptions: ▪ Ground node (0, gnd, ground, gnd!). ▪ Nodes assigned using BULK=node in MOSFET or BJT models. ▪ Nodes assigned using the .GLOBAL command.
parnam	Parameter name set to a value. Use only in the subcircuit. To override this value, assign it in the subcircuit call or set a value in a .PARAM command.
SubDefaultsList	*SubParam1=Expression* [*SubParam2=Expression*...]

Description

Use this command to define a subcircuit in your netlist. You can create a subcircuit description for a commonly used circuit and include one or more references to the subcircuit in your netlist.

When you use hierarchical subcircuits, you can pick default values for circuit elements in a `.SUBCKT` command. You can use this feature in cell definitions to simulate the circuit with typical values.

Use the `.ENDS` command to terminate a `.SUBCKT` command.

Note:

Using `-top subck_name` on the command line effectively eliminates the need for the `.subckt` *`subckt_name`* `and .ends` *`subckt_name`*.

Example 1

This example defines two subcircuits: `SUB1` and `SUB2`. These are resistor-divider networks, whose resistance values are parameters (variables). The `X1`, `X2`, and `X3` commands call these subcircuits. Because the resistor values are different in each call, these three calls produce different subcircuits.

```
*FILE SUB2.SP TEST OF SUBCIRCUITS
.OPTION LIST ACCT
   V1 1 0 1
.PARAM P5=5 P2=10
.SUBCKT SUB1 1 2 P4=4
   R1 1 0 P4
   R2 2 0 P5
   X1 1 2 SUB2 P6=7
   X2 1 2 SUB2
.ENDS
*
.MACRO SUB2 1 2 P6=11
   R1 1 2 P6
   R2 2 0 P2
.EOM
   X1 1 2 SUB1 P4=6
   X2 3 4 SUB1 P6=15
   X3 3 4 SUB2
*
.MODEL DA D CJA=CAJA CJP=CAJP VRB=-20
   IS=7.62E-18
+   PHI=.5 EXA=.5 EXP=.33
.PARAM CAJA=2.535E-16 CAJP=2.53E-16
.END
```

Example 2

This example implements an inverter that uses a *Strength* parameter. By default, the inverter can drive three devices. Enter a new value for the *Strength* parameter in the element line to select larger or smaller inverters for the application.

```
.SUBCKT Inv a y Strength=3
   Mp1 <MosPinList> pMosMod L=1.2u
   W='Strength * 2u'
   Mn1 <MosPinList> nMosMod L=1.2u
   W='Strength * 1u'
.ENDS
...
xInv0 a y0 Inv   $ Default devices: p device=6u,
      $ n device=3u
xInv1 a y1 Inv Strength=5   $ p device=10u,
   n device=5u
xInv2 a y2 Inv Strength=1   $ p device= 2u,
   n device=1u
...
```

Example 3

This example implements an IBIS model (in HSPICE only) that uses string parameters to specify the IBIS file name and IBIS model name.

```
* Using string parameters
.subckt IBIS vccq vss out in
+ IBIS_FILE=str('file.ibs')
+ IBIS_MODEL=str('ibis_model')
ven en 0 vcc
B1 vccq vss out in en v0dq0 vccq vss
+ file= str(IBIS_FILE) model=str(IBIS_MODEL)
.ends
```

See Also

.ENDS
.EOM
.MACRO
.MODEL
.OPTION LIST
.PARAM

.SURGE

Automatically detects and reports a current surge that exceeds the specified surge tolerance in HSPICE RF.

Syntax

`.SURGE` *surge_threshold surge_width node1* [*node2 ...noden*]

Arguments

Argument	Description
surge_threshold	Minimum absolute surge current.
surge_width	Defines the minimum duration of a surge.
noden	Any valid node name at current or lower subcircuit level.

Description

Use this command to automatically detect and report a current surge that exceeds the specified surge tolerance. The command reports any current surge that is greater than *surge_threshold* for a duration of more than *surge_width*.

Surge current is defined as the current flowing into or out of a node to the lower subcircuit hierarchy.

Example

In this example, the `.SURGE` command detects any current surge that has an absolute amplitude of more than 1mA, and that exceeds 100ns, x(xm.x1.a), x(xm.x2.c), and x(xn.y).

```
.SUBCKT sa a b
...
.ENDS
.SUBCKT sb c d
...
.ENDS
.SUBCKT sx x y
x1 x y sa
x2 x a sb
.ENDS
xm 1 2 sx
xn 2 a sx
.SURGE 1mA 100ns xm.x1.a xm.x2.c xn.y
```

.SWEEPBLOCK

Creates a sweep whose set of values is the union of a set of linear, logarithmic, and point sweeps in HSPICE RF.

Syntax

```
.SWEEPBLOCK swblockname sweepspec [sweepspec
+ [sweepspec [...]]]
```

Arguments

Argument	Description
swblockname	Assigns a name to SWEEPBLOCK.
sweepspec	You can specify an unlimited number of *sweepspec* parameters. Each `sweepspec` can specify a linear, logarithmic, or point sweep by using one of the following forms:*start stop increment* lin *npoints start stop* dec *npoints start stop* oct *npoints start stop* poi *npoints p1 p2 ...*

Description

Use this command to create a sweep whose set of values is the union of a set of linear, logarithmic, and point sweeps.

You can use this command to specify DC sweeps, parameter sweeps, AC, and HBAC frequency sweeps, or wherever HSPICE accepts sweeps.

For additional information, see "SWEEPBLOCK in Sweep Analyses" in the *HSPICE User Guide: RF Analysis*.

Example

The following example specifies a logarithmic sweep from 1 to 1e9 with more resolution from 1e6 to 1e7:

```
.sweepblock freqsweep dec 10 1 1g dec 1000 1meg 10meg
```

See Also

.AC
.DC

.ENV
.HB
.HBAC
.HBLSP
.HBNOISE
.HBOSC
.HBXF
.PHASENOISE
.TRAN

.TEMP (or) .TEMPERATURE

Specifies the circuit temperature for an HSPICE/HSPICE RF simulation.

Syntax

```
.TEMP t1 [t2 t3 ...]
```

Arguments

Argument	Description
t1 t2	Temperatures in ×C at when HSPICE/HSPICE RF simulates the circuit.

Description

Use this command to specify the circuit temperature for an HSPICE simulation. You can use either the .TEMP command or the TEMP parameter in the .DC, .AC, and .TRAN commands. HSPICE compares the circuit simulation temperature against the reference temperature in the TNOM option. HSPICE uses the difference between the circuit simulation temperature and the TNOM reference temperature to define derating factors for component values.

HSPICE RF supports only one .TEMP command in a netlist. If you use multiple .TEMP commands, only the last one will be used.

Note:

HSPICE allows multiple .TEMP commands in a netlist and performs multiple DC, AC or TRAN analyses for each temperature. Do not set the temperature to the same value multiple times.

When you use multiple temperature values in a .TEMP command, HSPICE RF performs multiple HB, SN, PHASENOISE, etc. analyses for each temperature. The simulation results for the different temperature values are saved using a file naming convention consistent with .ALTER commands.

Example 1

```
.TEMP -55.0 25.0 125.0
```

The .TEMP command sets the circuit temperatures for the entire circuit simulation. To simulate the circuit by using individual elements or model temperatures, HSPICE/HSPICE RF uses:

- Temperature as set in the .TEMP command.
- .OPTION TNOM setting (or the TREF model parameter).
- DTEMP element temperature.

Example 2

```
.TEMP 100
D1 N1 N2 DMOD DTEMP=30
D2 NA NC DMOD
R1 NP NN 100 TC1=1 DTEMP=-30
.MODEL DMOD D IS=1E-15 VJ=0.6 CJA=1.2E-13
+ CJP=1.3E-14 TREF=60.0
```

In this example:

- The .TEMP command sets the circuit simulation temperature to 100° C.
- You do not specify .OPTION TNOM so it defaults to 25° C.
- The temperature of the diode is 30° C above the circuit temperature as set in the DTEMP parameter.

That is:

- D1temp=100° C + 30° C=130° C.
- HSPICE/HSPICE RF simulates the D2 diode at 100° C.
- R1 simulates at 70° C.

Because the diode model command specifies TREF at 60° C, HSPICE/HSPICE RF derates the specified model parameters by:

- 70° C (130° C - 60° C) for the D1 diode.
- 40° C (100° C - 60° C) for the D2 diode.
- 45° C (70° C - TNOM) for the R1 resistor.

Example 3

```
.param mytemp =0
.temp '105 + 3*mytemp'
```

In this example, parameterized .TEMP is also supported.

See Also

.AC
.DC
.OPTION TNOM
.TRAN

.TF

Calculates DC small-signal values for transfer functions.

Syntax

```
.TF ov srcnam
```

Arguments

Argument	Description
ov	Small-signal output variable.
srcnam	Small-signal input source.

Description

Use this command to calculate DC small-signal values for transfer functions (ratio of output variable to input source). You do not need to specify `.OP`.

The `.TF` command defines small-signal output and input for DC small-signal analysis. When you use this command, HSPICE computes:

- DC small-signal value of the transfer function (output/input)
- Input resistance
- Output resistance

Example

```
.TF V(5,3) VIN
.TF I(VLOAD) VIN
```

For the first example, HSPICE computes the ratio of V(5,3) to `VIN`. This is the ratio of small-signal input resistance at VIN to the small-signal output resistance (measured across nodes 5 and 3). If you specify more than one `.TF` command in a single simulation, HSPICE runs only the last `.TF` command.

See Also

.DC

.TITLE

Sets the simulation title.

Syntax

```
.TITLE string_of_up_to_72_characters
-or-
string_of_up_to_72_characters
```

Arguments

Argument	Description
string	Any character string up to 72 characters long.

Description

Use this command to set the simulation title in the first line of the input file. This line is read and used as the title of the simulation, regardless of the line's contents. The simulation prints the title verbatim in each section heading of the output listing file.

To set the title you can place a .TITLE command on the first line of the netlist. However, the .TITLE syntax is not required.

In the second form of the syntax, the string is the first line of the input file. The first line of the input file is always the implicit title. If any command appears as the first line in a file, simulation interprets it as a title and does not execute it.

An .ALTER command does not support using the .TITLE command. To change a title for a .ALTER command, place the title content in the .ALTER command itself.

Example

```
.TITLE my-design_netlist
```

.TRAN

Starts a transient analysis that simulates a circuit at a specific time. In HSPICE RF you can run a parameter sweep around a single analysis, but the parameter sweep cannot change an .OPTION value. In addition, HSPICE RF does not support the .TRAN DATA command and only supports the data-driven syntax for parameter sweeps (for example, .TRAN AB sweepdata=name).

Syntax

Syntax for Single-Point Analysis:

```
.TRAN tstep1 tstop1 [START=val] [UIC]
```

Syntax for Double-Point Analysis:

```
.TRAN tstep1 tstop1 [tstep2 tstop2]
+ [START=val] [UIC] [SWEEP var type np pstart pstop]
.TRAN tstep1 tstop1 [tstep2 tstop2]
+ [START=val] [UIC] [SWEEP var START="param_expr1"
+ STOP="param_expr2" STEP="param_expr3"]
.TRAN tstep1 tstop1 [tstep2 tstop2] [START=val] [UIC]
+ [SWEEP var start_expr stop_expr step_expr]
```

Syntax for Multipoint Analysis:

```
.TRAN tstep1 tstop1 [tstep2 tstop2 ...tstepN tstopN]
+ [START=val] [UIC] [SWEEP var type np pstart pstop]
.TRAN tstep1 tstop1 [tstep2 tstop2 ...tstepN tstopN]
+ [START=val] [UIC] [SWEEP var START="param_expr1"
+ STOP="param_expr2" STEP="param_expr3"]
.TRAN tstep1 tstop1 [tstep2 tstop2 ...tstepN tstopN>
+ [START=val> [UIC]
+ [SWEEP var start_expr stop_expr step_expr]
```

Syntax for Data-Driven Sweep:

```
.TRAN DATA=datanm
.TRAN tstep1 tstop1 [tstep2 tstop2 ...tstepN tstopN]
+ [START=val] [UIC] [SWEEP DATA=datanm]
.TRAN DATA=datanm [SWEEP var type np pstart pstop]
.TRAN DATA=datanm [SWEEP var START="param_expr1"
+ STOP="param_expr2" STEP="param_expr3"]
.TRAN DATA=datanm
+ [SWEEP var start_expr stop_expr step_expr]
```

Syntax for Monte Carlo Analysis:

```
.TRAN tstep1 tstop1 [tstep2 tstop2 ...tstepN tstopN]
+ [START=val] [UIC] [SWEEP MONTE=MCcommand]
```

Syntax for Optimization:

```
.TRAN DATA=datanm OPTIMIZE=opt_par_fun
+ RESULTS=measnames MODEL=optmod
.TRAN  [DATA=filename] SWEEP OPTIMIZE=OPTxxx
+ RESULTS=ierr1 ... ierrn MODEL=optmod
```

Note:

If UIC is added to the .TRAN line, then no DC convergence is performed. The transient node voltages at t=0 are determined by searching for the first value found in this order: .IC, .NODESET, or 0v (in case neither is found). This causes either an invalid circuit state until the transient analysis is able to resolve the correct circuit state, or, possibly, a timestep too small error. Do not use UIC unless you know exactly why you are using it and you understand the implications of doing so.

Argument	Description
DATA=datanm	Data name, referenced in the .TRAN command.
MONTE= MCcommand	Where MCcommand can be any of the following: ■ *val* Specifies the number of random samples to produce. ■ val firstnum=*num* Specifies the sample number on which the simulation starts. ■ list num Specifies the sample number to execute. ■ list(*num1:num2 num3 num4:num5*) Samples from *num1* to *num2*, sample *num3*, and samples from *num4* to *num5* are executed (parentheses are optional).
np	Number of points or number of points per decade or octave, depending on what keyword precedes it.
param_expr...	Expressions you specify: *param_expr1...param_exprN*.

Argument	Description
pincr	Voltage, current, element, or model parameter; or any temperature increment value. If you set the *type* variation, use *np* (number of points), not *pincr*.
pstart	Starting voltage, current, or temperature; or any element or model parameter value. If you set the *type* variation to POI (list of points), use a list of parameter values, instead of *pstart pstop*.
pstop	Final value: voltage, current, temperature; element or model param.
START	Time when printing or plotting begins. Caution: If you use `.TRAN` with a `.MEASURE` command, a non-zero START time can cause incorrect `.MEASURE` results. Do not use non-zero START times in `.TRAN` commands when you also use `.MEASURE`.
SWEEP	Indicates that `.TRAN` specifies a second sweep.
tstep1...	Printing or plotting increment for printer output and the suggested computing increment for post-processing. This argument is always a positive value.
tstop1...	Time when a transient analysis stops incrementing by the first specified time increment (*tstep1*). If another tstep-tstop pair follows, analysis continues with a new increment. This argument is always a positive value.

Argument	Description
UIC	If you specify the UIC parameter in the .TRAN command, HSPICE does not calculate the initial DC operating point, but directly enters transient analysis. When you use .TRAN UIC, the .TRAN node values (at time zero) are determined by searching for the first value found in this order: from .IC value, then IC parameter on an element command, then .NODESET value, otherwise use a voltage of zero. Note that forcing a node value of the DC operating point might not satisfy KVL and KCL. In this event you might see activity during the initial part of the simulation. This might happen if you use UIC and do not specify some node values, when you specify too many (conflicting) .IC values are specified, or when you force node values and the topology changes. Forcing a node voltage applies a fixed equivalent voltage source during DC analysis and transient analysis removes the voltage sources to calculate the second and later time points. Therefore, to correct DC convergence problems use .NODESETs (without .TRAN UIC) liberally (when a good guess can be provided) and use .ICs sparingly (when the exact node voltage is known).
type	Any of the following keywords: ■ DEC – decade variation. ■ OCT – octave variation (the value of the designated variable is eight times its previous value). ■ LIN – linear variation. ■ POI – list of points.
var	Name of an independent voltage or current source, any element or model parameter, or the TEMP keyword (indicating a temperature sweep). You can use a source value sweep, referring to the source name (SPICE style). However, if you specify a parameter sweep, a .DATA command, or a temperature sweep you must choose a parameter name for the source value and subsequently refer to it in the .TRAN command. The parameter must not start with TEMP and should be defined in advance using the .PARAM command.
firstrun	MONTE=*val* value specifies the number of Monte Carlo iterations to perform. This argument specifies the desired number of iterations. HSPICE runs from num1 to num1+val-1.

Argument	Description
list	Iterations at which HSPICE performs a Monte Carlo analysis. You can write more than one number after *list*. The colon represents “from ... to ...". Specifying only one number makes HSPICE run at only the specified point.
OPTIMIZE	When used with .TRAN and SWEEP, this argument is either opt_par_fun or OPTxxx for a bisection/Monte Carlo analysis in HSPICE.

Description

Use to start a transient analysis that simulates a circuit at a specific time.

For single-point analysis, the values of the tstep, tstop1, and START arguments should obey the following rules:

```
START < tstop1
tstep <= tstop1 – START
```

For double-point analysis, the values of the tstep1, tstop1, tstep2, tstop2, and START arguments should obey the following rules:

```
START < tstop1 < tstop2
tstep1 <= tstop1 – START
tstep2 <= tstop2 – tstop1
```

In double-point analysis, if tstep2 < tstop1, tstop2 < tstop1, and START is not explicitly set, the command is interpreted as:

```
.TRAN tstep tstop1 start delmax
```

There can be three different “DELMAX” values involved in a .TRAN command:

- .OPTION DELMAX (value specified with this .OPTION)
- delmax (value that can be specified with the .TRAN command)
- “auto” DELMAX (value that is computed automatically)

When column 4 is interpreted as delmax, this command has a higher priority than the DELMAX option. The maximum internal timestep taken by HSPICE during transient analysis is referred to as Δt_{max}. Its value is normally computed automatically based on several timestep control settings. If you wish to override the automatically computed value, and force the maximum step size to be a specific value, you can do so with .OPTION DELMAX, or by specifying a delmax value with the .TRAN command. If not specified, HSPICE

automatically computes a DELMAX "auto" value, based on timestep control factors such as FS and RMAX. (For a complete list of timestep control factors, see Transient Control Options in the *HSPICE User Guide: Simulation and Analysis*.)

For multipoint analysis, the values of the tstep1, tstop1, ..., tstepN, tstopN, and START arguments should obey the following rules:

```
START < tstop1 < tstop2 < ... < tstopN
tstep1 <= tstop1 – START
tstep2 <= tstop2 – tstop1
...
tstepN <= tstopN – tstop(N-1)
```

The following syntax shows multiple timestep increments in HSPICE transient analysis:

```
.tran tstep1 tend1 tstep2 tend2 tstep3 tend3 ...
```

or

```
.tran tstep tend tstart delmax
```

The following limitation applies for HSPICE:
The ratio between tstop1 and tstep must be ≥ 1e09. For example, .TRAN 8n 8 is permissible, but .TRAN 0.1n 8 is not.

Example 1

This example performs and prints the transient analysis every 1 ns for 100 ns.

```
.TRAN 1NS 100NS
```

Example 2

This example performs the calculation every 0.1 ns for the first 25 ns; and then every 1 ns until 40 ns. Printing and plotting begin at 10 ns.

```
.TRAN .1NS 25NS 1NS 40NS START=10NS
```

Example 3

This example performs the calculation every 0.1 ns for 25 ns and delmax is set to 0.05 ns; Printing and plotting begin at 1 ns.

```
.TRAN .1NS 25NS 1NS .05NS
```

Example 4

This example does the calculation every 0.1 ns for 25 ns; and then every 1 ns for 40 ns; and then every 2 ns until 100 ns. Printing and plotting begin at 10 ns.

```
.TRAN .1NS 25NS 1NS 40NS 2NS 100NS START = 10NS
```

Example 5

This example performs the calculation every 10 ns for 1 µs. This example bypasses the initial DC operating point calculation. It uses the nodal voltages specified in the `.IC` command (or by IC parameters in element commands) to calculate the initial conditions.

```
.TRAN 10NS 1US UIC
```

Example 6

This example increases the temperature by 10 °C through the range -55 °C to 75 °C. It also performs transient analysis for each temperature.

```
.TRAN 10NS 1US UIC SWEEP TEMP -55 75 10
```

Example 7

This example analyzes each load parameter value at 1 pF, 5 pF, and 10 pF.

```
.TRAN 10NS 1US SWEEP load POI 3 1pf 5pf 10pf
```

Example 8

This example is a data-driven time sweep. It uses a data file as the sweep input. If the parameters in the data command are controlling sources, then a piecewise linear specification must reference them.

```
.TRAN data=dataname
```

Example 9

This example performs the calculation every `10ns` for `1us` from the 11th to 20th Monte Carlo trials.

```
.TRAN 10NS 1US SWEEP MONTE=10 firstrun=11
```

Example 10

This example performs the calculation every `10ns` for `1us` at the 10th trial, then from the 20th to the 30th trial, followed by the 35th to the 40th trial and finally at the 50th Monte Carlo trial.

```
.TRAN 10NS 1US SWEEP MONTE=list(10 20:30 35:40 50)
```

See Also

.IC
.NODESET
.OPTION DELMAX
Timing Analysis Using Bisection
Transient Analysis

.UNPROTECT or .UNPROT

Restores normal output functions previously restricted by a `.PROTECT` command as part of the encryption process in HSPICE.

Syntax

```
.UNPROTECT
```

Description

Use this command to restore normal output functions previously restricted by a `.PROTECT` command.

- Any elements and models located between `.PROTECT` and `.UNPROTECT` commands, inhibit the element and model listing from the `LIST` option.
- Neither the `.OPTION NODE` cross-reference, nor the `.OP` operating point printout list any nodes within the `.PROTECT` and `.UNPROTECT` commands.
- The `.UNPROTECT` command is encrypted during the encryption process.

Note:

If you use `.prot/.unprot` in a library or file that is not encrypted you might get warnings that the file is encrypted and the file or library is treated as a "black box."

The `.prot` and `.unprot` commands act similar to `.option brief=1` and `.option brief=0`, respectively.

See Also

.PROTECT or .PROT
.OPTION BRIEF

.VARIATION

Specifies global and local variations on model parameters in HSPICE.

Syntax

```
.Variation
   Define options
   Define common parameters that apply to all subblocks
   .Global_Variation
      Define the univariate independent random variables
      Define additional random variables through transformation
      Define variations of model parameters
   .End_Global_Variation
   .Local_variation
      Define the univariate independent random variables
      Define additional random variables through transformation
      Define variations of model parameters
      .Element_Variation
         Define variations of element parameters
      .End_Element_Variation
   .End_Local_Variation
   .Spatial_Variation
       Define the univariate independent random variables
       Define additional random variables through transformation
       Define variations of model parameters
   .End_Spatial_Variation
.End_Variation
```

Description

Use this command to specify global, local, and spatial variations on model parameters, resulting from variations in materials and manufacturing. If a Variation Block is read as part of `.ALTER` processing, then the contents are treated as additive. If the same parameters are redefined, HSPICE considers this an error.

For a detailed description of the Variation Block and usage examples, see Analyzing Variability and Using the Variation Block in the *HSPICE User Guide: Simulation and Analysis* and for Variation Block options, see Control Options and Syntax.

Parameters and Options

Constant Parameter

Definition, which can be referenced anywhere within the Variation Block:

parameter PARAM=value

Univariate Independent Random Variable

parameter IVarName=N() normal distribution

parameter IVarName=U() uniform distribution

parameter IVarName=CDF(xn,yn) cumulative distribution function

Transformed Random Variable

parameter TVarName=expression(IVarName*<IVarName>*)

Variation Definition for Model Parameter

modelType modelName paramName=Expression_For_Sigma

Variation Definition for Element Parameter

modelType paramName=Expression_For_Sigma

modelType(condition) paramName=Expression_For_Sigma

Expression_For_Sigma

Implicit definition: Normal Distribution with 0 mean and Sigma equal content	
value | expression	absolute variation
value % | expression %	relative variation

Referencing previously defined Random Variable	
perturb('expression(IVarName|TVarName*<IVarName><TVarName>*)')	absolute
perturb('expression(IVarName|TVarName*<IVarName><TVarName>*)') %	relative

Access Function

For element parameter (for example w, l, x, y):
get_E(elementParameter)

For netlist parameter (for example .param vdd, temper):
get_P(Parameter)

Options

For detailed information on `.VARIATION` command control parameters and examples, see the *HSPICE User Guide: Simulation and Analysis,* Analyzing Variability and Using the Variation Block and Monte Carlo Analysis Using the Variation Block Flow.

Note:

Note that ".OPTION" with a leading period does not work for options specified in the Variation Block.

The correct Variation Block syntax is:

```
Option optionName = value
```

Options
option Ignore_Local_Variation=No|Yes
option Ignore_Global_Variation=No|Yes
option ignore_Spatial_Variation=No|Yes
option Ignore_Interconnect_Variation=No|Yes
option Normal_Limit=value
option Output_Sigma_Value=value
option Vary_Only_Subckt=Subckt_List | Do_Not_Vary_Subckt=Subckt_List
option Sampling_Method=SRS | Factorial | OFAT | LHS

.VEC

Calls a digital vector file from an HSPICE/HSPICE RF netlist.

Syntax

```
.VEC 'digital_vector_file'
```

Description

Use this command to call a digital vector file from an HSPICE netlist. A digital vector file consists of three parts:

- Vector Pattern Definition section
- Waveform Characteristics section
- Tabular Data section.

The `.VEC` file must be a text file. If you transfer the file between UNIX/Linux and Windows, use text mode. See Chapter 4, Digital Vector File Commands for more information.

Example

This is a fragment from a netlist with a call to a digital vector file.

```
*file: mos2bit_v.sp - adder - 2 bit all-nand-gate binary adder
*uses digital vector input
.options post nomod
.option opts fast
*
.tran .5ns 60ns
*
.vec 'digstim.vec'
...
```

3

HSPICE and RF Netlist Simulation Control Options

Describes the HSPICE and HSPICE RF simulation control options you can set using various forms of the `.OPTION` *command.*

You can set HSPICE and HSPICE RF simulation control options using the `.OPTION` command. This chapter provides a list of the options grouped by usage, followed by detailed descriptions of the individual options in an alphabetical list. Note that in many cases an option is only usable in either the HSPICE or HSPICE RF mode of operation. In a few instances, an option has different functionality, depending on which mode (HSPICE or HSPICE RF) has been invoked. The description of the command notes the differences.

The control options described in this chapter fall into the following broad categories:

- General Control Options
- Input/Output Controls
- Model Analysis
- HSPICE Analysis Options
- HSPICE RF Analysis Options
- Transient and AC Small Signal Analysis Options
- Transient Control Options
- .VARIATION Block Control Options
- .DESIGN_EXPLORATION Block Control Options

Notes on Default Values

The typical behavior for options is:

- Option not specified: value is default value, typically “OFF” or 0.
- Option specified but without value: typically turns the option “ON” or to a value of 1.

If an option has more than two values allowed, specifying it without a value sets it to 1, if appropriate.

In most cases, options without values are allowed only for flags that can be on or off, and specifying the option without a value turns it on. There are a few options (such as POST), where there are more than two values allowed, but you can still specify it without a value. Usually, you should expect it to be 1.

HSPICE Control Options Grouped By Function

General Control Options

Netlist Parser Control

.OPTION ALTCC	.OPTION DIAGNOSTIC (or) .OPTION DIAGNO	.OPTION NOTOP	.OPTION SEARCH
.OPTION ALTCHK	.OPTION NOELCK	.OPTION NOWARN	.OPTION WARNLIMIT (or) .OPTION WARNLIM
.OPTION BADCHR	.OPTION NOMOD	.OPTION PARHIER (or) .OPTION PARHIE	

Output Listing Control

.OPTION ACCT	.OPTION INGOLD	.OPTION NOISEMINFREQ	.OPTION STATFL
.OPTION AUTOSTOP (or) .OPTION AUTOST	.OPTION LENNAM	.OPTION NUMDGT	.OPTION VFLOOR
.OPTION BRIEF	.OPTION LIST	.OPTION OPTLST	
.OPTION CAPTAB	.OPTION MCBRIEF	.OPTION OPTS	
.OPTION CO	.OPTION NODE	.OPTION PATHNUM	

.MEAS Options

.OPTION MEASFAIL	.OPTION MEASFILE	.OPTION MEASOUT	.OPTION PUTMEAS
.OPTION EM_RECOVERY			

.BIASCHK Options

.OPTION BIASFILE	.OPTION BIASNODE	.OPTION BIASPARALLEL	.OPTION BIAWARN
.OPTION BIASINTERVAL			

Input/Output Controls

I/O Control Options

.OPTION D_IBIS .OPTION MONTECON .OPTION POST .OPTION POSTLVL

.OPTION INTERP .OPTION OPFILE .OPTION POSTLVL .OPTION POSTTOP

.OPTION ITRPRT .OPTION PROBE

Back Annotation Post-Layout Options

.OPTION BA_ACTIVE .OPTION BA_ERROR .OPTION BA_FILE .OPTION BA_TERMINAL

Interface Control Options

.OPTION ARTIST .OPTION CSDF .OPTION DLENCSDF .OPTION PSF

Model Analysis

.OPTION APPENDALL .OPTION DEFAS .OPTION DEFPS .OPTION NCWARN

.OPTION ASPEC .OPTION DEFL .OPTION DEFSA .OPTION WL

.OPTION BSIM4PDS .OPTION DEFNRD .OPTION DEFSB .OPTION WNFLAG

.OPTION DCAP .OPTION DEFNRS .OPTION DEFSD

.OPTION DEFAD .OPTION DEFPD .OPTION DEFW

Custom Models

.OPTION CMIFLAG .OPTION CMIUSRFLAG .OPTION CUSTCMI

Model Control

.OPTION HIER_SCALE .OPTION MACMOD .OPTION MODMONTE .OPTION SEED

Scaling

.OPTION SCALE .OPTION SCALM

Temperature

.OPTION TNOM .OPTION XDTEMP

Resistance

.OPTION RESMIN

Verilog-A

.OPTION SPMODEL .OPTION VAMODEL

Diode and BJT

.OPTION DCAP .OPTION EXPLI

Inductor and Mutual Inductors

.OPTION GENK .OPTION KLIM

RC Reduction

.OPTION LA_FREQ .OPTION LA_MINC .OPTION LA_TOL .OPTION SIM_LA

.OPTION LA_MAXR .OPTION LA_TIME

HSPICE Analysis Options

AC Options

.OPTION ACOUT .OPTION LIMPTS .OPTION UNWRAP

OP and DC

Error Tolerance

.OPTION ABSH .OPTION ABSV .OPTION RELH .OPTION RELV
.OPTION ABSI .OPTION ABSVDC .OPTION RELI .OPTION RELVDC
.OPTION ABSMOS .OPTION KCLTEST .OPTION RELMOS .OPTION VNTOL
.OPTION ABSTOL .OPTION MAXAMP .OPTION RELTOL

Matrix Control

.OPTION NOPIV .OPTION PIVREF .OPTION PIVREL .OPTION PIVTOL
.OPTION PIVOT

Convergence

.OPTION CONVERGE .OPTION DCON .OPTION GRAMP .OPTION ITL2
.OPTION CSHDC .OPTION DCSTEP .OPTION GSHDC .OPTION OFF
.OPTION DCFOR .OPTION DCTRAN .OPTION ICSWEEP .OPTION NCWARN
.OPTION DCHOLD .OPTION GMINDC .OPTION ITL1 .OPTION SYMB
.OPTION DCIC .OPTION GMAX .OPTION ITLPTRAN

Pole/Zero Analysis

.OPTION CSCAL .OPTION GSCAL .OPTION PZABS .OPTION (X0R,X0I)
.OPTION FMAX .OPTION ITLPZ .OPTION PZTOL .OPTION (X1R,X1I)
.OPTION FSCAL .OPTION LSCAL .OPTION RITOL .OPTION (X2R,X21)

Transient and AC Small Signal Analysis Options

Transient/AC Accuracy Options

.OPTION FFT_ACCURATE .OPTION GMIN .OPTION RISETIME
(or) .OPTION RISETI

Transient/AC Algorithm Options

.OPTION ITL4 .OPTION MAXORD .OPTION METHOD .OPTION PURETP

Transient Control Options

Transient Control Method Options

.OPTION METHOD .OPTION WACC

Transient Control Limit Options

.OPTION AUTOSTOP .OPTION GMIN .OPTION ITL4 .OPTION RMAX
(or) .OPTION AUTOST

Pseudo Tran and Tran

Error Tolerance

.OPTION ABSH
.OPTION ABSV
.OPTION ABSVAR
.OPTION ABSVDC
.OPTION CHGTOL
.OPTION EM_RECOVERY
.OPTION KCLTEST
.OPTION MAXAMP
.OPTION RELH
.OPTION RELQ
.OPTION RELV
.OPTION RELVDC
.OPTION TRTOL
.OPTION VNTOL

Speed and Accuracy

.OPTION ACCURATE
.OPTION CSHUNT
.OPTION CVTOL
.OPTION DELMAX
.OPTION DI
.OPTION DV
.OPTION DVDT
.OPTION DVTR
.OPTION FAST
.OPTION FS
.OPTION FT
.OPTION GMIN
.OPTION GSHUNT
.OPTION IMAX
.OPTION IMIN
.OPTION ITL3
.OPTION ITL4
.OPTION ITL5
.OPTION NEWTOL
.OPTION RMAX
.OPTION RMIN
.OPTION RUNLVL
.OPTION SLOPETOL
.OPTION TIMERES
.OPTION TRCON

Bypass

.OPTION BYPASS
.OPTION BYTOL
.OPTION MBYPASS

Integration Method

.OPTION LVLTIM
.OPTION METHOD
.OPTION MAXORD
.OPTION MU
.OPTION MTTHRESH
.OPTION PURETP

Noise

.OPTION NOISEMINFREQ

.FFT Controls

.OPTION FFT_ACCURATE .OPTION FFTOUT

Transmission Lines

.OPTION RISETIME .OPTION WACC
(or) .OPTION RISETI

HSPICE RF Analysis Options

HB Options

.OPTION HBACKRYLOVDIM	.OPTION HBKRYLOVDIM	.OPTION HBTOL
.OPTION HBACKRYLOVITR	.OPTION HBKRYLOVMAXITER	.OPTION HBTRANFREQSEARCH
.OPTION HBACTOL	.OPTION HBKRYLOVTOL	.OPTION HBTRANINIT
.OPTION HBCONTINUE	.OPTION HBLINESEARCHFAC	.OPTION HBTRANPTS
.OPTION HBFREQABSTOL	.OPTION HBMAXITER	.OPTION HBTRANSTEP
.OPTION HBFREQRELTOL	.OPTION HBMAXOSCITER	.OPTION LOADHB
.OPTION HB_GIBBS	.OPTION HBPROBETOL	.OPTION SAVEHB
.OPTION HBJREUSE	.OPTION HBSOLVER	.OPTION TRANFORHB
.OPTION HBJREUSETOL		

Phase Noise Analysis

.OPTION BPNMATCHTOL	.OPTION PHASENOISEKRYLOVITER	.OPTION PHNOISELORENTZ

.OPTION PHASENOISEKRYLOVDIM .OPTION PHASENOISETOL .OPTION PHNOISEAMPM

Power Analysis

.OPTION SIM_POWER_ANALYSIS .OPTION SIM_POWERDC_HSPICE .OPTION SIM_POWERSTOP

.OPTION SIM_POWER_TOP .OPTION SIM_POWERPOST

.OPTION SIM_POWERDC_ACCURACY .OPTION SIM_POWERSTART

RC Network Reduction

.OPTION SIM_LA .OPTION SIM_LA_MINC .OPTION SIM_LA_TOL

.OPTION SIM_LA_FREQ .OPTION SIM_LA_MINMODE .OPTION SIM_LA_TIME

.OPTION SIM_LA_MAXR

Simulation Output

.OPTION SIM_POSTAT .OPTION SIM_POSTSCOPE .OPTION SIM_POSTTOP

.OPTION SIM_POSTDOWN .OPTION SIM_POSTSKIP

Shooting Newton Options

.OPTION LOADSNINIT .OPTION SAVESNINIT .OPTION SNACCURACY .OPTION SNMAXITER

DSPF Options

.OPTION SIM_DELTAI
.OPTION SIM_DELTAV
.OPTION SIM_DSPF
.OPTION SIM_DSPF_ACTIVE
.OPTION SIM_DSPF_INSERROR
.OPTION SIM_DSPF_LUMPCAPS
.OPTION SIM_DSPF_MAX_ITER
.OPTION SIM_DSPF_RAIL
.OPTION SIM_DSPF_SCALEC
.OPTION SIM_DSPF_SCALER
.OPTION SIM_DSPF_VTOL

SPEF Options

.OPTION SIM_SPEF
.OPTION SIM_SPEF_ACTIVE
.OPTION SIM_SPEF_INSERROR
.OPTION SIM_SPEF_LUMPCAPS
.OPTION SIM_SPEF_MAX_ITER
.OPTION SIM_SPEF_PARVALUE
.OPTION SIM_SPEF_RAIL
.OPTION SIM_SPEF_SCALEC
.OPTION SIM_SPEF_SCALER
.OPTION SIM_SPEF_VTOL

Transient Accuracy Options

.OPTION FFT_ACCURATE
.OPTION SIM_ACCURACY
.OPTION SIM_ORDER
.OPTION SIM_TG_THETA
.OPTION SIM_TRAP
.OPTION SIM_OSC_DETECT_TOL

.OPTION ABSH

Sets the absolute current change through voltage-defined branches.

Syntax

```
.OPTION ABSH=x
```

Default

`0.0`

Description

Use this option to set the absolute current change through voltage-defined branches (voltage sources and inductors). Use this option with options `DI` and `RELH` to check for current convergence.

See Also

.OPTION DI
.OPTION RELH

.OPTION ABSI

Sets the absolute error tolerance for branch currents in diodes, BJTs, and JFETs during DC and transient analysis.

Syntax

```
.OPTION ABSI=x
```

Default `1e-9` when `KCLTEST=0` or `1e-6` when `KCLTEST=1`.

Description

Use this option to set the absolute error tolerance for branch currents in diodes, BJTs, and JFETs during DC and transient analysis. Decrease `ABSI` if accuracy is more important than convergence time.

To analyze currents less than 1 nanoamp, change `ABSI` to a value at least two orders of magnitude smaller than the minimum expected current.
Min value: 1e-25; Max value: 10.

See Also

.DC
.OPTION ABSMOS
.OPTION KCLTEST
.TRAN

.OPTION ABSMOS

Specifies the current error tolerance for MOSFET devices in DC or transient analysis.

Syntax

```
.OPTION ABSMOS=x
```

Default `1e-06 (1.00u)` (amperes)

Description

Use this option to specify the current error tolerance for MOSFET devices in DC or transient analysis. The `ABSMOS` setting determines whether the drain-to-source current solution has converged. The drain-to-source current converged if:

- The difference between the drain-to-source current in the last iteration and the current iteration is less than `ABSMOS`, or
- This difference is greater than `ABSMOS`, but the percent change is less than `RELMOS`.

Min value: 11e-15; Max value 10.

If other accuracy tolerances also indicate convergence, HSPICE solves the circuit at that timepoint and calculates the next timepoint solution. For low-power circuits, optimization, and single transistor simulations, set `ABSMOS=1e-12`.

See Also

.DC
.OPTION RELMOS
.TRAN

.OPTION ABSTOL

Sets the absolute error tolerance for branch currents in DC and transient analysis.

Syntax

```
.OPTION ABSTOL=x
```

Default

`1e-9`

Description

Use this option to set the absolute error tolerance for branch currents in DC and transient analysis. Decrease `ABSTOL` if accuracy is more important than convergence time. `ABSTOL` is the same as `ABSI`.
Min value: 1e-25; Max value: 10.

See Also

.DC
.OPTION ABSI
.OPTION ABSMOS
.TRAN

.OPTION ABSV

Sets the absolute minimum voltage for DC and transient analysis.

Syntax

```
.OPTION ABSV=x
```

Default `50` uV

Description

Use this option to set the absolute minimum voltage for DC and transient analysis. `ABSV` is the same as `VNTOL`.

- If accuracy is more critical than convergence, decrease `ABSV`.
- If you need voltages less than 50 uV, reduce `ABSV` to two orders of magnitude less than the smallest desired voltage. This ensures at least two significant digits.

Typically, you do not need to change `ABSV`, except to simulate a high-voltage circuit. A reasonable value for 1000-volt circuits is 5 to 50 uV.
Default value: 5e-05; Min value: 0; Max value: 10.

See Also

.DC
.OPTION VNTOL
.TRAN

.OPTION ABSVAR

Sets the absolute limit for maximum voltage change between time points.

Syntax

```
.OPTION ABSVAR=<volts>
```

Default `0.5` (volts)

Description

Use this option to set the absolute limit for the maximum voltage change from one time point to the next. Use this option with `.OPTION DVDT`. If the simulator produces a convergent solution that is greater than `ABSVAR`, HSPICE discards the solution, sets the timestep to a smaller value and recalculates the solution. This is called a timestep reversal.

For additional information, see "DVDT Dynamic Timestep" in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.OPTION DVDT

.OPTION ABSVDC

Sets the minimum voltage for DC and transient analysis.

Syntax

```
.OPTION ABSVDC=volts
```

Default 50uV.

Description

Use this option to set the minimum voltage for DC and transient analysis. If accuracy is more critical than convergence, decrease ABSVDC. If you need voltages less than 50 uV, reduce ABSVDC to two orders of magnitude less than the smallest voltage. This ensures at least two digits of significance. Typically, you do not need to change ABSVDC unless you simulate a high-voltage circuit. For 1000-volt circuits, a reasonable value is 5 to 50 uV.

See Also

.DC
.OPTION VNTOL
.TRAN

.OPTION ACCT

Generates a detailed accounting report.

Syntax

```
.OPTION ACCT
.OPTION ACCT=[1|2]
```

Default 1

Arguments

Argument	Description
.OPTION ACCT	Enables reporting.
.OPTION ACCT=1 (default)	Is the same as ACCT without arguments.
.OPTION ACCT=2	Enables reporting and matrix statistic reporting.

Description

Use this option to generate a detailed accounting report.

Example

```
.OPTION ACCT=2
```

The ratio of `TOT.ITER` to `CONV.ITER` is the best measure of simulator efficiency. The theoretical ratio is 2:1. In this example the ratio is 2.57:1. SPICE generally has a ratio from 3:1 to 7:1.

In transient analysis, the ratio of `CONV.ITER` to `# POINTS` is the measure of the number of points evaluated to the number of points printed. If this ratio is greater than about 4:1, the convergence and time step control tolerances might be too tight for the simulation.

See Also

.DC
.TRAN

.OPTION ACCURATE

Selects a time algorithm for circuits such as high-gain comparators.

Syntax

```
.OPTION ACCURATE=[0|1]
```

Default 0

Description

Use this option to select a time algorithm that uses `LVLTIM=3` and `DVDT=2` for circuits such as high-gain comparators. Use this option with circuits that combine high gain and large dynamic range to guarantee accurate solutions in HSPICE. When set to `1`, this option sets these control options:

```
LVLTIM=3
DVDT=2
RELVAR=0.2
ABSVAR=0.2
FT=0.2
RELMOS=0.01
```

The default does not set the above control options.

In HSPICE RF, this option turns on `.OPTION FFT_ACCURATE` and is subordinate to `.OPTION SIM_ACCURACY`.

To see how use of the ACCURATE option impacts the value settings when used with .METHOD=GEAR, and other options, see Appendix B, How Options Affect other Options.

See Also

.OPTION ABSVAR
.OPTION DVDT
.OPTION FFT_ACCURATE
.OPTION FT
.OPTION LVLTIM
.OPTION METHOD
.OPTION RELMOS
.OPTION RELVAR
.OPTION SIM_ACCURACY

.OPTION ACOUT

Specifies the method for calculating differences in AC output values.

Syntax

```
.OPTION ACOUT=0|1
```

Default

1

Description

Use this option to specify a method for calculating the differences in AC output values for magnitude, phase, and decibels for prints and plots.

- ACOUT=1 selects the HSPICE method which calculates the *difference* of the magnitudes of the values real and imaginary.
- ACOUT=0: selects the SPICE method which calculates the *magnitude* of the differences real and imaginary in HSPICE. (Rarely used, but available for backward compatability.)

Examples

```
ACOUT=1
VR(N1,N2) = REAL [V(N1,0)] - REAL [V(N2,0)]
VI(N1,N2) = IMAG [V(N1,0)] - IMAG [V(N2,0)] Magnitude
VM(N1,0) = [VR(N1,0)^2 + VI(N1,0)^2]0.5
VM(N2,0) = [VR(N2,0)^2 + VI(N2,0)^2]0.5
VM(N1,N2) = VM(N1,0) - VM(N2,0)
Phase
VP(N1,0) = ARCTAN[VI(N1,0)/VR(N1,0)]
VP(N2,0) = ARCTAN[VI(N2,0)/VR(N2,0)]
VP(N1,N2) = VP(N1,0) - VP(N2,0)
Decibel
VDB(N1,N2) = 20 * LOG10(VM(N1,0)/VM(N2,0))

ACOUT=0
VR(N1,N2) = REAL [V(N1,0) - V(N2,0)]
VI(N1,N2) = IMAG [V(N1,0) - V(N2,0)]
Magnitude
VM(N1,N2) = [VR(N1,N2)^2+VI(N1,N2)^2]0.5 Phase
VP(N1,N2) = ARCTAN[VI(N1,N2)/VR(N1,N2)]
Decibel
VDB(N1,N2) = 20 * LOG10[VM(N1,N2)]
```

.OPTION ALTCC

Sets onetime reading of the input netlist for multiple `.ALTER` commands.

Syntax

```
.OPTION ALTCC=[-1|0|1]
```

Default

`0`

Description

Use this option to enable HSPICE to read the input netlist only once for multiple `.ALTER` commands.

- `ALTCC=1` reads input netlist only once for multiple `.ALTER` commands.
- `ALTCC=0` or `-1` disables this option. HSPICE does not output a warning message during transient analysis. Results are output following analysis.

`.OPTION ALTCC` or `.OPTION ALTCC=1` ignores parsing of an input netlist before an `.ALTER` command during standard cell library characterization only when an `.ALTER` command changes parameters, source stimulus, analysis, or passive elements. Otherwise, this option is ignored.

See Also

.ALTER
.LIB

.OPTION ALTCHK

Disables (or re-enables) topology checking in redefined elements (in altered netlists).

Syntax

```
.OPTION ALTCHK=0|1
```

Default

0

Description

By default, HSPICE automatically reports topology errors in the latest elements in your top-level netlist. It does not report errors in elements that you redefine by using the .ALTER command (altered netlist).

To enable topology checking redefined elements in the .ALTER block, set:

```
.OPTION ALTCHK=1
```

or

```
.OPTION ALTCHK
```

To disable topology checking in redefined elements (that is, to check topology only in the top-level netlist, not in the altered netlist), set:

```
.OPTION ALTCHK=0
```

See Also

.ALTER

.OPTION APPENDALL

Allows the top hierarchical level to use the .APPENDMODEL command even if the MOSFET model is embedded in a subcircuit.

Syntax

```
.OPTION APPENDALL
```

Description

Use this option when, for example, MOSFET model cards from fabs might be embedded in subcircuit definitions. The option ends the need to edit fab model files to include .APPENDMODEL commands in subcircuit definitions.

When this option is declared above the .APPENDMODEL command, then the main (uppermost) circuit level hierarchy can be used, even if the MOSFET model is embedded in a subcircuit. With this option, if the .APPENDMODEL command appears both in the main circuit and in a subcircuit, the .APPENDMODEL in the subcircuit takes priority.

Without this option, the rules of .APPENDMODEL remain unchanged.

Examples

In this example, the .APPENDMODEL in the main circuit is used.

```
.option appendall
.appendmodel n_ra mosra nch nmos
.SUBCKT mosra_test 1 2 3 4
M1 1 2 3 4 nch L=PL W=PW
.model nch nmos level= ...
.ENDS
```

In this example, the .APPENDMODEL in the subcircuit is used.

```
.option appendall
.appendmodel n_ra mosra nch nmos
.SUBCKT mosra_test 1 2 3 4
M1 1 2 3 4 nch L=PL W=PW
.model nch nmos level= ...
.appendmodel n_ra1 mosra nch nmos
.ENDS
```

See Also

.APPENDMODEL
.MODEL
.MOSRA

.OPTION ARTIST

Enables the Cadence™ Virtuoso® Analog Design Environment interface.

Syntax

```
.OPTION ARTIST=[0|1|2]
```

Default 0

Description

Enables the Virtuoso® Analog Design Environment if `ARTIST=2`. This option requires a specific license. For HSPICE RF, this option allows you to include HSPICE RF analyses such as Harmonic Balance, Shooting Newton, and their associated small-signal analyses and use their native waveform viewer. This option requires a specific license.

This option is generally used together with `.OPTION PSF`. If you use `.OPTION PSF=1` or `2` with `ARTIST=1` or `2` then the output format is always binary (Parameter Storage Format) and you have to use the Cadence ADE converter utility to change the binary format to ASCII format.

Note:

The PSF and SDA writers used in HSPICE rely on libraries that are not currently available in 64 bit versions, and 64 bit HSPICE cannot link the 32-bit libraries. If you inspect the log file, you will see the message `"**warning** 64bit cannot support option psf, artist or sda".`

The syntax is:

ADE_install_dir/platform`/tools/dfII/bin/psf -i` *input_file* `-o` *output_file*

See Also

.OPTION PSF

.OPTION ASPEC

Sets HSPICE or HSPICE RF to ASPEC-compatibility mode.

Syntax

```
.OPTION ASPEC=0|1
```

Default `0`

Description

Use this option to set the application to ASPEC-compatibility mode. When you set this option to `1`, the simulator reads ASPEC models and netlists, and the results are compatible.

If you set *ASPEC*, the following model parameters default to *ASPEC* values:

- ACM=1: Changes the default values for CJ, IS, NSUB, TOX, U0, and UTRA.
- Diode Model: TLEV=1 affects temperature compensation for PB.
- MOSFET Model: TLEV=1 affects PB, PHB, VTO, and PHI.
- SCALM, SCALE: Sets the model scale factor to microns for length dimensions.
- WL: Reverses implicit order for stating width and length in a MOSFET command. The default (WL=0) assigns the length first, then the width.

See Also

.OPTION SCALE
.OPTION SCALM
.OPTION WL

.OPTION AUTOSTOP (or) .OPTION AUTOST

Stops a transient analysis in HSPICE or HSPICE RF after calculating all TRIG-TARG, FIND-WHEN, and FROM-TO measure functions.

Syntax

```
.OPTION AUTOSTOP
-or-
.OPTION AUTOSTOP='expression'
```

Default 0

Description

Use this option to terminate a transient analysis in HSPICE after calculating all TRIG-TARG, FIND-WHEN, and FROM-TO measure functions. This option can substantially reduce CPU time. You can use the AUTOSTOP option with any measure type. You can also use the result of the preceding measurement as the next measured parameter.

When using .OPTION AUTOSTOP='expression', the 'expression' can only involve measure results, a logical AND (&&) or a logical OR(||). Using these types of expressions ends the simulation if any one of a set of .MEASURE commands succeeds, even if the others are not completed.

Also terminates the simulation after completing all .MEASURE commands. This is of special interest when testing corners.

Example

```
.option autostop='m1&&m2||m4'
.meas tran m1  trig v(bd_a0)  val='ddv/2'  fall=1  targ v(re_bd)
+ val='ddv/2'  rise=1
.meas tran m2  trig v(bd_a0)  val='ddv/2'  fall=2  targ v(re_bd)
+ val='ddv/2'  rise=2
.meas tran m3  trig v(bd_a0)  val='ddv/2'  rise=2  targ v(re_bd)
+ val='ddv/2'  rise=3
.meas tran m4  trig v(bd_a0)  val='ddv/2'  fall=3  targ v(re_bd)
+ val='ddv/2'  rise=4
.meas tran m5  trig v(bd_a0)  val='ddv/2' rise=3  targ v(re_bd)
+ val='ddv/2'  rise=5
```

In this example, when either m1 and m2 are obtained or just m4 is obtained, the transient analysis ends.

See Also

.MEASURE (or) .MEAS

.OPTION BA_ACTIVE

Specifies the active net filename for parasitic expansion.

Syntax

```
.OPTION BA_ACTIVE = <active_net_filename>
```

Description

Conducts selective parasitic expansion. The active net filename contains the chosen nets. If no file is specified, all the nets (nodes) are selected. `active_net_filename` is the name of the file that contains information about active nets. You must use this option with `BA_FILE`, or it has no effect.

Example

```
.option ba_active = "./hspice/NETLIST/DSPF/active.rcxt"
```

.OPTION BA_ERROR

Mode for handling error on nets.

Syntax

```
.OPTION BA_Error=0|1|2|3
```

Default `2` (LUMPCAP)

Description

Specifies means to handle an error on nets, where:

- 0: EXIT—Terminates the simulation with an error message
- 1: NO—Does not expand anything on the net.
- 2: LUMPCAP—Adds only the total lumped net capacitance
- 3: YES—Expands whatever can be expanded

Example

```
.option ba_error = 2
```

.OPTION BA_FILE

Launches full parasitic back-annotation.

Syntax

```
.OPTION BA_FILE = "filename"
```

Description

This file is the parasitic file name. `filename` is the name of the file that contains parasitic information in SPEF or DSPF format. This option launches back-annotation. Use `.OPTION BA_ACTIVE` with `.OPTION BA_FILE` to launch selective parasitic expansion. To view examples of the SPEF and DSPF file structures, see Post-Layout Back Annotation.

Example

```
.option ba_file = "./hspice/NETLIST/DSPF/add4.spf"
```

.OPTION BA_TERMINAL

Back annotation terminal mapping format.

Syntax

```
.OPTION BA_TERMINAL = "terminal_name_in_file
   recognized_name"
```

Description

Specifies the terminal mapping with the format: "terminal_name_in_file recognized_name".

Table 1 Default rules for element terminal names

Terminal Index	M, J Elements	Q Elements	R,C,D Elements
1	D*	C*	A*, P*
2	G*	B*	M*, B*, C*, N*
3	S*	E*	S*
4	B*	S*	n/a

Example

```
ba_terminal="my_drain drain"
```

.OPTION BADCHR

Generates a warning on finding a nonprintable character in an input file.

Syntax

```
.OPTION BADCHR=[0|1]
```

Default `0`

Description

Use this option to generate a warning on finding a nonprintable character in an input file by setting to `1`.

.OPTION BDFATOL

Sets the absolute tolerance for the global accuracy control of the Backward Differentiation Formulae integration method.

Syntax

```
.OPTION BDFATOL=val
```

Default

`1e-3`

Description

Use this option to set the absolute tolerance of the circuit convergence integration method BDF (a higher order integration algorithm than Backward-Euler, Gear, or Trapezoidal).

Any value other than 1e-3 is overridden if `.OPTION RUNLVL=3` (the default setting for `runlvl`). The value of the option appears in the *.lis* file.

Example

```
.OPTION METHOD=BDF
+.OPTIONS BDFATOL=1e-4 BDFRTOL=1e-4
```

See Also

.OPTION METHOD
.OPTION BDFRTOL

.OPTION BDFRTOL

Sets the relative tolerance for the global accuracy control of the Backward Differentiation Formulae integration method.

Syntax

```
.OPTION BDFATOL=val
```

Default

`1e-3`

Description

Use this option to set the relative tolerance of the circuit convergence integration method BDF (a higher order integration algorithm than Backward-Euler, Gear, or Trapezoidal).

Any value other than 1e-3 is overridden if `.OPTION RUNLVL=3` (the default setting for `runlvl`). The value of the option appears in the *.lis* file.

Example

```
.OPTION METHOD=BDF
+.OPTIONS BDFRTOL=1e-4 BDFATOL=1e-4
```

See Also

.OPTION METHOD
.OPTION BDFATOL

.OPTION BEEP

Enables or disables audible alert tone when simulation returns a message.

Syntax

```
.OPTION BEEP=[0|1]
```

Default `0`

Description

Use this option to enable or disable the audible alert tone when simulation returns a message.

- `BEEP=1` Turns on an audible tone when simulation returns a message (such as `HSPICE job completed`).
- `BEEP=0` Turns off the audible tone.

.OPTION BIASFILE

Sends `.BIASCHK` command results to a specified file.

Syntax

```
.OPTION BIASFILE=<filename>
```

Default *.lis

Description

Use this option to output the results of all `.BIASCHK` commands to a file that you specify. If you do not set this option, HSPICE outputs the `.BIASCHK` results to the *.lis file.

Example

```
.OPTION BIASFILE='biaschk/mos.bias'
```

See Also

.BIASCHK

.OPTION BIASINTERVAL

Controls the level of information output during transient analysis.

Syntax

```
.OPTION BIASINTERVAL=[0|1|2|3]
```

Example

```
.OPTION BIASINTERVAL=1
```

Default 0

Description

Use this option with the `.BIASCHK interval` argument to control the level of information output during transient analysis.

- `BIASINTERVAL=0`: Ignores the `interval` argument.
- `BIASINTERVAL=1`: Output the total number of suppressed violation regions for those elements being monitored. Violation warning messages that are generated in these suppressed regions are removed from the output.
- `BIASINTERVAL=2`: Output detailed information regarding suppressed violation regions. This includes element information, start time, stop time, and peak values. Also, violation warning messages that are generated in these suppressed regions are removed from the output.
- `BIASINTERVAL=3`: Output detailed information about all violation regions. Also, violation warning messages that are generated in these regions are removed from the output.

See Also

.BIASCHK

.OPTION BIASNODE

Specifies whether to use node names or port names in element commands.

Syntax

```
.OPTION BIASNODE=[0|1]
```

Description

Use this option to specify whether to use node names or port names in element commands in `.BIASCHK` warning messages.

- `BIASNODE=1`: use node names instead of port names
- `BIASNODE=0`: use port names (for example, ng of MOS element)

Example

```
.OPTION BIASNODE=1
```

Default `0`

See Also

.BIASCHK

.OPTION BIASPARALLEL

Controls whether .BIASCHK sweeps the parallel elements being monitored.

Syntax

```
.OPTION BIASPARALLEL=[0|1]
```

Default 0

Description

Use this option with the .BIASCHK mname argument to control whether .BIASCHK sweeps the parallel elements being monitored.

- BIASPARALLEL=1: sweep parallel elements. If node voltage is also being monitored, only the first element is used to generate warning messages.
- BIASPARALLEL=0: do not sweep parallel elements.

Example

```
.OPTION BIASPARALLEL=1
```

See Also

.BIASCHK

.OPTION BIAWARN

Controls whether HSPICE outputs warning messages when local max bias voltage exceeds limit during transient analysis.

Syntax

```
.OPTION BIAWARN=[0|1]
```

Default

`0`

Description

Use this option to control whether HSPICE outputs warning messages when a local max bias voltage exceeds the limit during transient analysis.

- `BIAWARN=1`: Output warning messages. When transient analysis is completed, the results are output as filtered by noise.
- `BIAWARN=0`: Do not output a warning message. When the transient analysis is completed, output the results.

Example

```
.OPTION BIAWARN=1
```

See Also

.TRAN

.OPTION BINPRNT

Outputs the binning parameters of the CMI MOSFET model.

Syntax

```
.OPTION BINPRNT
```

Default 0

Description

Use this option to output the binning parameters of the CMI MOSFET model. Currently available only for Level 57.

.OPTION BPNMATCHTOL

Determines the minimum required match between the NLP and PAC phase noise algorithms in HSPICE RF.

Syntax

```
.OPTION BPNMATCHTOL=val
```

Default `0.5`dB

Description

Use this option to determines the minimum required match between the NLP and PAC phase noise algorithms. An acceptable range is 0.05dB to 5dB.

See Also

.OPTION PHASENOISEKRYLOVDIM
.OPTION PHASENOISEKRYLOVITER
.OPTION PHASENOISETOL
.OPTION PHNOISELORENTZ

.OPTION BRIEF

Stops echoing (printback) of data file to stdout until HSPICE reaches an `.OPTION BRIEF=0` or `.END` command.

Syntax

```
.OPTION BRIEF=[0|1]
```

Default

`0`

Description

Use this option to terminate echoing (printback) of the data file to *stdout* until HSPICE finds an `.OPTION BRIEF=0` or the `.END` command. It also resets the `LIST`, `NODE` and `OPTS` options, and sets `NOMOD`. `BRIEF=0` enables printback. The `NXX` option is the same as `BRIEF`. `BRIEF=1` disables printback. `.OPTION BRIEF=1` and .`OPTION BRIEF=0` act similar to the commands `.PROTECT` and `.UNPROTECT`, respectively.

For information on how `BRIEF` impacts other options, see Appendix B, How Options Affect other Options.

See Also

.END
.OPTION LIST
.OPTION NODE
.OPTION NXX
.OPTION OPTS
.PROTECT or .PROT
.UNPROTECT or .UNPROT

.OPTION BSIM4PDS

Flag to control the BSIM4 Ps_{eff} (effective source perimeter) and Pd_{eff} (effective drain perimeter) model equation calculation.

Syntax

```
.OPTION BSIM4PDS=0|1
```

Default

`0`

Description

Setting `BSIM4PDS=1` enhances the ps_{eff} and pd_{eff} calculation, so that when the calculated ps_{eff} and pd_{eff} is negative, HSPICE uses the PAeffGeo function to recalculate it. (This option solves the issue of negative ps_{eff} and pd_{eff} causing potential non-convergence issues.)

When `BSIM4PDS=0`, HSPICE strictly follows the UCB code, and results in no recalculation if negative ps_{eff} or pd_{eff} occurs.

Note:

This option is only available for BSIM4 (Level 54).

.OPTION BYPASS

Bypasses model evaluations if the terminal voltages stay constant.

Syntax

```
.OPTION BYPASS=[0|1|2]
```

Default `1` for MESFETs, JFETs, or BJTs; `2` for MOSFETs and diodes

Description

Use this option to bypass model evaluations if the terminal voltages do not change. Values can be `0` (off), `1` (on), or `2` (advanced algorithm, applies to BSIM3v3, BSIM4, BSIM3SOI (LEVEL=57), BSIM4SOI (LEVEL 70), HVMOS (LEVEL 66), and PSP (LEVEL=69) MOSFETs in special cases).

To speed up simulation, `BYPASS=1` does not update the status of latent devices. `BYPASS=2` uses linear prediction to update the devices and balance speed and accuracy.

(Assuming `BYPASS` is not explicitly set otherwise): When the `BYPASS` option is not given in the netlist, its value is determined by the value of `RUNLVL` and `ACCURATE`. When `RUNLVL=0` then `BYPASS=1`; when `RUNLVL=0` + `ACCURATE=1` then `BYPASS=0`; when `RUNLVL=1` through `6`, then `BYPASS=2`.

See Also

- .OPTION ACCURATE
  .OPTION RUNLVL

.OPTION BYTOL

Sets a voltage tolerance at which a MOSFET, MESFET, JFET, BJT, or diode becomes latent.

Syntax

```
.OPTION BYTOL=x
```

Default `100.00u`

Description

Use this option to specify a voltage tolerance at which a MOSFET, MESFET, JFET, BJT, or diode becomes latent. HSPICE does not update status of latent devices. The default=`MBYPASS` x `VNTOL`.

See Also

.OPTION MBYPASS
.OPTION VNTOL

.OPTION CAPTAB

Adds up all the capacitances attached to a node and prints a table of single-plate node capacitances.

Syntax

```
.OPTION CAPTAB
```

Default 0

Description

Use this option to print a compiled table of single-plate node capacitances for diodes, BJTs, MOSFETs, JFETs, and passive capacitors at each operating point.

.OPTION CHGTOL

Sets a charge error tolerance.

Syntax

```
.OPTION CHGTOL=x
```

Default `1.00f`

Description

Use this option to set a charge error tolerance if you set `LVLTIM=2`. Use `CHGTOL` with `RELQ` to set the absolute and relative charge tolerance for all HSPICE capacitances. The default is `1e-15` (coulomb).
Min value: 1e-20; Max value: 10.

See Also

.OPTION CHGTOL
.OPTION LVLTIM
.OPTION RELQ

.OPTION CMIFLAG

Loads and links the dynamically linked Common Model Interface (CMI) library.

Syntax

```
.OPTION CMIFLAG=0|1
```

Description

Use this option to load and link the compiled CMI object *.so* file to HSPICE/ HSPICE RF during simulation runs. If this option parameter is set with no value or to 1, then the CMI *.so* file is loaded as a dynamically-linked object file.

If this option parameter does not exist (deemed as default) in the netlist, or is explicitly set to 0, no loading or linking will take place.

See Also

.OPTION CUSTCMI

.OPTION CMIPATH

Enables automatic selection of correct Custom CMI *.so* library platform.

Syntax

```
.OPTION CMIPATH='LIB_DIRECTORY'
```

Description

This option allows you to automatically select the correct custom CMI *.so* library platform, even though you might not have the right information about the platform HSPICE is running on. This functionality eliminates the need to manually search for the correct platform and allows for efficient CMI *.so* library distribution and customer applications. The solution to this issue keeps the environment variable `hspice_lib_models` backward compatible in its usage model, but users can add the control option `.OPTION CMIPATH='LIB_DIRECTORY'` to the model file.

For the UNIX OS, HSPICE provides two scripts, `hspice` and `hspice64` to invoke the right HSPICE executable for the platform on which HSPICE is being invoked to run. These scripts are enhanced to recognize the correct machine and platform for automatic CMI *.so* library selection. For the Windows OS, no HSPICE script is required, since all Windows platforms share the same single CMI .so library:*LIB_DIRECTORY*/WIN for all Windows platforms.

For information on the HSPICE CMI, contact your Synopsys technical support team.

.OPTION CMIUSRFLAG

Flag to control .OPTION SCALE parsing into the External Common Model Interface (CMI).

Syntax

```
.OPTION CMIUSRFLAG=0|1
```

Default 0

Description

Flag to control the CMI element instance parameter value (unit) scaling. It permits users and/or foundry model development teams to choose the desired scaling for the instance parameters of the MOSFET devices that call a foundry's CMI model libraries.

When this option parameter is set with no value or to 1, the products of option parameters SCALE and GEOSHRINK are passed and made available to scale the CMI model instance parameter values.

If the CMIUSRFLAG option parameter does not exist in the netlist (default), or is explicitly set to 0, then the option parameters SCALE and GEOSHRINK are not accessible in the CMI; and the element instance parameter scaling is not activated for the foundry CMI models and libraries.

Note:

The CMIUSRFLAG option is only available for CMI MOS Level 101.

Example

In this example, the value scale*geoshrink=0.9e-6 is parsed to the external CMI.

```
.option cmiflag=1
.option scale=1e-6 geoshrink=0.9 cmiusrflag=1
...
.model nch nmos level=101 ...
```

See Also

- .OPTION SCALE
- .OPTION GEOSHRINK

.OPTION CONVERGE

Invokes different methods for solving nonconvergence problems.

Syntax

```
.OPTION CONVERGE=[-1|0|1|2|3|4|5]
```

Default 0

Description

Use this option to run different methods for solving nonconvergence problems.

Note:

In HSPICE RF, this option is ignored because it is replaced by automated algorithms.

- CONVERGE=-1: Use with DCON=-1 to disable autoconvergence.
- CONVERGE=0: Autoconvergence.
- CONVERGE=1: Use the Damped Pseudo Transient algorithm. If simulation does not converge within the set CPU time (in the CPTIME control option), then simulation halts.
- CONVERGE=2: Use a combination of DCSTEP and GMINDC ramping. Not used in the autoconvergence flow.
- CONVERGE=3: Invoke the source-stepping method. Not used in the autoconvergence flow.
- CONVERGE=4: Use the gmath ramping method.
- CONVERGE=5: Use the gshunt ramping method. Even you did not set it in an .OPTION command, the CONVERGE option activates if a matrix floating-point overflows or if HSPICE reports a “timestep too small” error. The default is 0. If a matrix floating-point overflows, then CONVERGE=1.

See Also

.OPTION DCON
.OPTION DCSTEP
.OPTION DCTRAN
.OPTION GMINDC

.OPTION CPTIME

Sets the maximum CPU time allotted for a simulation.

Syntax

```
.OPTION CPTIME=x
```

Default `10.00x`

Description

Use this option to set the maximum CPU time, in seconds, allotted for this simulation job. When the time allowed for the job exceeds `CPTIME`, HSPICE prints or plots the results up to that point and concludes the job. Use this option if you are uncertain how long the simulation takes, especially when you debug new data files. The default is `1e7` (400 days).

.OPTION CSCAL

Sets the capacitance scale for Pole/Zero analysis.

Syntax

```
.OPTION CSCAL=x
```

Default `1.0e+12`

Description

Use this option to set the capacitance scale for Pole/Zero analysis. HSPICE multiplies capacitances by `CSCAL`.

See Also

.OPTION FMAX
.OPTION GSCAL
.OPTION ITLPZ
.OPTION LSCAL
.OPTION PZABS

.OPTION CSDF

Selects the Common Simulation Data Format (Viewlogic-compatible graph data file format).

Syntax

```
.OPTION CSDF=0|1
```

Default `0`

Description

Use this option to specify whether HSPICE/ HSPICE RF outputs CSDF data when you run a HSPICE simulation.

- If `CSDF=0`, CSDF output is disabled.
- If `CSDF=1`, HSPICE produces CSDF output.

See Also

.OPTION POST

.OPTION CSHDC

Adds capacitance from each node to ground; used only with the `CONVERGE` option.

Syntax

```
.OPTION CSHDC=x
```

Default `1.00p`

Description

Use this option to add capacitance from each node to ground. This is the same option as `CSHUNT`; use CSHDC only with the `CONVERGE` option. When defined, .OPTION CSHDC is the same as .OPTION CSHUNT, except that CSHDC becomes invalid after DC OP analysis, while CSHUNT stays in both DC OP and transient analysis.

See Also

.OPTION CONVERGE
.OPTION CSHUNT

.OPTION CSHUNT

Adds capacitance from each node to ground.

Syntax

```
.OPTION CSHUNT=x
```

Default 0

Description

Use this option to add capacitance from each node to ground. Add a small CSHUNT to each node to solve internal “timestep too small” timestep problems caused by high frequency oscillations or numerical noise. When defined, .OPTION CSHUNT is the same as .OPTION CSHDC, except that CSHDC becomes invalid after DC OP analysis, while CSHUNT stays in both DC OP and transient analysis.

See Also

.OPTION CSHDC
.OPTION GSHUNT

.OPTION CUSTCMI

Turns on gate direct tunneling current modeling and instance parameter support.

Syntax

```
.OPTION CUSTCMI=x
```

Default 0

Description

Use this option to turn on gate direct tunneling current modeling and instance parameter support. You set `.OPTION CUSTCMI=1` jointly with `.OPTION CMIFLAG` to turn on gate direct tunneling current modeling and instance parameter support for customer CMI. `.OPTION CUSTCMI=0` to turns off that feature.

See Also

.OPTION CMIFLAG

.OPTION CVTOL

Changes the number of numerical integration steps when calculating the gate capacitor charge for a MOSFET.

Syntax

```
.OPTION CVTOL=x
```

Default `200.00m`

Description

Use this option to change the number of numerical integration steps when calculating the gate capacitor charge for a MOSFET by using `CAPOP=3`. See the discussion of `CAPOP=3` in the "Overview of MOSFET Models" chapter of the *HSPICE Reference Manual: MOSFET Models* for explicit equations and discussion.

.OPTION D_IBIS

Specifies the directory containing the IBIS files.

Syntax

```
.OPTION D_IBIS='ibis_files_directory'
```

Description

Use this option to specify the directory containing the IBIS files. If you specify several directories, the simulation looks for IBIS files in the local directory (the directory from which you run the simulation). It then checks the directories specified through `.OPTION D_IBIS` in the order that `.OPTION` cards appear in the netlist. You can use the `D_IBIS` option to specify up to 40 directories.

Example

```
.OPTION d_ibis='/home/user/ibis/models'
```

.OPTION DCAP

Specifies equations used to calculate depletion capacitance for Level 1 and 3 diodes and BJTs.

Syntax

```
.OPTION DCAP
```

Default

2

Description

Use this option to specify equations for HSPICE to use when calculating depletion capacitance for Level 1 and 3 diodes and BJTs. The *HSPICE Reference Manual: Elements and Device Models* describes these equations in the section Using Diode Capacitance Equations.

.OPTION DCCAP

Generates C-V plots.

Syntax

```
.OPTION DCCAP=0|1
```

Default `0` (off)

Description

Use this option to generate C-V plots. Prints capacitance values of a circuit (both model and element) during a DC analysis. You can use a DC sweep of the capacitor to generate C-V plots. If not set, MOS device or voltage-variable capacitance values are not evaluated and the printed value ise zero.

See Also

.DC

.OPTION DCFOR

Sets the number of iterations to calculate after a circuit converges in the steady state.

Syntax

```
.OPTION DCFOR=x
```

Default 0

Description

Use this option to set the number of iterations to calculate after a circuit converges in the steady state. The number of iterations after convergence is usually zero, so DCFOR adds iterations (and computation time) to the DC circuit solution. DCFOR ensures that a circuit actually, not falsely, converges.

Use this option with .OPTION DCHOLD and the .NODESET command to enhance DC convergence.

See Also

.DC
.NODESET
.OPTION DCHOLD

.OPTION DCHOLD

Specifies how many iterations to hold a node at the .NODESET voltage values.

Syntax

.OPTION DCHOLD=*n*

Default 1

Description

Note:

In HSPICE RF, this option is ignored; it is replaced by automated algorithms.

Use this option to specify how many iterations to hold a node at the .NODESET voltage values.

Use DCFOR and DCHOLD together to initialize DC analysis. DCFOR and DCHOLD enhance the convergence properties of a DC simulation. DCFOR and DCHOLD work with the .NODESET command. The effects of DCHOLD on convergence differ, according to the DCHOLD value and the number of iterations before DC convergence.

If a circuit converges in the steady state in fewer than DCHOLD iterations, the DC solution includes the values set in .NODESET.

If a circuit requires more than DCHOLD iterations to converge, HSPICE ignores the values set in the .NODESET command, and calculates the DC solution by setting the .NODESET fixed-source voltages as open circuited.

See Also

.DC
.NODESET
.OPTION DCFOR

.OPTION DCIC

Specifies whether to use or ignore `.IC` commands in the netlist.

Syntax

```
.OPTION DCIC=0|1
```

Default

`1`

Description

Use this option to specify whether to use or ignore `.IC` commands in the netlist.

- `DCIC=1` (default): Each point in a DC sweep analysis acts like an operating point and all `.IC` commands in the netlist are used.
- `DCIC=0`: `.IC` commands in the netlist are ignored for DC sweep analysis.

See Also

.IC
.DC

.OPTION DCON

Disables autoconvergence (when `DCON=-1` and `CONVERGE=-1`).

Syntax

```
.OPTION DCON=x
```

Default

`0`

Description

If a circuit cannot converge, HSPICE automatically sets `DCON=1` and calculates the following:

$$DV = max\left(0.1,\ \frac{V_{max}}{50}\right)$$, if DV =1000

$$GRAMP = max\left(6,\ \log_{10}\left(\frac{I_{max}}{GMINDC}\right)\right) \qquad ITL1 = ITL1 + 20 \cdot GRAMP$$

V_{max} is the maximum voltage and I_{max} is the maximum current.

- If the circuit still cannot converge, HSPICE sets `DCON=2`, which sets `DV=1e6`.
- If the circuit uses discontinuous models or uninitialized flip-flops, simulation might not converge. Set `DCON=-1` and `CONVERGE=-1` to disable autoconvergence. HSPICE lists all nonconvergent nodes and devices.

See Also

.OPTION CONVERGE
.OPTION DV

.OPTION DCSTEP

Converts DC model and element capacitors to a conductance.

Syntax

```
.OPTION DCSTEP=n
```

Default `0`(seconds)

Description

Use this option to convert DC model and element capacitors to a conductance to enhance DC convergence properties. HSPICE divides the value of the element capacitors by `DCSTEP` to model DC conductance.

See Also

.DC

.OPTION DCTRAN

Invokes different methods to solve nonconvergence problems.

Syntax

```
.OPTION DCTRAN=x
```

Default

o

Description

Use this option to run different methods to solve nonconvergence problems. `DCTRAN` is an alias for `CONVERGE`.

See Also

.OPTION CONVERGE

.OPTION DEFAD

Sets the default MOSFET drain diode area.

Syntax

```
.OPTION DEFAD=0|1
```

Default 0

Description

Use this option to set the default MOSFET drain diode area.

.OPTION DEFAS

Sets the default MOSFET source diode area.

Syntax

```
.OPTION DEFAS=x
```

Default `0`

Description

Use this option to set the default MOSFET source diode area.

.OPTION DEFL

Sets the default MOSFET channel length.

Syntax

```
.OPTION DEFL=x
```

Default `100.00u` (1e-4m)

Description

Use this option to set the default MOSFET channel length.

.OPTION DEFNRD

Sets the default number of squares for the drain resistor on a MOSFET.

Syntax

```
.OPTION DEFNRD=n
```

Default 0

Description

Use this option to set the default number of squares for the drain resistor on a MOSFET.

.OPTION DEFNRS

Sets the default number of squares for the source resistor on a MOSFET.

Syntax

```
.OPTION DEFNRS= n
```

Default 0

Description

Use this option to set the default number of squares for the source resistor on a MOSFET.

.OPTION DEFPD

Sets the default MOSFET drain diode perimeter.

Syntax

```
.OPTION DEFPD=n
```

Default 0

Description

Use this option to set the default MOSFET drain diode perimeter.

.OPTION DEFPS

Sets the default MOSFET source diode perimeter.

Syntax

```
.OPTION DEFPS=x
```

Default 0

Description

Use this option to set the default MOSFET source diode perimeter.

.OPTION DEFSA

Sets the default BSIM4 MOSFET SA parameter in HSPICE.

Syntax

```
.OPTION DEFSA=x
```

Default `0.0`

Description

Use this option to set the default distance between the S/D diffusion edge to the poly gate edge from one side in the BSIM STI/LOD model.

.OPTION DEFSB

Sets the default BSIM4 MOSFET SB parameter.

Syntax

```
.OPTION DEFSB=x
```

Default `0.0`

Description

Use this option to set the default distance between the S/D diffusion edge to the poly gate edge from side opposite the SA side in the BSIM STI/LOD model.

.OPTION DEFSD

Sets default for BSIM4 MOSFET SD parameter.

Syntax

```
.OPTION DEFSD=x
```

Default `0.0`

Description

Use this option to set the default for the distance between neighboring fingers (SD parameter) in a BSIM STI/LOD model.

.OPTION DEFW

Sets the default MOSFET channel width.

Syntax

```
.OPTION DEFW=x
```

Default `100.00u`

Description

Use this option to set the default MOSFET channel width. The default is `1e-4m`.

.OPTION DELMAX

Sets the maximum allowable step size of the timesteps taken during transient analysis in HSPICE/HSPICE RF.

Syntax

```
.OPTION DELMAX=x
```

Default (Computed automatically)

Description

Use this option to set the maximum allowable step size of the internal timestep. The maximum internal timestep taken by HSPICE during transient analysis is referred to as Δt_{max}. Its value is normally computed automatically based on several timestep control settings. If you wish to override the automatically computed value, and force the maximum step size to be a specific value, you can do so with `.OPTION DELMAX`, or by specifying a *delmax* value with the `.TRAN` command. If not specified, HSPICE automatically computes a `DELMAX` "auto" value, based on timestep control factors such as `FS` and `RMAX`. (For a complete list of timestep control factors, see Timestep Control for Accuracy in the Transient Analysis chapter of the *HSPICE User Guide: Simulation and Analysis*.)

The initial calculated `DELMAX` "auto" value, shown in the output listing, is generally not the value used for simulation. The calculated `DELMAX` value is automatically adjusted by the timestep control methods, `DVDT`, `RUNLVL` and `LVLTIM`.

If `DELMAX` is defined in an `.OPTION` command, its priority is higher than the value given with a `.TRAN` command and it overrides the `DELMAX` "auto" value calculations. Min value: -1e10; Max value 1e10.

See Also

.TRAN
.OPTION DVDT
.OPTION RUNLVL
.OPTION LVLTIM
.OPTION FS
.OPTION RMAX
Appendix B, How Options Affect other Options

.OPTION DI

Sets the maximum iteration to iteration current change in HSPICE.

Syntax

```
.OPTION DI=n
```

Default `100.00`

Description

Use this option to set the maximum iteration to iteration current change through voltage-defined branches (voltage sources and inductors). Use this option only if the value of the `ABSH` control option is greater than 0.

See Also

.OPTION ABSH

.OPTION DIAGNOSTIC (or) .OPTION DIAGNO

Logs the occurrence of negative model conductances.

Syntax

```
.OPTION DIAGNOSTIC
```

Description

Use this option to log the occurrence of negative model conductances.

.OPTION DLENCSDF

Specifies how many digits to include in scientific notation (exponents) or to the right of the decimal point when using Common Simulation Data Format.

Syntax

```
.OPTION DLENCSDF=x
```

Default

5

Description

If you use the Common Simulation Data Format (Viewlogic graph data file format) as the output format, this digit length option specifies how many digits to include in scientific notation (exponents) or to the right of the decimal point. Valid values are any integer from 1 to 10.

If you assign a floating decimal point or if you specify less than 1 or more than 10 digits, HSPICE uses the default. For example, it places 5 digits to the right of a decimal point.

.OPTION DV

Specifies maximum iteration to iteration voltage change for all circuit nodes in both DC and transient analyses.

Syntax

```
.OPTION DV=x
```

Default `1.00k`

Description

Use this option to specify maximum iteration to iteration voltage change for all circuit nodes in both DC and transient analysis. High-gain bipolar amplifiers can require values of 0.5 to 5.0 to achieve a stable DC operating point. Large CMOS digital circuits frequently require about 1 V. The default is 1000 (or 1e6 if `DCON=2`).

See Also

.DC
.OPTION DCON
.TRAN

.OPTION DVDT

Adjusts the timestep based on rates of change for node voltage.

Syntax

```
.OPTION DVDT=0|1|2|3|4
```

Default 4

Description

Use this option to adjust the timestep based on rates of change for node voltage.

- 0: Original algorithm
- 1: Fast
- 2: Accurate
- 3, 4: Balance speed and accuracy
- The ACCURATE option also increases the accuracy of the results.

For additional information, see "DVDT Dynamic Timestep" in the *HSPICE User Guide: Simulation and Analysis*.

For information on how DVDT values impact other options, see Appendix B, How Options Affect other Options.

See Also

.OPTION ACCURATE
.OPTION DELMAX

.OPTION DVTR

Limits the voltage in transient analysis.

Syntax

```
.OPTION DVTR=x
```

Default `1.00k`

Description

Use this option to limit the voltage in transient analysis. The default is `1000`.

.OPTION EM_RECOVERY

Provides a coefficient value for measuring “recovered” average current such as electromigration for bipolar currents.

Syntax

```
.OPTION EM_RECOVERY=value
```

Default

`1`

Description

This option is used in a transient analysis with the `.MEAS` keyword `em_avg` (electromigration average) using the From-To function. `.OPTION EM_RECOVERY` assists in measuring “recovered” average current from an electromigration perspective. The option can have a coefficient value between 0.0 and 1.0. Recovered average current is especially meaningful for bipolar currents (for example output of the inverter), as the mathematical average for such a waveform is zero.

Example

```
.option em_recovery=0.9
```

See Also

.MEASURE (AVG, EM_AVG, INTEG, MIN, MAX, PP, and RMS)

.OPTION EPSMIN

Specifies the smallest number a computer can add or subtract.

Syntax

```
.OPTION EPSMIN=x
```

Default `1e-28`

Description

Use this option to specify the smallest number that a computer can add or subtract, a constant value. This options helps avoid zero denominator issues.

.OPTION EXPLI

Enables the current-explosion model parameter.

Syntax

```
.OPTION EXPLI=x
```

Default `0` (amp/area effective)

Description

Use this option to enable the current-explosion model parameter. PN junction characteristics, above the explosion current are linear. HSPICE/HSPICE RF determines the slope at the explosion point. This improves simulation speed and convergence.

.OPTION EXPMAX

Specifies the largest exponent that you can use for an exponential before overflow occurs.

Syntax

```
.OPTION EXPMAX=x
```

Default `80.00`

Description

Use this option to specify the largest exponent that you can use for an exponential before overflow occurs. Typical value for an IBM platform is 350.

.OPTION FAST

Disables status updates for latent devices; this speeds up simulation.

Syntax

```
.OPTION FAST
```

Default 0

Description

Use this option to set additional options, which increase simulation speed with minimal loss of accuracy.

To speed up simulation, this option disables status updates for latent devices. Use this option for MOSFETs, MESFETs, JFETs, BJTs, and diodes.

A device is latent if its node voltage variation (from one iteration to the next) is less than the value of either the BYTOL control option or the BYPASSTOL element parameter. (If FAST is on, HSPICE sets BYTOL to different values for different types of device models.)

Besides the FAST option, you can also use the NOTOP and NOELCK options to reduce input preprocessing time. Increasing the value of the MBYPASS or BYTOL option, also helps simulations to run faster, but can reduce accuracy. To see how use of FAST impacts the value settings of other options, see Appendix B, How Options Affect other Options.

See Also

.OPTION BYTOL
.OPTION MBYPASS
.OPTION NOELCK
.OPTION NOTOP

.OPTION FFT_ACCURATE

Dynamically adjusts the time step so that each FFT point is a real simulation point in HSPICE/HSPICE RF.

Syntax

```
.OPTION FFT_ACCURATE=x
```

Default 0

Description

Use this option to dynamically adjust the time step so that each FFT point is a real simulation point. This eliminates interpolation error and provides the highest FFT accuracy with minimal overhead in simulation time.

See Also

.OPTION ACCURATE
.OPTION SIM_ACCURACY (RF)

.OPTION FFTOUT

Prints 30 harmonic fundamentals.

Syntax

```
.OPTION FFTOUT=0|1
```

Default

`0`

Description

Use this option to print 30 harmonic fundamentals sorted by size, THD, SNR, and SFDR, but only if you specify a `FFTOUT` option and a `.FFT freq=xxx` command.

See Also

.FFT

.OPTION FMAX

Sets the maximum frequency value of angular velocity, for poles and zeros.

Syntax

```
.OPTION FMAX=x
```

Default `1.0e+12`

Description

Use this option to set the maximum frequency value of angular velocity for Pole/Zero analysis. The units of value are in rad/sec.

See Also

.OPTION CSCAL
.OPTION FSCAL
.OPTION GSCAL
.OPTION ITLPZ
.OPTION LSCAL
.OPTION PZABS
.OPTION PZTOL
.OPTION RITOL
.PZ

.OPTION FS

Decreases FS value to help circuits that have timestep convergence difficulties.

Syntax

```
.OPTION FS=x
```

Default 250.00m

Description

Use this option to decrease delta (internal timestep) by the specified fraction of a timestep (TSTEP) for the first time point of a transient. Decreases the FS value to help circuits that have timestep convergence difficulties. DVDT=3 uses FS to control the timestep. $Delta = FS \cdot [MIN(TSTEP, DELMAX, BKPT)]$

- You specify DELMAX.
- BKPT is related to the breakpoint of the source.
- The .TRAN command sets TSTEP.

See Also

.OPTION DELMAX
.OPTION DVDT
.TRAN

.OPTION FSCAL

Sets the frequency scale for Pole/Zero analysis.

Syntax

```
.OPTION FSCAL=x
```

Default `1e-9`

Description

Use this option to set the frequency scale for Pole/Zero analysis. HSPICE multiplies capacitances by `FSCAL`.

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION GSCAL
.OPTION ITLPZ
.OPTION LSCAL
.OPTION PZABS
.OPTION PZTOL
.OPTION RITOL
.PZ

.OPTION FT

Decreases delta by a specified fraction of a timestep for iteration set that does not converge.

Syntax

```
.OPTION FT=x
```

Default

```
0.25
```

Description

Use this option to decrease delta (the internal timestep) by a specified fraction of a timestep (`TSTEP`) for an iteration set that does not converge. If `DVDT=2` or `DVDT=4`, `FT` controls the timestep.

See Also

.OPTION DVDT
.TRAN

.OPTION GDCPATH

Adds conductance to nodes having no DC path to ground.

Syntax

```
.OPTION GDCPATH[=x]
```

Default `1e-12`

Description

Use this option to add conductance to nodes having no DC path to ground. You use this option to help solve no DC path to ground problems. If you specify `GDCPATH` in a netlist without a value that value is assumed to be `1e-12`.

.OPTION GENK

Automatically computes second-order mutual inductance for several coupled inductors.

Syntax

```
.OPTION GENK= 0|1
```

Default 1

Description

Use this option to automatically calculate second-order mutual inductance for several coupled inductors. The default (1) enables the calculation.

.OPTION GEOSHRINK

Element scaling factor used with .OPTION SCALE.

Syntax

```
.OPTION GEOSHRINK=X
```

Default 1.00

Description

Use this option as a global model to apply to all elements. In addition to .OPTION SCALE, use this option to further scale geometric element instance parameters whose default units are meters. The final instance geometric parameters are then be calculated as:

```
final_dimension = original_dimension * SCALE * GEOSHRINK
```

The effective scaling factor is the product of the two parameters; HSPICE uses scale*geoshrink to scale the parameters/dimensions.

The default value for both SCALE and GEOSHRINK is 1.

If a model library contains devices other that MOSFET, such as R, L, C, diode, bjt... etc., and/or the netlist is a post-layout design with RCs, the shrink factor is applied to all elements.

Examples

Example 1: If there is more than one geoshrink option set, only the last geoshrink is used.

```
.option geoshrink=0.8
.option geoshrink=0.9
```

Then the final_dimension = original_dimension * SCALE * 0.9

Example 2: If there is more than one geoshrink and scale in the model card, only the last scale and the last geoshrink are used.

```
.option scale=2u
.option scale=1u
.option geoshrink=0.8
.option geoshrink=0.9
```

Then the final_dimension = original_dimension * 1u * 0.9

See Also

.OPTION SCALE
.OPTION CMIUSRFLAG

.OPTION GMAX

Specifies the maximum conductance in parallel with a current source for `.IC` and `.NODESET` initialization circuitry.

Syntax

```
.OPTION GMAX=x
```

Default `100.00` (mho)

Description

Use this option to specify the maximum conductance in parallel with a current source for `.IC` and `.NODESET` initialization circuitry. Some large bipolar circuits require you to set `GMAX=1` for convergence.

See Also

.IC
.NODESET

.OPTION GMIN

Specifies the minimum conductance added to all PN junctions for a time sweep in transient analysis for HSPICE/HSPICE RF.

Syntax

```
.OPTION GMIN=x
```

Default `1.00p`

Description

Use this option to specify the minimum conductance added to all PN junctions for a time sweep in transient analysis.
Min value: 1e-30; Max value: 100.

See Also

.OPTION GMINDC

.OPTION GMINDC

Specifies conductance in parallel for PN junctions and MOSFET nodes in DC analysis.

Syntax

```
.OPTION GMINDC=x
```

Default `1.00p`

Description

Use this option to specify conductance in parallel for all PN junctions and MOSFET nodes except gates in DC analysis. `GMINDC` helps overcome DC convergence problems caused by low values of off-conductance for pn junctions and MOSFETs. You can use `GRAMP` to reduce `GMINDC` by one order of magnitude for each step. Set `GMINDC` between `1e-4` and the `PIVTOL` value. Min value: 1e-30; Max value: 100.

Large values of `GMINDC` can cause unreasonable circuit response. If your circuit requires large values to converge, suspect a bad model or circuit. If a matrix floating-point overflows and if `GMINDC` is `1.0e-12` or less, HSPICE sets it to `1.0e-11`. HSPICE manipulates `GMINDC` in auto-converge mode.

See Also

.DC
.OPTION GRAMP
.OPTION PIVTOL

.OPTION GRAMP

Specifies a conductance range over which DC operating point analysis sweeps GMINDC.

Syntax

```
.OPTION GRAMP=x
```

Default 0

Description

Use this option to specify a conductance range over which the DC operating point analysis sweeps GMINDC. HSPICE sets this value during auto-convergence. Use GRAMP with the GMINDC option to find the smallest GMINDC value that results in DC convergence.

GRAMP specifies a conductance range over which the DC operating point analysis sweeps GMINDC. HSPICE replaces GMINDC values over this range, simulates each value, and uses the lowest GMINDC value where the circuit converges in a steady state.

If you sweep GMINDC between 1e-12 mhos (default) and 1e-6 mhos, GRAMP is 6 (value of the exponent difference between the default and the maximum conductance limit). In this example:

- HSPICE first sets GMINDC to 1e-6 mhos and simulates the circuit.
- If circuit simulation converges, HSPICE sets GMINDC to 1e-7 mhos and simulates the circuit.
- The sweep continues until HSPICE simulates all values of the GRAMP ramp.

If the combined GMINDC and GRAMP conductance is greater than 1e-3 mho, false convergence can occur.

Min value: 0; Max value: 1000.

See Also

.DC
.OPTION GMINDC

.OPTION GSCAL

Sets the conductance scale for Pole/Zero analysis.

Syntax

```
.OPTION GSCAL=x
```

Default 1e+3

Description

Use this option to set the conductance scale for Pole/Zero analysis. HSPICE multiplies the conductance and divides the resistance by `GSCAL`.

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION LSCAL
.OPTION PZABS
.OPTION PZTOL
.OPTION RITOL
.PZ

.OPTION GSHDC

Adds conductance from each node to ground when calculating the DC operating point of the circuit.

Syntax

```
.OPTION GSHDC=x
```

Default

`0`

Description

Use this option to add conductance from each node to ground when calculating the DC operating point of the circuit (`.OP`).

See Also

.OPTION GSHUNT

.OPTION GSHUNT

Adds conductance from each node to ground.

Syntax

```
.OPTION GSHUNT=x
```

Default

0

Description

Use this option to add conductance from each node to ground. Add a small GSHUNT to each node to help solve “timestep too small” problems caused by either high-frequency oscillations or numerical noise.

See Also

.OPTION CSHUNT

.OPTION HBACKRYLOVDIM

Specifies the dimension of the Krylov subspace used by the Krylov solver.

Syntax

`.OPTION HBACKRYLOVDIM=`*`value`*

Default

`300`

Description

Use this option to specify the dimension of the Krylov subspace that the Krylov solver uses.

The *value* parameter must specify an integer greater than zero. The range is 1 to infinity.

This option overrides the corresponding PAC option if specified in the netlist.

When this option is not specified in the netlist if `HBACKRYLOVDIM` < `HBKRYLOVDIM`, then `HBACKRYLOVDIM` = `HBKRYLOVDIM`.

See Also

.HB
.OPTION HBKRYLOVDIM

.OPTION HBACKRYLOVITR

Specifies the number of GMRES solver iterations performed by the HB engine.

Syntax

```
.OPTION HBACKRYLOVITR=value
```

Default `1000`

Description

Use this option to specify the number of Generalized Minimum Residual (GMRES) solver iterations that the HB engine performs.

The *value* parameter must specify an integer greater than zero. The range is 1 to infinity.

This option overrides the corresponding PAC option if specified in the netlist.

See Also

.HB
.OPTION HBKRYLOVDIM

.OPTION HBACTOL

Specifies the absolute error tolerance for determining convergence.

Syntax

```
.OPTION HBACTOL=value
```

Default `1.e-8`

Description

Use this option to specify the absolute error tolerance for determining convergence. The *value* parameter must specify a real number greater than zero. The range is 1.e-14 to infinity.

This option overrides the corresponding PAC option if specified in the netlist.

When this option is not specified in the netlist if `HBACTOL` > `HBTOL`, then `HBACTOL` = `HBTOL`.

See Also

.HB
.OPTION HBTOL

.OPTION HBCONTINUE

Specifies whether to use the sweep solution from the previous simulation as the initial guess for the present simulation.

Syntax

```
.OPTION HBCONTINUE= 0|1
```

Default

`1`

Description

Use this option to specify whether to use the sweep solution from the previous simulation as the initial guess for the present simulation.

- `HBCONTINUE=1` Use solution from previous simulation as the initial guess.
- `HBCONTINUE=0`: Start each simulation in a sweep from the DC solution.

See Also

.HB

.OPTION HBFREQABSTOL

Specifies the maximum absolute change in frequency between solver iterations for convergence.

Syntax

```
.OPTION HBFREQABSTOL=value
```

Default 1Hz

Description

Use this option to specify the maximum absolute change in frequency between solver iterations for convergence.

This option is an additional convergence criterion for oscillator analysis.

See Also

.HBOSC

.OPTION HBFREQRELTOL

Specifies the maximum relative change in frequency between solver iterations for convergence.

Syntax

```
.OPTION HBFREQRELTOL=value
```

Syntax `1.e-9`

Description

Use this option to specify the maximum relative change in frequency between solver iterations for convergence.

This option is an additional convergence criterion for oscillator analysis.

See Also

.HBOSC

.OPTION HB_GIBBS

Option for HBTRAN output to minimize Gibbs' phenonema.

Syntax

```
.OPTION HB_GIBBS=<n>
```

Default 0

Description

Minimize any Gibbs' phenomenon that may occur in transforming a square-wave signal from the frequency domain to the time domain. < n >=0 (defaults to zero, which is equivalent to not using it at all). The result is that the HBTRAN waveforms are filtered by a $(\sin c(x))^N$ function before being transformed to the time domain via FFT. This option applies only to single-tone output.

Example

```
.option hb_gibbs = 2
...
.print hbtran v(2)
```

See Also

The *HSPICE User Guide: RF Analysis*, Minimizing Gibbs Phenomenon

.OPTION HBJREUSE

Controls when to recalculate the Jacobson matrix.

Syntax

```
.OPTION HBJREUSE=0|1
```

Default Conditional, see below

Description

Use this option to control when to recalculate the Jacobson matrix.

- `HBJREUSE=0`: Recalculates the Jacobian matrix at each iteration. This is the default if `HBSOLVER=1`.
- `HBJREUSE=1`: Reuses the Jacobian matrix for several iterations if the error is sufficiently reduced. This is the default if `HBSOLVER=0`.

See Also

.HB
.OPTION HBSOLVER

.OPTION HBJREUSETOL

Determines when to recalculate Jacobian matrix if `HBJREUSE=1.0`.

Syntax

```
.OPTION HBJREUSETOL=value
```

Default `0.05`

Description

Determines when to recalculate Jacobian matrix (if `HBJREUSE=1.0`).

This is the percentage by which HSPICE RF must reduce the error from the last iteration so you can use the Jacobian matrix for the next iteration. The *value* parameter must specify a real number between 0 and 1.

See Also

.HB
.OPTION HBJREUSE

.OPTION HBKRYLOVDIM

Specifies the dimension of the subspace used by the Krylov solver.

Syntax

```
.OPTION HBKRYLOVDIM=value
```

Default 40

Description

Use this option to specify the dimension of the Krylov subspace that the Krylov solver uses.

The *value* parameter must specify an integer greater than zero.

See Also

.HB

.OPTION HBKRYLOVTOL

Specifies the error tolerance for the Krylov solver.

Syntax

```
.OPTION HBKRYLOVTOL=value
```

Default

`0.01`

Description

Use this option to specify the error tolerance for the Krylov solver.

The *value* parameter must specify a real number greater than zero.

See Also

.HB

.OPTION HBKRYLOVMAXITER

Specifies the maximum number of GMRES solver iterations performed by the HB engine.

Syntax

```
.OPTION HBKRYLOVMAXITER=value
```

Default

`500`

Description

Use this option to specify the maximum number of Generalized Minimum Residual (GMRES) solver iterations that the HB engine performs.

Analysis stops when the number of iterations reaches this value.

See Also

.HB

.OPTION HBLINESEARCHFAC

Specifies the line search factor.

Syntax

```
.OPTION HBLINESEARCHFAC=value
```

Default `0.35`

Description

Use this option to specify the line search factor.

If Newton iteration produces a new vector of HB unknowns with a higher error than the last iteration, then scale the update step by this value and try again. The *value* parameter must specify a real number between 0 and 1.

See Also

.HB

.OPTION HBMAXITER

Specifies the maximum number of Newton-Raphson iterations performed by the HB engine.

Syntax

```
.OPTION HBMAXITER=value
```

Default `10000`

Description

Use this option to specify the maximum number of Newton-Raphson iterations that the HB engine performs.

Analysis stops when the number of iterations reaches this value.

See Also

.HB

.OPTION HBMAXOSCITER

Specifies the maximum number of outer-loop iterations for oscillator analysis.

Syntax

```
.OPTION HBMAXOSCITER=value
```

Default

`10000`

Description

Use this option to specify the maximum number of outer-loop iterations for oscillator analysis.

See Also

.HBOSC

.OPTION HBPROBETOL

Searches for a probe voltage at which the probe current is less than the specified value.

Syntax

```
.OPTION HBPROBETOL=value
```

Default

`1.e-9`

Description

Use this option to cause oscillator analysis to try to find a probe voltage at which the probe current is less than the specified value.

This option defaults to the value of the `HBTOL` option, which defaults to 1.e-9.

See Also

.HBOSC
.OPTION HBTOL

.OPTION HBSOLVER

Specifies a preconditioner for solving nonlinear circuits.

Syntax

```
.OPTION HBSOLVER=0|1|2
```

Default `1`

Description

Use this option to specify a preconditioner for solving nonlinear circuits.

- `HBSOLVER=0`: Invokes the direct solver.
- `HBSOLVER=1` Invokes the matrix-free Krylov solver.
- `HBSOLVER=2`: Invokes the two-level hybrid time-frequency domain solver.

See Also

.HBOSC
.OPTION HBJREUSE

.OPTION HBTOL

Specifies the absolute error tolerance for determining convergence.

Syntax

```
.OPTION HBTOL=value
```

Default `1.e-9`

Description

Use this option to specify the absolute error tolerance for determining convergence.

The *value* parameter must specify a real number greater than zero.

See Also

.HB

.OPTION HBTRANFREQSEARCH

Specifies the frequency source for the HB analysis of a ring oscillator.

Syntax

```
.OPTION HBTRANFREQSEARCH=[1|0]
```

Default

`1`

Description

Use this option to specify the frequency source for the HB analysis of a ring oscillator.

- `HBTRANFREQSEARCH=1`: HB analysis calculates the oscillation frequency from the transient analysis
- `HBTRANFREQSEARCH=0`: HB analysis assumes that the period is 1/f, where f is the frequency specified in the tones description.

See Also

.HB
.HBOSC
.OPTION HBTOL

.OPTION HBTRANINIT

Selects transient analysis for initializing all state variables for HB analysis of a ring oscillator.

Syntax

```
.OPTION HBTRANINIT=time
```

Description

Use this option to cause HB to use transient analysis to initialize all state variables for HB analysis of a ring oscillator.

The *time* parameter is defined by when the circuit has reached (or is near) steady-state. The default is 0.

See Also

.HB
.HBOSC

.OPTION HBTRANPTS

Specifies the number of points per period for converting time-domain data results into the frequency domain for HB analysis of a ring oscillator.

Syntax

```
.OPTION HBTRANPTS=npts
```

Default `4*nh`

Description

Use this option to specify the number of points per period for converting the time-domain data results from transient analysis into the frequency domain for HB analysis of a ring oscillator.

The *npts* parameter must be set to an integer greater than 0. The units are in nharms (nh).

This option is relevant only if you set `.OPTION HBTRANINIT`. You can specify either `.OPTION HBTRANPTS` or `.OPTION HBTRANSTEP`, but not both.

See Also

.HB
.HBOSC
.OPTION HBTRANINIT
.OPTION HBTRANSTEP

.OPTION HBTRANSTEP

Specifies transient analysis step size for the HB analysis of a ring oscillator.

Syntax

`.OPTION HBTRANSTEP=`*`stepsize`*

Default `1/4*nh*f0`

Description

Use this option to specify transient analysis step size for the HB analysis of a ring oscillator.

The *stepsize* parameter must be set to a real number. The default is 1/(4*nh*f0), where *nh* is the nharms value and *f0* is the oscillation frequency.

This option is relevant only if you set `.OPTION HBTRANINIT`.

Note:

You can specify either `.OPTION HBTRANPTS` or `.OPTION HBTRANSTEP`, but not both.

See Also

.HB
.HBOSC
.OPTION HBTRANINIT
.OPTION HBTRANPTS

.OPTION HIER_DELIM

Replaces the caret delimiter with a period when used for HSPICE in ADE only.

Syntax

```
.OPTION HIER_DELIM= 0|1
```

Default 0

Description

Use `.OPTION HIER_DELIM` to change the hierarchy delimiter from a caret (^) to a period (.) for HSPICE in ADE. When `.OPTION HIER_DELIM=1`, a caret (^) is changed to a period(.). This option only works for HSPICE in ADE; it must work with `.OPTION PSF` and `.OPTION ARTIST`.

- `0`: Maintains the caret.
- `1`: Replaces the caret with a period.

See Also

- .OPTION ARTIST
- .OPTION PSF

.OPTION HIER_SCALE

Uses S-parameters to scale subcircuits.

Syntax

```
.OPTION HIER_SCALE=x
```

Default 0

Description

Use this option so you can use the S-parameter to scale subcircuits.

- 0 Interprets S as a user-defined parameter.
- 1 Interprets S as a scale parameter.

.OPTION ICSWEEP

Saves the current analysis result of a parameter or temperature sweep as the starting point in the next analysis.

Syntax

```
.OPTION ICSWEEP=0|1
```

Default `1`

Description

Use this option to save the current analysis result of a parameter or temperature sweep as the starting point in the next analysis in the sweep.

- If `ICSWEEP=1`, the next analysis uses the current results.
- If `ICSWEEP=0`, the next analysis does not use the results of the current analysis.

.OPTION IMAX

Specifies the maximum timestep in timestep algorithms for transient analysis.

Syntax

```
.OPTION IMAX=x
```

Default

8

Description

Use this option to specify the maximum timestep in algorithms for transient analysis. IMAX sets the maximum iterations to obtain a convergent solution at a timepoint. If the number of iterations needed is greater than IMAX, the internal timestep (delta) decreases by a factor equal to the FT transient control option. The new timestep calculates a new solution. IMAX also works with the IMIN transient control option. IMAX is the same as ITL4.

See Also

.OPTION FT
.OPTION IMIN
.OPTION ITL4

.OPTION IMIN

Specifies the minimum timestep in timestep algorithms for transient analysis.

Syntax

```
.OPTION IMIN=x
```

Default

3

Description

Use this option to specify the minimum number of iterations required to obtain convergence for transient analysis. If the number of iterations is less than IMIN, the internal timestep (delta) doubles.

Use this option to decrease simulation times in circuits where the nodes are stable most of the time (such as digital circuits). If the number of iterations is greater than IMIN, the timestep stays the same unless the timestep exceeds the IMAX option. IMIN is the same as ITL3.

See Also

.OPTION IMAX
.OPTION ITL3

.OPTION INGOLD

Controls whether HSPICE prints output in exponential form or engineering notation in HSPICE/HSPICE RF.

Syntax

```
.OPTION INGOLD=[0|1|2]
```

Default `0` (engineering notation)

Arguments

Parameter	Description	Defaults
INGOLD=0	Engineering Format	1.234K, 123M
INGOLD=1	G Format (fixed and exponential)	1.234e+03, .123
INGOLD=2	E Format (exponential SPICE)	1.234e+03, .123e-1

Description

Use this option to control whether HSPICE prints output in exponential form (scientific notation) or engineering notation. Engineering notation provides two to three extra significant digits and aligns columns to facilitate comparison, as shown below:

```
F=1e-15    M=1e-3
P=1e-12    K=1e3
N=1e-9     X=1e6
U=1e-6     G=1e9
```

HSPICE RF prints variable values in engineering notation by default. To use the exponential form, specify `.OPTION INGOLD=1` or `2`. To print variable values in exponential form, specify `.OPTION INGOLD=1` or `2`.

Currently, `INGOLD` does not control the number format in measure files(**.mt#*/**.ms#*/**.ma#*). It only controls the number format in *.lis* file.

Example

```
.OPTION INGOLD=2
```

See Also

.OPTION NCWARN

.OPTION INTERP

Limits output to only the .TRAN timestep intervals for post-analysis tools.

Syntax

```
.OPTION INTERP=0|1
```

Default 0

Description

Use to limit output for post-analysis tools to only the .TRAN timestep intervals for some post-analysis tools. This option can be used to reduce the size of the post-processing output. By default, HSPICE outputs data at internal timepoints. In some cases, INTERP produces a much larger design *.tr#* file, especially for smaller timesteps, and it also leads to longer runtime.

Note:

Since HSPICE uses the post-processing output to compute the .MEASURE command results, interpolation errors result if you use the INTERP option and your netlist also contains .MEASURE commands. Using the INTERP option with .MEASURE commands is not recommended.

When you run data-driven transient analysis (.TRAN DATA) in an optimization routine, HSPICE forces INTERP=1. All measurement results are at the time points specified in the data-driven sweep. To measure only at converged internal timesteps (for example, to calculate the AVG or RMS), set ITRPRT=1.

See Also

.OPTION ITRPRT
.TRAN

.OPTION IPROP

Controls whether to treat all of the circuit information as IP protected.

Syntax

```
.OPTION IPROP 0|1
```

Default 0

Description

Use to control whether to treat all of the circuit information as IP protected and not output this information during simulation.

- 0= Output information (IP not protected)
- 1=Do not output information (IP protected)

.OPTION ITL1

Specifies the maximum DC iteration limit.

Syntax

```
.OPTION ITL1=n
```

Default `200`

Description

Use this option to specify the maximum DC iteration limit. Increasing this value rarely improves convergence in small circuits. Values as high as 400 have resulted in convergence for some large circuits with feedback (such as operational amplifiers and sense amplifiers). However, most models do not require more than 100 iterations to converge. Set `.OPTION ACCT` to list how many iterations an operating point requires.

See Also

.DC
.OPTION ACCT

.OPTION ITL2

Specifies the iteration limit for the DC transfer curve.

Syntax

```
.OPTION ITL2=n
```

Default

50

Description

Use this option to specify the iteration limit for the DC transfer curve. Increasing this limit improves convergence only for very large circuits.

See Also

.DC

.OPTION ITL3

Specifies minimum timestep in timestep algorithms for transient analysis.

Syntax

```
.OPTION ITL3=x
```

Default

3

Description

Use this option to specify the minimum timestep in timestep algorithms for transient analysis. ITL3 is the minimum number of iterations required to obtain convergence. If the number of iterations is less than ITL3, the internal timestep (delta) doubles.

Use this option to decrease simulation times in circuits where the nodes are stable most of the time (such as digital circuits). If the number of iterations is greater than IMIN, the timestep stays the same unless the timestep exceeds the IMAX option. ITL3 is the same as IMIN.

See Also

.OPTION IMAX
.OPTION IMIN

.OPTION ITL4

Specifies maximum timestep in timestep algorithms for transient analysis in HSPICE/HSPICE RF.

Syntax

```
.OPTION ITL4=x
```

Default

`8`

Description

Use this option to specify the maximum timestep in timestep algorithms for transient analysis. `ITL4` sets the maximum iterations to obtain a convergent solution at a timepoint. If the number of iterations needed is greater than `ITL4`, the internal timestep (delta) decreases by a factor equal to the `FT` transient control option. HSPICE uses the new timestep to calculate a new solution. `ITL4` also works with the `IMIN` transient control option. For HSPICE, `ITL4` is the same as `IMAX`.

See Also

.OPTION FT
.OPTION IMAX
.OPTION IMIN

.OPTION ITL5

Sets an iteration limit for transient analysis.

Syntax

```
.OPTION ITL5=x
```

Default `0`(infinite number of iterations)

Description

Use this option to set an iteration limit for a transient analysis. If a circuit uses more than `ITL5` iterations, the program prints all results up to that point.

.OPTION ITLPTRAN

Controls iteration limit used in the final try of the pseudo-transient method.

Syntax

```
.OPTION ITLPTRAN=x
```

Default

30

Description

Use this option to control the iteration limit used in the final try of the pseudo-transient method in OP or DC analysis. If a simulation fails in the final try of the pseudo-transient method, provide a higher value.

See Also

.DC
.OP

.OPTION ITLPZ

Sets the iteration limit for pole/zero analysis.

Syntax

```
.OPTION ITLPZ=x
```

Default `100`

Description

Use this option to set the iteration limit for pole/zero analysis.

See Also

.OPTION CSCAL
.OPTION GSCAL
.PZ
.OPTION FMAX

.OPTION ITRPRT

Enables printing of output variables at their internal time points.

Syntax

```
.OPTION ITRPRT 0|1
```

Default `0`

Description

Use this option to enable printing of output variables at their internal time points.

When set to `1`, HSPICE prints output variables at their internal transient simulation time points. In addition, if you use the `-html` option when invoking HSPICE, then HSPICE prints the values to a separate file (**.printtr0*).

.OPTION KCLTEST

Activates the KCL (Kirchhoff's Current Law) test.

Syntax

```
.OPTION KCLTEST=0|1
```

Default `0`

Description

Use this option to activate the KCL test. This increases simulation time, especially for large circuits, but checks the solution with a high degree of accuracy.

If you set this value to 1, HSPICE sets these options:

- Sets `RELMOS` and `ABSMOS` options to `0` (off).
- Sets `ABSI` to `1e-6 A`.
- Sets `RELI` to `1e-6`.

To satisfy the KCL test, each node must satisfy this condition:

$$\left|\Sigma i_b\right| < RELI \cdot \ \Sigma\left|i_b\right| + ABSI$$

In this equation, the i_bs are the node currents.

See Also

.OPTION ABSI
.OPTION ABSMOS
.OPTION RELI
.OPTION RELMOS

.OPTION KLIM

Sets the minimum mutual inductance.

Syntax

```
.OPTION KLIM=x
```

Default `0.01`

Description

Use this option to set the minimum mutual inductance below which automatic second-order mutual inductance calculation no longer proceeds. `KLIM` is unitless (analogous to coupling strength, specified in the K-element). Typical `KLIM` values are between .5 and 0.0.

.OPTION LA_FREQ

Specifies the upper frequency for which accuracy must be preserved.

Syntax

`.OPTION LA_FREQ=`*`value`*

Default 1GHz

Description

Use this option to specify the upper frequency for which accuracy must be preserved.

The *value* parameter specifies the upper frequency for which the `PACT` algorithm must preserve accuracy. If *value* is 0, the algorithm drops all capacitors because only DC is of interest.

The maximum frequency required for accurate reduction depends on both the technology of the circuit and the time scale of interest. In general, the faster the circuit, the higher the maximum frequency.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.OPTION SIM_LA
.OPTION LA_TIME

.OPTION LA_MAXR

Specifies the maximum resistance for linear matrix reduction.

Syntax

```
.OPTION LA_MAXR=value
```

Default `1e15` ohms

Description

Use this option to specify the maximum resistance for linear matrix reduction.

The *value* parameter specifies the maximum resistance preserved in the reduction. The linear matrix reduction process assumes that any resistor greater than *value* has an infinite resistance and drops the resistor after reduction is completed.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.OPTION SIM_LA

.OPTION LA_MINC

Specifies the minimum capacitance for linear matrix reduction.

Syntax

```
.OPTION LA_MINC=<value>
```

Dedault 1e-16 farads.

Description

Use this option to specify the minimum capacitance for linear matrix reduction.

The *value* parameter specifies the minimum capacitance preserved in the reduction.

The linear matrix reduction process lumps any capacitor smaller than *value* to ground after the reduction completes.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.OPTION SIM_LA

.OPTION LA_TIME

Specifies the minimum time for which accuracy must be preserved.

Syntax

```
.OPTION LA_TIME=value
```

Default 1ns

Description

Use this option to specify the minimum time for which accuracy must be preserved.

The *value* parameter specifies the minimum switching time for which the PACT algorithm preserves accuracy.

Waveforms that occur more rapidly than the minimum switching time are not accurately represented.

This option is simply an alternative to `.OPTION LA_FREQ`. The default is equivalent to setting `LA_FREQ=1GHz`.

Note:

Higher frequencies (smaller times) increase accuracy, but only up to the minimum time step used in HSPICE.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: Simulation and Analysis*.

Example

For a circuit having a typical rise time of 1ns, either set the maximum frequency to 1 GHz, or set the minimum switching time to 1ns:

```
.OPTION LA_FREQ=1GHz
-or-
.OPTION LA_TIME=1ns
```

However, if spikes occur in 0.1ns, HSPICE does not accurately simulate them. To capture the behavior of the spikes, use:

```
.OPTION LA_FREQ=10GHz
-or-
.OPTION LA_TIME=0.1ns
```

See Also

.OPTION SIM_LA
.OPTION LA_FREQ

.OPTION LA_TOL

Specifies the error tolerance for the PACT algorithm.

Syntax

```
.OPTION LA_TOL=value
```

Default 0.05.

Description

Use this option to specify the error tolerance for the PACT algorithm.

The *value* parameter must specify a real number between 0.0 and 1.0.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.OPTION SIM_LA

.OPTION LENNAM

Specifies maximum name length for printing operating point analysis results.

Syntax

```
.OPTION LENNAM=x
```

Default `16` (characters)

Description

Use this option to specify the maximum length of names in the printout of operating point analysis results. The maximum value is 1024. `.OPTION LENNAME` prints the full related name of the transistor in the noise tables and OP tables.

Example

```
...
.OPTIONS POST=1 LENNAM=40
...
```

.OPTION LIMPTS

Specifies the number of points to print in AC analysis.

Syntax

```
.OPTION LIMPTS=x
```

Default

`2001`

Description

Use this option to specify the number of points to print or plot in AC analysis. You do not need to set `LIMPTS` for a DC or transient analysis. HSPICE spools the output file to disk.

See Also

.AC
.DC
.TRAN

.OPTION LIMTIM

Specifies the amount of CPU time reserved to generate prints.

Syntax

```
.OPTION LIMTIM=x
```

Default `2` (seconds)

Description

Use this option to specify the amount of CPU time reserved to generate prints and plots if a CPU time limit (`CPTIME=x`) terminates simulation. Default is normally sufficient for short printouts.

See Also

.OPTION CPTIME

.OPTION LISLVL

Controls whether of not HSPICE suppresses the circuit number to circuit name directory information in the list file.

Syntax

```
LISLVL=0|1
```

Default `0`

Description

`LISLVL=0` prints the circuit name directory information in the *.lis* file.

If the value is `1`, the circuit number and circuit name directory information is not output to the *.lis* file.

.OPTION LIST

Prints a list of netlist elements, node connections, and values for components, voltage and current sources, parameters, and more.

Syntax

```
.OPTION LIST
```

Default

0

Description

Use this option to print a list of:

- Netlist elements
- Node connections
- Element values for passive and active components
- Independent and dependent voltage and current source values
- Parameter values

It also prints effective sizes of elements and key values.

Note:

This option is suppressed by the BRIEF option.

See Also

.OPTION BRIEF
.OPTION UNWRAP
.OPTION VFLOOR

.OPTION LOADHB

Loads state variable information from a specified file.

Syntax

```
.OPTION LOADHB='filename'
```

Description

Use this option to load the state variable information contained in the specified file. These values are used to initialize the HB simulation.

See Also

.HB
.OPTION SAVEHB

.OPTION LOADSNINIT

Loads the operating point saved at the end of Shooting Newton analysis initialization.

Syntax

```
.OPTION LOADSNINIT="filename"
```

Description

Use this option to load the operating point file saved at the end of SN initialization, which is used as initial conditions for the Shooting-Newton method.

.OPTION LSCAL

Sets the inductance scale for Pole/Zero analysis.

Syntax

```
.OPTION LSCAL=x
```

Default `1e+6`

Description

Use this option to set the inductance scale for Pole/Zero analysis. HSPICE multiplies inductance by `LSCAL`.

Note:

Scale factors must satisfy the following relations:

$$GSCAL = CSCAL \cdot FSCAL$$

$$GSCAL = \frac{1}{LSCAL \cdot FSCAL}$$

If you change scale factors, you might need to modify the initial Muller points, (X0R, X0I), (X1R, X1I) and (X2R, X2I), even though HSPICE internally multiplies the initial values by (1.0e-9/GSCAL).

The three complex starting-trial points, in the Muller (x1R,X1I) algorithm for pole/zero analysis are listed below with their defaults. HSPICE multiplies these initial points, and FMAX, by FSCAL.

Starting-Trial Points	Defaults	
.OPTION (X0R,X0I)	X0R=-1.23456e6	X0I=0.0
.OPTION (X1R,X1I)	X1R=1.23456e5	X1I=0.0
.OPTION (X2R,X21)	X2R=+1.23456e6	X2I=0.0

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION ITLPZ
.OPTION PZABS
.OPTION PZTOL

.OPTION RITOL
.OPTION (X0R,X0I)
.OPTION (X1R,X1I)
.OPTION (X2R,X21)
.PZ

.OPTION LVLTIM

Selects the timestep algorithm for transient analysis.

Syntax

```
.OPTION LVLTIM=[1|2|3]|4
```

Default 1

Description

Use this option, (levels 1-3, only) to select the timestep algorithm for transient analysis.

- `LVLTIM=1` (default) uses the `DVDT` timestep control algorithm.
- `LVLTIM=2` uses the local truncation error (LTE) timestep control method. You can apply `LVLTIM=2` to the TRAP method.
- `LVLTIM=3` uses the `DVDT` timestep control method with timestep reversal.
- `LVLTIM=4` is invalid if set by user; it is invoked by the RUNLVL option only to enhance the LTE time step control method used by the latest RUNLVL algorithm.

The local truncation algorithm `LVLTIM=2` (LTE) provides a higher degree of accuracy than `LVLTIM=1` or `3` (`DVDT`). If you use this option, errors do not propagate from time point to time point, which can result in an unstable solution.

Selecting the `GEAR` method changes the value of `LVLTIM` to `2` automatically. For information on how LVLTIM values impact other options, see Appendix B, How Options Affect other Options.

See Also

.OPTION CHGTOL
.OPTION DVDT
.OPTION FS
.OPTION FT
.OPTION RELQ

.OPTION MACMOD

Enables HSPICE MOSFET to access the subcircuit definition when there is no matching model reference or enables an HSPICE X-element to access the model reference when there is no matching subcircuit definition.

Syntax

```
.OPTION MACMOD=[1|2|3|0]
```

Default 0

Description

When macmod=1, HSPICE seeks a subckt definition for the M*** element if no model reference exists. The desired subckt name must match (case insensitive) the mname field in the M*** instance command. In addition, the number of terminals of the subckt must match with the M*** element referencing it; otherwise HSPICE exits the simulation based on no definition for the M*** element.

The following limitations apply when macmod=1:

1. Element template output does not support MOSFET elements which use subckt definitions.
2. This feature will not support a MOSFET element whose mname is defined by a string parameter.
3. The number of terminals for a HSPICE MOSFET element must be within the range of 3-7; any number of terminals that is out of this range causes the simulation to fail.

When macmod=2, HSPICE seeks a MOSFET model definition when it cannot find a matching subckt or Verilog-A definition for an X-element. The targeted MOSFET MODEL card could be either an HSPICE built-in MOSFET model or CMI MOSFET model. If the model card that matched the X-element reference name is not a type of MOSFET model, the simulator exits and displays an error message indicating that the reference is not found.

The following limitations apply when macmod=2:

1. The feature of “string parameter supported in MOSFET model name” is not applied to X-elements that are mapped to MOSFET model cards; that is, the reference name of the X-element must be constant string characters.
2. Subckt direct port probing command, isub() is not supported on X-elements mapped to MOSFET model cards.

3. HSPICE MOSRA analysis might not be performed on the X-elements, even when they directly map to MOSFET model cards.

When `macmod=3`, HSPICE enables the same features as when `macmod=1`. HSPICE seeks a `.subckt` definition for an M-element if there is no matching model reference; HSPICE seeks a `.model` MOSFET definition for an X-element if there is no matching `.subckt` or Verilog-A definition. Usage considerations and limitations remain the same for both features, respectively.

Note:

When `MACMOD=2` or `3`, for the X-element that maps to an M-element, if it has an instance parameter named '`Multi`' (case insensitive), then '`Multi`' is used as an alias for the '`M`' factor (the `M` multiply parameter).

When `macmod=0`: if there is no `.option MACMOD` in the input files or `MACMOD=0`, then neither of the features is enabled. HSPICE ignores the option `MACMOD` when any value other than `1 | 2 | 3 | 0` is set.

The `MACMOD` option is a global option; if there are multiple `MACMOD` options in one simulation, HSPICE uses the value of the last `MACMOD` option.

For examples and detailed discussion, see MOSFET Element Support Using .OPTION MACMOD in the *HSPICE User Guide: Simulation and Analysis*.

.OPTION MAXAMP

Sets the maximum current through voltage-defined branches.

Syntax

```
.OPTION MAXAMP=x
```

Default 0

Description

Use this option to set the maximum current through voltage-defined branches (voltage sources and inductors). If the current exceeds the MAXAMP value, HSPICE reports an error.

.OPTION MAXORD

Specifies the maximum order of integration for the GEAR method.

Syntax

```
.OPTION MAXORD=[1|2|3]
```

Default

2

Description

Use this option to specify the maximum order of integration for the GEAR method. When the GEAR method is used, based on the circuit behavior, HSPICE/HSPICE RF automatically switches the GEAR order on the fly.

The value of the parameter can be either 1, 2, or 3:

- MAXORD=1 selects the first-order GEAR (Backward-Euler) integration.
- MAXORD=2 selects the second-order GEAR (Gear-2), which is more stable and accurate than MAXORD=1.
- MAXORD=3 selects the third-order or high GEAR (Gear-3), which is most accurate, since it uses 3 previous time points to estimate the next time point.

Example

This example selects the Backward-Euler integration method.

```
.OPTION MAXORD=1 METHOD=GEAR
```

See Also

.OPTION METHOD
.OPTION RUNLVL (HSPICE)

.OPTION MBYPASS

Computes the default value of the BYTOL control option.

Syntax

```
.OPTION MBYPASS=x
```

Default 2.00

Description

Use this option to calculate the default value of the BYTOL control option:

```
BYTOL=MBYPASS x VNTOL=0.100m
```

Also multiplies the RELV voltage tolerance. Set MBYPASS to about 0.1 for precision analog circuits.

- Default is 1 for DVDT=0, 1, 2, or 3.
- Default is 2 for DVDT=4.

See Also

.OPTION BYTOL
.OPTION DVDT
.OPTION RELV

.OPTION MCBRIEF

Controls how HSPICE outputs Monte Carlo parameters.

Syntax

```
.OPTION MCBRIEF=0|1|2|3
```

Default `0`

Description

Use this option to control how HSPICE outputs Monte Carlo parameters:

- `MCBRIEF=0`: Outputs all Monte Carlo parameters
- `MCBRIEF=1`: Suppresses the MC results in the **.mt#* file.
- `MCBRIEF=2`: Outputs the Monte Carlo parameters into a *.lis* file only.
- `MCBRIEF=3`: Outputs the Monte Carlo parameters into the measure files only.

.OPTION MEASDGT

Formats the .MEASURE command output in both the listing file and the .MEASURE output files.

Syntax

```
.OPTION MEASDGT=x
```

Default

4.0

Description

Use this option to format the .MEASURE command output in both the listing file and the .MEASURE output files (.ma0, .mt0, .ms0, and so on).

The value of x is typically between 1 and 7, although you can set it as high as 10.

Use MEASDGT with .OPTION INGOLD=x to control the output data format.

Example

For example, if MEASDGT=5, then .MEASURE displays numbers as:

- Five decimal digits for numbers in scientific notation.
- Five digits to the right of the decimal for numbers between 0.1 and 999.

In the listing (.lis), file, all .MEASURE output values are in scientific notation so .OPTION MEASDGT=5 results in five decimal digits.

See Also

.OPTION INGOLD
.MEASURE (or) .MEAS

.OPTION MEASFAIL

Specifies where to print the failed measurement output.

Syntax

```
.OPTION MEASFAIL=0|1
```

Default

`1`

Description

Use this option to specify where to print the failed measurement output. You can assign this option the following values:

- `MEASFAIL=0`, outputs “0” into the *.mt#*, *.ms#*, or *.ma#* file, and prints “failed” in the *.lis* file.
- `MEASFAIL=1`, prints “failed” in the *.mt#*, *.ms#*, or *.ma#* file, and in the *.lis* file.

See Also

.MEASURE (or) .MEAS

.OPTION MEASFILE

Controls whether measure information outputs to single or multiple files when an `.ALTER` command is present in the netlist.

Syntax

```
.OPTION MEASFILE=0|1
```

Default `0`

Description

Use this option to control whether the measure information outputs to a single or multiple files when an `.ALTER` command is present in the netlist. You can assign this option the following values:

- `MEASFILE=0`, outputs measure information to several files.
- `MEASFILE=1` , outputs measure information to a single file.

See Also

.ALTER
.MEASURE (or) .MEAS

.OPTION MEASOUT

Outputs `.MEASURE` command values and sweep parameters into an ASCII file.

Syntax

```
.OPTION MEASOUT=x
```

Default `0|1`

Description

Use this option to output `.MEASURE` command values and sweep parameters into an ASCII file. Post-analysis processing (AvanWaves or other analysis tools) uses this <*design*>.mt# file, where # increments for each `.TEMP` or `.ALTER` block.

For example, for a parameter sweep of an output load, which measures the delay, the .mt# file contains data for a delay-versus-fanout plot. You can set this option to `0` (off) in the *hspice.ini* file.

See Also

.ALTER
.MEASURE (or) .MEAS
.TEMP (or) .TEMPERATURE

.OPTION METHOD

Sets the numerical integration method for a transient analysis for HSPICE/ HSPICE RF.

Syntax

```
.OPTION METHOD=GEAR | TRAP [PURETP] | BDF
```

Default TRAP

Description

Use this option to set the numerical integration method for a transient analysis.

- TRAP selects trapezoidal rule integration. This method inserts occasional Backward-Euler timesteps to avoid numerical oscillations. You can use the PURETP option to turn this oscillation damping feature off.
- TRAP PURETP selects pure trapezoidal rule integration. This method is recommended for high-Q LC oscillators and crystal oscillators.
- GEAR selects Gear integration, which sets .OPTION LVLTIM=2.
- GEAR MU=0 selects Backward-Euler integration.
- BDF selects the high order integration method based on the backward differentiation formulation.

  Note:

  To change LVLTIM from 2 to 1 or 3, set LVLTIM=1 or 3 after the METHOD=GEAR option. This overrides METHOD=GEAR, which sets LVLTIM=2.

TRAP (trapezoidal) integration usually reduces program execution time with more accurate results. However, this method can introduce an apparent oscillation on printed or plotted nodes, which might not result from circuit behavior. To test this, run a transient analysis by using a small timestep. If oscillation disappears, the cause is the trapezoidal method.

The GEAR method is a filter, removing oscillations that occur in the trapezoidal method. Highly non-linear circuits (such as operational amplifiers) can require very long execution times when you use the GEAR method. Circuits that do not converge in trapezoidal integration, often converge if you use GEAR.

The BDF method is an high order integration method based on the backward differentiation formulae. The key features include: variable order, variable step size, and high order polynomial interpolation. The BDF limitations are listed below.

`METHOD=BDF` supports the following models *only*:

- MOSFET, levels 1-54
- BJT, level 1 only
- Diodes, all
- Resistors, all
- Capacitors, excludes DC block
- Independent sources: V and I
- Dependent sources: E/F/G/H
- L, excludes AC choke
- K, excludes magnetic core, ideal transformer

When RUNLVL is turned off (`=0`), `method=GEAR` sets `bypass=0`; the user can reset bypass value by using `.option bypass=`*`value`*. Also, when RUNLVL is turned off, there is an order dependency with GEAR and ACCURATE options; if `method=GEAR` is set after the ACCURATE option, then the ACCURATE option does not take effect; if `method=GEAR` is set before the ACCURATE option, then both GEAR and ACCURATE take effect.

If GEAR is used with RUNLVL, then GEAR only determines the numeric integration method; anything else is controlled by RUNLVL; there is no order dependency with RUNLVL and GEAR. Since there is no order dependency with RUNLVL and GEAR, or RUNLVL and ACCURATE, then:

```
.option ACCURATE method=GEAR RUNLVL
```

is equivalent to

```
.option method=GEAR ACCURATE RUNLVL
```

To see how use of the GEAR method impacts the value settings of ACCURATE and other options, see Appendix B, How Options Affect other Options.

Example 1

This example sets pure trapezoidal method integration. No Gear-2 or Backward-Euler is mixed in. Use this setting when you simulate harmonic oscillators.

```
.option method=trap puretp
```

Example 2

This example sets pure Backward-Euler integration.

```
.option method=gear maxord=1
```

Example 3

This example sets pure Gear-2 integration.

```
.option method=gear
```

Example 4

This example sets the higher order backward differentiation formulation integration for supported models.

```
.option method=bdf
```

See Also

.OPTION ACCURATE
.OPTION LVLTIM
.OPTION MAXORD
.OPTION MTTHRESH
.OPTION PURETP
.OPTION MU
.OPTION RUNLVL

.OPTION MODMONTE

Controls how random values are assigned to parameters with Monte Carlo definitions.

Syntax

```
.OPTION MODMONTE=0|1
```

Default 0

Description

Ordinarily, the assignment of a random value is only done once, then used several times. The exception to this rule is for model parameters. Since a model definition is only done once, the behavior described above would assign the same parameter value to all devices referencing that model. To overcome this, `.OPTION MODMONTE` lets you decide if all instances of a device should get the same or unique model parameters. Use this option to control how random values are assigned to parameters with Monte Carlo definitions.

- If `MODMONTE=1`, then within a single simulation run, each device that shares the same model card and is in the same Monte Carlo index receives a different random value for parameters that have a Monte Carlo definition.
- If `MODMONTE=0`, then within a single simulation run, each device that shares the same model card and is in the same Monte Carlo index receives the same random value for its parameters that have a Monte Carlo definition.

Example 1

In the following example, transistors M1 through M3 have the same random `vto` model parameter for each of the five Monte Carlo runs through the use of the `MODMONTE` option.

```
...
.option MODMONTE=0 $$ MODMONTE defaults to 0;OK to omit this line.
.param vto_par=agauss(0.4, 0.1, 3)
.model mname nmos level=53 vto=vto_par version=3.22
M1 11 21 31 41 mname W=20u L=0.3u
M2 12 22 32 42 mname W=20u L=0.3u
M3 13 23 33 43 mname W=20u L=0.3u
...
.dc v1 0 vdd 0.1 sweep monte=5
.end
```

Example 2

In Example 2, transistors M1 through M3 have different values of the `vto` model parameter for each of the Monte Carlo runs by the means of setting `.option MODMONTE=1`.

```
...
.option MODMONTE=1
.param vto_par=agauss(0.4, 0.1, 3)
.model mname nmos level=54 vto=vto_par
M1 11 21 31 41 mname W=20u L=0.3u
M2 12 22 32 42 mname W=20u L=0.3u
M3 13 23 33 43 mname W=20u L=0.3u
...
.dc v1 0 vdd 0.1 sweep monte=5
.end
```

See Also

.MODEL

.OPTION MONTECON

Continues a Monte Carlo analysis in HSPICE by retrieving the next random value, even if non-convergence occurs.

Syntax

```
.OPTION MONTECON=0|1
```

Default `1`

Description

Use this option to retrieve the next random value, even if non-convergence occurs. A random value can be too large or too small to cause convergence to fail. Other types of analyses can use this Monte Carlo random value.

.OPTION MOSRALIFE

Does the MOSRA “lifetime” computation.

Syntax

`.OPTION MOSRALIFE=`*`degradation_type_keyword`*

Description

Use this option to compute the lifetime calculation for the degradation type specified. If the option is not specified or the keyword cannot be identified by the MRAlifetimeDeg function, HSPICE does not do the lifetime computation.

The option is used in conjunction with two others, `.OPTION DegFN=`*`val`* and `.OPTION DegFP=`*`val`* which is NMOS's or PMOS’s degradation value at lifetime, respectively.

The options apply to all MOSFETs. The lifetime value is printed in the RADEG file.

See Also

.MOSRA
MOSFET Model Reliability Analysis (MOSRA)

.OPTION MOSRASORT

Enables the descending sort for reliability degradation (RADEG) output.

Syntax

`.OPTION MOSRASORT=`*`degradation_type_keyword`*

Default `delvth0`

Description

Use this option `mosrasort` to enable the descending sort for reliability degradation (RADEG) output.

If the `mosrasort` option is not specified, or the degradation type keyword is not recognized, HSPICE does not do the sorting. (Degradation type keywords are listed in the *HSPICE Application Note: Unified Custom Reliability Modeling API (MOSRA API)*, available by contacting the HSPICE technical support team.)

If you only specify the option `mosrasort`, and do not specify the degradation type keyword, HSPICE sorts RADEG by the `delvth0` keyword.

HSPICE sorts the output separately in lists, one for NMOS, one for PMOS. HSPICE prints the NMOS list first, and then the PMOS list.

Example

In the following usage, the option does a descending sort for RADEG output on `delvth0`'s value.

```
.option mosrasort=delvth0
```

See Also

.MOSRA
MOSFET Model Reliability Analysis (MOSRA)

.OPTION MRAAPI

Loads and links the dynamically linked MOSRA API library.

Syntax

```
.OPTION MRAAPI=0|1
```

Default 0

Description

Use this option to load and link the compiled MOSRA API object *.so* file to HSPICE during simulation runs. If this option parameter is set with no value or to 1, then the MOSRA API *.so* file is loaded as a dynamically-linked object file.

If this option parameter does not exist in the netlist, or is explicitly set to 0, no loading or linking takes place.

.OPTION MRAPAGED

Enables the MOSRA API to work in Paged mode.

Syntax

```
.OPTION MRAPAGED=0|1
```

Default 0

Description

Use this option to select the delta_P or Paged mode. If this option parameter is set to 1, then the MOSRAAPI works in Paged mode. If this option parameter does not exist (deemed as default) in the netlist, or is explicitly set to 0, MOSRAAPI works in delta_P mode.

- 0: delta_P mode
- 1: Paged mode

.OPTION MTTHRESH

Reduces the default active device limit for multithreading.

Syntax

```
.OPTION MTTHRESH=N
```

Default 512

Description

Use the option to reduce the default active device limit to allow multithreading for faster simulation. If $N < 2$ or the option is not set, the `mtthresh` defaults to 512. If the count of VCCSs(G), VCVSs(E), CCCSs(F), CCVSs(H), MOSFETs, BJTs, or diodes is more than `mtthresh`, the circuit can be run in multithreading mode.

.OPTION MU

Defines the integration method coefficient.

Syntax

```
.OPTION MU=x
```

Default `0.5`

Description

Use this option to define the integration method coefficient. The value range is 0.0 to 0.5. The default integration method is trapezoidal which corresponds to the default coefficient value of 0.5. If the value is set to 0, then the integration method becomes backward-Euler. A value between 0 and 0.5 is a blend of the trapezoidal and backward-Euler integration methods.

See Also

.OPTION METHOD

.OPTION NCFILTER

Filters negative conductance warning messages according to the setting value.

Syntax

`.OPTION NCFILTER=`*`val`*

Default `-1e-12`

Description

When `.option ncwarn` is set, use this option to filter the negative conductance warning messages according to the setting value. If gds, gm, gmbs < *value*, a warning message is reported. When `ncwarn` is set, this filter is automatically enabled. The legal range of *val* is –1e20 to 0.

See Also

.OPTION NCWARN

.OPTION NCWARN

Allows turning on a switch to report a warning message for negative conductance on MOSFETs.

Syntax

```
.OPTION NCWARN=0|1
```

Default 0

Description

Use the option to turn on (`.option NCWARN=1`), printing out of the first occurrence of MOSFET related “negative conductance” in the listing file; if you want to check the entire negative conductance on MOSFETs, use `.option DIAGNOSTIC` to print all these warning messages.`NCWARN=0` (default) turns off all warning messages on negative conductance.

See Also

.OPTION DIAGNOSTIC (or) .OPTION DIAGNO
.OPTION NCFILTER

.OPTION NEWTOL

Calculates one or more iterations past convergence for every calculated DC solution and timepoint circuit solution.

Syntax

```
.OPTION NEWTOL=x
```

Default 0

Description

Use this option to calculate one or more iterations past convergence for every calculated DC solution and timepoint circuit solution. If you do not set NEWTOL after HSPICE determines convergence the convergence routine ends and the next program step begins.

.OPTION NODE

Prints a node cross-reference table.

Syntax

```
.OPTION NODE=x
```

Default

0

Description

Use this option to print a node cross-reference table. The BRIEF option suppresses NODE. The table lists each node and all elements connected to it. A code indicates the terminal of each element. A colon (:) separates the code from the element name.

The codes are:

+ — Diode anode
- — Diode cathode
B — BJT base
B — MOSFET or JFET bulk
C — BJT collector
D — MOSFET or JFET drain
E — BJT emitter
G — MOSFET or JFET gate
S — BJT substrate
S — MOSFET or JFET source

Example

```
1 M1:B D2:+ Q4:B
```

This sample part of a cross-reference line indicates that the bulk of M1, the anode of D2 and the base of Q4, all connect to node 1.

See Also

.OPTION BRIEF

.OPTION NOELCK

Bypasses element checking to reduce preprocessing time for very large files.

Syntax

```
.OPTION NOELCK 0|1
```

Default 0

Description

Use this option to bypass element checking to reduce preprocessing time for very large files. HSPICE typically checks for duplicate element definitions. If `.option NOELCK` is set (1), HSPICE skips the element checking and the simulation runs even if there is a duplicate element definition. For the duplicate elements, HSPICE uses the last definition it finds.

When `NOELCHK` is not turned on, if HSPICE finds a duplicate element definition, it issues an error and aborts the simulation.

Example

In the following netlist:

```
R1 1 2 1k
R2 2 0 1k
C1 2 end 1p
C1 2 0 1n
```

...unless `.option NOELCHK` is set to 1, HSPICE aborts the simulation and issue an error message.

```
**error** attempts to redefine c1    at line   xx and line yy
```

.OPTION NOISEMINFREQ

Specifies the minimum frequency of noise analysis in HSPICE/HSPICE RF.

Syntax

```
.OPTION NOISEMINFREQ=x
```

Default `1e-5`

Description

Use this option to specify the minimum frequency of noise analysis. If the frequency of noise analysis is smaller than the minimum frequency, then HSPICE automatically sets the frequency for `NOISEMINFREQ`.

.OPTION NOMOD

Suppresses the printout of model parameters.

Syntax

```
.OPTION NOMOD
```

Default

0

Description

Use this option to suppress the printout of model parameters.

.OPTION NOPAGE

Suppresses page ejects for title headings.

Syntax

```
.OPTION NOPAGE
```

Default

0

Description

Use this option to suppress page ejects for title headings.

.OPTION NOPIV

Controls whether HSPICE automatically switches to pivoting matrix factors.

Syntax

```
.OPTION NOPIV=0|1
```

Default 0

Description

Use this option to prevent HSPICE from automatically switching to pivoting matrix factors if a nodal conductance is less than PIVTOL. NOPIV=1 inhibits pivoting.

See Also

.OPTION PIVTOL

.OPTION NOTOP

Suppresses topology checks to increase preprocessing speed.

Syntax

```
.OPTION NOTOP=0|1
```

Default 0

Description

Use this option to suppress topology checks to increase the speed for preprocessing very large files. HSPICE normally checks the netlist topology and reports a warning or error message. The different topologies that HSPICE checks includes inductor/voltage loops, dangling nodes, stacked current sources and current sources in a closed capacitor loop. If you set the NOTOP option to 1, these checks will not be performed and there will be no warning or error messages issued for these topologies.

Example

If you run the following netlist:

```
R1 1 2 1k
R2 2 0 1k
C1 2 end 1p
```

...the dangling node check function causes HSPICE to issue a warning in the *.lis* file.

```
only 1 connection at node 0:end ...
```

If .option NOTOP is set, the topology check is skipped and you will not get the warning.

.OPTION NOWARN

Suppresses warning messages.

Syntax

```
.OPTION NOWARN=0|1
```

Default 0

Description

Use this option to suppress all conflicting parameter warning messages, except those generated from commands in .ALTER blocks.

.OPTION WARNLIMIT can be used to limit the number of a same warning message.

See Also

.ALTER
.OPTION WARNLIMIT (or) .OPTION WARNLIM

.OPTION NUMDGT

Controls the listing printout accuracy.

Syntax

```
.OPTION NUMDGT=x
```

Default

`4`

Description

Use this option to control the listing printout (.*lis*) accuracy. The value of *x* is typically between 1 and 7, although you can set it as high as 10. This option does not affect the accuracy of the simulation.This option does, however, affect the results files (ASCII and binary) if you use the `.OPTION POST_VERSION=2001` setting. The default setting is 5 digits for results for printout accuracy when using `POST_VERSION=2001`.

See Also

.OPTION POST_VERSION

.OPTION NUMERICAL_DERIVATIVES

Diagnostic-only option for checking a problem with the device models.

Syntax

```
.OPTION NUMERICAL_DERIVATIVES=0|1
```

Default `0`

Description

This option can be used to help diagnose convergence problems or suspected inaccuracies in small-signal analyses such as HBAC, HBNOISE, or PHASENOISE. If a convergence or accuracy problem stems from an inaccuracy in the current or charge derivatives returned by a transistor or diode model, setting this option to 1 will resolve the problem, although with a performance decrease.

If `NUMERICAL_DERIVATIVES=1` resolves the problem, please contact Synopsys support so that the underlying transistor model issue can be resolved.

If you are confident that the models are providing accurate derivatives, do *not* use this option.

.OPTION NXX

Stops echoing (printback) of the data file to stdout.

Syntax

```
.OPTION NXX
```

Default

0

Description

Use this option to terminate echoing (printback) of the data file to stdout until HSPICE finds an .OPTION BRIEF=0 or the .END command. It also resets the LIST, NODE and OPTS options and sets NOMOD. When BRIEF=0, it enables printback. NXX is the same as BRIEF.

See Also

.OPTION BRIEF
.OPTION LIST
.OPTION NODE
.OPTION OPTS

.OPTION OFF

Initializes terminal voltages to zero for active devices not initialized to other values.

Syntax

```
.OPTION OFF=x
```

Default

0

Description

Use this option to initialize terminal voltages to zero if you did not initialize them to other values for all active devices. For example, if you did not initialize both drain and source nodes of a transistor (using .NODESET, .IC commands, or connecting them to sources), then OFF initializes all nodes of the transistor to 0.

HSPICE checks the OFF option before element IC parameters. If you assigned an element IC parameter to a node, simulation initializes the node to the element IC parameter value, even if the OFF option previously set it to 0.

You can use the OFF element parameter to initialize terminal voltages to 0 for specific active devices. Use the OFF option to help find exact DC operating-point solutions for large circuits.

See Also

.DC
.IC
.NODESET

.OPTION OPFILE

Outputs the operating point information to a file.

Syntax

```
.OPTION OPFILE= 0|1
```

Default 0

Description

Use this option to output the operating point information to a file.

- If *value* is `1`, operating point information is output to a file named <*design*>.dp#.
- If *value* is `0`, the operating point information outputs to stdout.

.OPTION OPTCON

Continues running a bisection analysis (with multiple .ALTER commands) even if optimization failed.

Syntax

```
.OPTION OPTCON=0|1
```

Default `0`

Description

Use this option to override how HSPICE treats bisection measure failure. With this option turned on, Instead of issuing an error and exiting the simulation, HSPICE treats a bisection search failure like a measurement failure and completes the simulation, or continues if `.ALTER` commands are specified.

HSPICE behaves in the following ways when `.OPTION OPTCOM=1`:

1. HSPICE no longer exits with a job aborted error message, instead it outputs failed parameters to the screen with a job concluded message.
2. It prints a `failed` value for the target parameters in the measurement file. However, if option `MEASFAIL=0` is present, it will log `0` in the measurement file instead of “failed”.

Examples

```
.option optcon=1
r1 1 0 2000
v1 1 0 3
.param target=0.5
.param x=opt1(0, 0, 1)
.model opt_model opt method=bisection relout=1e6
relin=0.0005
.meas tran y param = x goal = target
.tran 1.0e-10 1.0e-9 sweep optimize=opt1 results=y
 model=opt_model
.alter target=1.5
.param target=1.5
.alter target=0.75
.param target=0.75
.end
```

If a bisection search fails because of endpoints having the same sign, for example, screen output might appear as follows:

```
>info:          ***** hspice job concluded
 the maximum number of iterations (    14)was
 exceeded. however, results might be accurate.
x           = 3.556e-09
y           = 1.7103E+00
>info:          ***** hspice job concluded
 **Warning** endpoints have same sign in bisection
x           = failed
y           = failed
>info:          ***** hspice job concluded
Output stored in file => test.lis
```

See Also

.ALTER
.OPTION MEASFAIL

.OPTION OPTLST

Outputs additional optimization information.

Syntax

```
.OPTION OPTLIST=0|1|2|3
```

Default 0

Description

Use this option to output additional optimization information:

- OPTLIST=0: No information (default).
- OPTLIST=1: Prints parameter, Broyden update and bisection results information.
- OPTLIST=2: Prints gradient, error, Hessian, and iteration information.
- OPTLIST=3: Prints all of the above and Jacobian.

.OPTION OPTS

Prints current settings for all control options.

Syntax

```
.OPTION OPTS
```

Description

Use this option to print the current settings for all control options. If you change any of the default values of the options, the `OPTS` option prints the values that the simulation actually uses. The `BRIEF` option suppresses `OPTS`.

See Also

.OPTION BRIEF

.OPTION PARHIER (or) .OPTION PARHIE

Specifies scoping rules.

Syntax

```
.OPTION PARHIER=[GLOBAL|LOCAL]
```

Default GLOBAL

Description

Use this option to specify scoping rules.

Example

```
.OPTION parhier=<global | local>
.PARAM DefPwid=1u
.SUBCKT Inv a y DefPwid=2u DefNwid=1u
   Mp1 <MosPinList> pMosMod L=1.2u W=DefPwid
   Mn1 <MosPinList> nMosMod L=1.2u W=DefNwid
.ENDS
```

This example explicitly shows the difference between local and global scoping for using parameters in subcircuits.

.OPTION PATHNUM

Prints subcircuit path numbers instead of path names.

Syntax

```
.OPTION PATHNUM
```

Default `0`

Description

When set to `1`, this option prints subcircuit path numbers instead of path names.

.OPTION PHASENOISEKRYLOVDIM

Specifies the dimension of the Krylov subspace that the Krylov solver uses.

Syntax

```
.OPTION PHASENOISEKRYLOVDIM
```

Default `500`

Description

Specifies the dimension of the Krylov subspace that the Krylov solver uses. This must be an integer greater than zero.

See Also

.OPTION BPNMATCHTOL
.OPTION PHASENOISEKRYLOVITER
.OPTION PHASENOISETOL
.OPTION PHNOISELORENTZ

.OPTION PHASENOISEKRYLOVITER

Specifies the maximum number of Krylov iterations that the phase noise Krylov solver takes.

Syntax

```
.OPTION PHASENOISEKRYLOVITER
```

Default `1000`

Description

Specifies the maximum number of Krylov iterations that the phase noise Krylov solver takes. Analysis stops when the number of iterations reaches this value.

See Also

.OPTION BPNMATCHTOL
.OPTION PHASENOISEKRYLOVDIM
.OPTION PHASENOISETOL
.OPTION PHNOISELORENTZ

.OPTION PHASENOISETOL

Specifies the error tolerance for the phase noise solver.

Syntax

```
.OPTION PHASENOISETOL
```

Default `1e-8`

Description

Specifies the error tolerance for the phase noise solver. This must be a real number greater than zero.

See Also

.OPTION BPNMATCHTOL
.OPTION PHASENOISEKRYLOVDIM
.OPTION PHASENOISEKRYLOVITER
.OPTION PHNOISELORENTZ

.OPTION PHNOISELORENTZ

Turns on a Lorentzian model for the phase noise analysis.

Syntax

```
.OPTION PHNOISELORENTZ= 0|1|2
```

Default

`1`

Description

Turns on a Lorentzian model for the phase noise analysis.

- `0`: Uses a linear approximation to a lorentzian model
- `1` (default): Applies a lorentzian model to all noise sources
- `2`: Applies a lorentzian model to all non-frequency dependent noise sources

See Also

.OPTION BPNMATCHTOL
.OPTION PHASENOISEKRYLOVDIM
.OPTION PHASENOISEKRYLOVITER
.OPTION PHASENOISETOL

.OPTION PHNOISEAMPM

Allows you to separate amplitude modulation and phase modulation components in a phase noise simulation.

Syntax

```
.OPTION PHNOISEAMPM=0|1
```

Default

`0`

Description

Use this option to enable HSPICE RF to calculate separate amplitude (am) and phase modulation (pm) components using the output and measure syntax of a .PHASENOISE simulation. A value of 0 sets the Periodic AC (PAC) phase noise amplitude modulation (AM) component to zero and the results will be identical to earlier releases. A value of 1 calculates separate AM and phase noise components. When `.OPTION PHNOISEAMPM=1`, then

`.MEASURE PHASENOISE` extends output variables to the set:
`<am[noise]> <pm[noise]>`

Example

The following explicitly sets the calculation for separate am and pm calculation.

```
.opt phnoiseampm=1
```

See Also

.PHASENOISE
Amplitude Modulation/Phase Modulation Separation

.OPTION PIVOT

Selects a pivot algorithm.

Syntax

```
.OPTION PIVOT=x
```

Default 10

Description

Use this option to select a pivot algorithm. Use these algorithms to reduce simulation time and to achieve convergence in circuits that produce hard-to-solve matrix equations. PIVOT selects the numerical pivoting algorithm that is used to manipulate the matrices. Pivoting affects both DC and transient analysis. Usually the reason for choosing a pivot method other than either the default (10) or 0 is that the circuit contains both very large and very small conductances.To select the pivot algorithm, set PIVOT as follows:

- PIVOT=0: Original nonpivoting algorithm.
- PIVOT=1: Original pivoting algorithm.
- PIVOT=2: Picks the largest pivot in the row.
- PIVOT=3: Picks the best pivot in a row.
- PIVOT=10: Fast, nonpivoting algorithm; requires more memory.
- PIVOT=11: Fast, pivoting algorithm; requires more memory than PIVOT values less than 11.
- PIVOT=12: Picks the largest pivot in the row; requires more memory than PIVOT values less than 12.
- PIVOT=13: Fast, best pivot: faster; requires more memory than PIVOT values less than 13.

The fastest algorithm is PIVOT=13. This algorithm can improve simulation time up to ten times on very large circuits but requires substantially more memory for simulation.

Some circuits with large conductance ratios, such as switching regulator circuits, might require pivoting.

If PIV0T=0 or 10, HSPICE automatically changes from a nonpivoting to a row-pivot strategy if it detects any diagonal-matrix entry less than PIVTOL. This strategy provides the time and memory advantages of nonpivoting inversion and avoids unstable simulations and incorrect results. Use .OPTION NOPIV to

prevent HSPICE from pivoting. For very large circuits, `PIVOT=10`, `11`, `12`, or `13` can require excessive memory.

If HSPICE switches to pivoting during a simulation, it displays this message, followed by the node numbers that cause the problem:

```
pivot change on the fly
```

Use `.OPTION NODE` to cross-reference a node to an element. The `SPARSE` option is the same as `PIVOT`.

See Also

.OPTION NODE
.OPTION NOPIV
.OPTION PIVREF
.OPTION PIVREL
.OPTION PIVTOL

.OPTION PIVREF

Sets a pivot reference.

Syntax

```
.OPTION PIVREF=x
```

Default

`100.00x`

Description

Use this option to set a pivot reference. Use `PIVREF` in `PIVOT=11`, `12`, or `13` to limit the size of the matrix. The default is `1e+8`.

See Also

.OPTION PIVOT

.OPTION PIVREL

Sets the maximum and minimum ratio of a row or matrix.

Syntax

```
.OPTION PIVREL=x
```

Default `100u`

Description

Use this option to set the maximum and minimum ratio of a row or matrix. Use only if `PIVOT=1`. Large values for `PIVREL` can result in very long matrix pivot times; however, if the value is too small, no pivoting occurs. Start with small values of `PIVREL` by using an adequate but not excessive value for convergence and accuracy. The default is `1e-4`.

See Also

.OPTION PIVOT

.OPTION PIVTOL

Sets the absolute minimum value for which HSPICE accepts a matrix entry as a pivot.

Syntax

```
.OPTION PIVTOL=x
```

Default `1.00f`

Description

Use this option to set the absolute minimum value for which HSPICE accepts a matrix entry as a pivot. PIVTOL is used to prevent numeric overflow conditions like divide by 0. If the conductance is less than the value of PIVTOL, HSPICE rebuilds the matrix and chooses the PIVOT algorithm. If the conductance is greater than the value of PIVTOL, the PIVTOL value replaces the conductance in the matrix. When a non-pivot algorithm is selected by setting `PIVOT=0` or `10`, then pivtol is the minimum conductance in the matrix and not a pivot.

The default value of PIVTOL is 1e-15 and the range of PIVTOL is Min:1e-35, Max:1, excluding 0. The value of PIVTOL must be less than GMIN or GMINDC. Values that approach 1 increase the pivot. The example below shows how you can correct a “maximum conductance on node error.”

Note:

If PIVTOL is set too small, you run the risk of creating an overflow condition and a convergence problem. If you set the value to `0`, an out-of-bounds error is reported.

Example

If you get an error message such as:

```
**error**  maximum conductance on node 1:v75 } =(  9.2414D-23)
is less than pivtol in transient analysis.
Check hookup for this node, set smaller option pivtol and rerun.
```

—the error message informs that the node conductance value is less than the value of PIVTOL. Decrease the PIVTOL value so that it is less than the value in the error message. The valid range of pivtol values is between 1e-35 to 1, excluding 0. For this case a setting pivtol to 1e-25 resolves the error.

See Also

.OPTION GMIN
.OPTION GMINDC
.OPTION PIVOT

.OPTION POST

Saves simulation results for viewing by an interactive waveform viewer.

Note:

The behavior for .OPTION POST in HSPICE RF is different from the same option used in HSPICE.

Syntax

HSPICE Syntax

```
.OPTION POST=[0|1|2|3|ASCII|BINARY|CSDF]
```

HSPICE RF Syntax

```
.OPTION
   POST=[0|1|2|3|ASCII|BINARY|CSDF|NW|P|TW|UT|VCD|WDBA]
```

Default `0` or `1` if `POST` is is declared without a value.

Description

Use this option to save simulation results for viewing by an interactive waveform viewer.

Use `.OPTION POST` to provide output without specifying other parameters. The defaults for the `POST` option supply usable data to most parameters.

- `POST=0`: Does not output simulation results.
- `POST=1, BINARY`: (Default if `POST` is declared without a value) Output format is binary.
- `POST=2, ASCII`: Output format is ASCII.
- `POST=3`: Output format is New Wave binary (which enables you to generate *.tr0* files that are larger than 2 gigabytes on Linux platforms).
- `POST=CSDF`: Output format is Common Simulation Data Format (Viewlogic-compatible graph data file format).

Options available to HSPICE RF only:

- `POST=NW`: Output format is XP/AvanWaves.
- `POST=TW`: Output format is TurboWave.
- `POST=UT`: Output format is Veritools Undertow.
- `POST=VCD`: Output format is value change dump. Use with a .LPRINT command.

- POST=WDBA: Output format is XP/CosmosScope.
- POST=XP: Output format is XP/AvanWaves/CosmosScope.

By default, HSPICE outputs single precision for both time and signal data. If you want to get double precision data, in the netlist set:

```
.OPTION POST POST_VERSION=2001
```

Note:

.OPTION POST in HSPICE is *not* a global option to dump output in general and then use other options to specify another format. Other options such as PSF, CSDF, SDA, ZUKEN override POST if they are specified after POST, and vice versa. This is unlike HSPICE RF which allows values beyond [0|1|2|3|ASCII|BINARY|CSDF].

HSPICE uses the last output control option if multiple output control options are specified in the netlist.

Examples

In this example the option artist/PSF will overwrite the option post. HSPICE will use the control output option artist/psf.

```
.option artist=2 psf=2
.option post
```

In this example, the option artist/PSF overrides the option post.

```
.option post
.option artist=2 psf=2
```

See Also

.OPTION POST_VERSION

.OPTION POSTLVL

Limits the data written to your waveform file to a specified level of nodes.

Syntax

```
.OPTION POSTLVL=n
```

Default

0

Description

Limits the data written to your waveform file to the level of nodes specified by the *n* parameter. This option differs from POSTSTOP in that it specifies the signals of one given level at any level.

Example

```
.OPTION POSTLVL=2
```

This example limits the data written to the waveform file to only the second-level nodes.

See Also

.OPTION POSTTOP

.OPTION POST_VERSION

Specifies the post-processing output version for HSPICE/HSPICE RF.

Syntax

```
.OPTION POST_VERSION=x
```

Default `9601`

Description

Use this option to set the post-processing output version:

- `x=9007` truncates the node name in the post-processor output file to a maximum of 16 characters.
- `x=9601` sets the node name length for the output file consistent with input restrictions (1024 characters) and limits the number of output variables to 9999.
- `x=2001` uses an output file header that displays the correct number of output variables when the number exceeds 9999. This option also changes the digit-number precision in results files to match the value of `.OPTION NUMDGT` (when < 5).

By default, HSPICE outputs single precision for both time and signal data. If you want to get double precision data, in the netlist set:

```
.OPTION POST POST_VERSION=2001
```

If you set `.OPTION POST_VERSION=2001 POST=2` in the netlist, HSPICE returns more accurate ASCII results.

```
.OPTION POST_VERSION=2001
```

To use binary values (with double precision) in the output file, include the following in the input file:

```
*****************************************
.option post (or post=1) post_version=2001
*****************************************
```

For more accurate simulation results, comment this format.

Example

If you need to probe more than 9999 signals, set the `POST_VERSION` option to `2001`; for example,

```
.OPTION POST_VERSION=2001
```

HSPICE now outputs all the signals into a waveform file and the correct number of output signals is shown rather than **** when the number of signals exceeds 9999. You can load this waveform file in CosmosScope or AvanWaves to view the signals.

See Also

.OPTION NUMDGT
.OPTION POST

.OPTION POSTTOP

Limits the data written to the waveform file to data from only the top *n* level nodes.

Syntax

```
.OPTION POSTTOP=n
```

Description

Use this option to limit the data written to your waveform file to data from only the top *n* level nodes. This option outputs instances up to `n` levels deep. If you do not specify either the `.OPTION PROBE` or the `.OPTION POSTTOP` options, HSPICE/HSPICE RF outputs all levels. To enable the waveform display interface, you also need to specify the `.OPTION POST` option. This option differs from `.OPTION STOPLVL` in that it specifies the signals of one or multiple levels from the top level down.

Example

```
POSTTOP=1
```

This example limits the data written to the waveform file to only the top-level nodes.

See Also

.OPTION POST
.OPTION PROBE
.OPTION POSTLVL

.OPTION PROBE

Limits post-analysis output to only variables specified in `.PROBE` and `.PRINT` commands for HSPICE/HSPICE RF.

Syntax

```
.OPTION PROBE=0|1
```

Default `0`

Description

When turned on (`1`), allows you to set post-analysis output to only variables specified in `.PROBE`, `and PRINT` commands. `0`=off.

By default, HSPICE outputs all voltages and power supply currents in addition to variables listed in `.PROBE`, `and .PRINT` commands. Using this option can significantly decrease the sizes of simulation output files.

If `.OPTION PROBE` is not set:

- All node voltage and source currents are output to **.tr#, *.ac#, *.sw#* files.
- If measured, the resistor or MOSFET current is also output to **.tr#, *.ac#,* or **.sw#* files.
- If the resistor or MOSFET current are determined by measurement variables, and `.OPTION PUTMEAS` is reset (set to 0), these measurement variables are not output to waveform files.

See Also

.PRINT
.PROBE
.OPTION PROBE
.OPTION PUTMEAS

.OPTION PSF

In a standalone HSPICE simulation, specifies whether binary (Parameter Storage Format) or ASCII data is output. When used with HSPICE RF, specifies whether binary or ASCII data is output when you run an HSPICE simulation from t6he Cadence™ Virtuoso® Analog Design Environment.

Syntax

```
.OPTION PSF=0|1|2
```

Default 0

Description

Use this option to specify whether HSPICE RF outputs binary (Parameter Storage Format) or ASCII data when you run an HSPICE RF simulation through the Cadence Virtuoso Analog Design Environment.

If you use `.OPTON PSF` only (without `.OPTION ARTIST`), the value of *x* can be 1 or 2.

- If `.OPTION PSF=1`, HSPICE produces binary output.
- If `.OPTION PSF=2`, HSPICE produces ASCII output.

Note:

The PSF and SDA writers used in HSPICE rely on libraries that are not currently available in 64 bit versions, and 64 bit HSPICE cannot link the 32-bit libraries. If you inspect the log file, you will see the message `"**warning** 64bit cannot support option psf, artist or sda"`.

See Also

.OPTION ARTIST

.OPTION PURETP

Specifies the integration method to use for reversal time point in HSPICE/ HSPICE RF.

Syntax

```
.OPTION PURETP=x
```

Default

`0`

Description

Use this option to specify the integration method to use for reversal time point.

If you set `PURETP=1` and HSPICE finds non-convergence, it uses `TRAP` (instead of `Bbackward-Euler`) for the reversed time point.

Use this option with an `.OPTION METHOD=TRAP` command to help some oscillating circuits to oscillate if the default simulation process cannot satisfy the result.

See Also

.OPTION METHOD

.OPTION PUTMEAS

Controls the output variables listed in the .MEASURE command.

Syntax

```
.OPTION PUTMEAS=0|1
```

Default 1

Description

Use this option to control the output variables listed in the .MEASURE command.

- 0: Does not save variable values listed in the .MEASURE command into the corresponding output file (such as .tr#, .ac# or .sw#). This option decreases the size of the output file.
- 1: Default. Saves variable values listed in the .MEASURE command to the corresponding output file (such as .tr#, .ac# or .sw#). This option is similar to the output of HSPICE 2000.4.

See Also

.MEASURE (or) .MEAS

.OPTION PZABS

Sets absolute tolerances for poles and zeros.

Syntax

```
.OPTION PZABS=x
```

Default `1.0e-2`

Description

Use this option to set absolute tolerances for poles and zeros in Pole/Zero analysis. Use this option as follows:

If $(X_{\text{real}} + X_{\text{real}} < PZABS)$, then X_{real} and $X_{\text{imag}} = 0$. You can also use this option for convergence tests.

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION LSCAL
.OPTION PZTOL
.OPTION RITOL

.OPTION PZTOL

Sets the relative tolerance for poles and zeros.

Syntax

```
.OPTION PZTOL=x
```

Default `1.0e-6`

Description

Use this option to set relative tolerances for poles and zeros in Pole/Zero analysis.

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION LSCAL
.OPTION PZABS
.OPTION RITOL

.OPTION RANDGEN

Specifies the random number generator used in traditional Monte Carlo analysis.

Syntax

```
.OPTION RANDGEN= ['moa'|1|0]
```

Default 0

Description

Use this option to specify the random number generator used in HSPICE traditional Monte Carlo analysis. If `RANDGEN= 'moa'` or `1`, then a multiply-with-carry type random number generator with longer cycle is used. If `RANDGEN=0`, then the traditional random number generator is used.

Setting `.option RANDGEN='MOA'` or `1` is the equivalent to using `Replicates=`*`value`* in the Variation Block functionality for Monte Carlo when sampling using Latin Hypercube. See Latin Hyper Cube Sampling in the *HSPICE User Guide: Simulation and Analysis*.

Note:

The `.OPTION SEED` command is also valid for the new random number generator without usage change.

See Also

.OPTION RUNLVL
.OPTION SEED

.OPTION RELH

Sets the relative current tolerance from iteration to iteration through voltage-defined branches.

Syntax

```
.OPTION RELH=x
```

Default `50.00m`

Description

Use this option to set the relative current tolerance through voltage-defined branches (voltage sources and inductors) from iteration to iteration.

This option can also be used to check current convergence, but only if the value of the `ABSH` option is greater than zero.

See Also

.OPTION ABSH

.OPTION RELI

Sets the relative error/tolerance change from iteration to iteration.

Syntax

```
.OPTION RELI=x
```

Default `10.00m`

Description

Use this option to set the relative error/tolerance change from iteration to iteration.

This option determines convergence for all currents in diode, BJT, and JFET devices. (`RELMOS` sets tolerance for MOSFETs). This value is the change in current from the value calculated at the previous timepoint.

- Default`=0.01` for `.OPTION KCLTEST=0`.
- Default`=1e-6` for `.OPTION KCLTEST=1`.

See Also

.OPTION RELMOS
.OPTION KCLTEST

.OPTION RELMOS

Sets the relative error tolerance for drain-to-source current from iteration to iteration.

Syntax

```
.OPTION RELMOS=x
```

Default `50.00m (0.05 or 5%)`

Description

Use this option to set the relative error tolerance for drain-to-source current from iteration to iteration.

This option determines convergence for currents in MOSFET devices while `.OPTION RELI` sets the tolerance for other active devices.

This option also sets the change in current from the value calculated at the previous timepoint. HSPICE uses the `.OPTION RELMOS` value only if the current is greater than the `.OPTION ABSMOS` floor value.

Min value: 1e-07; Max value 10.

See Also

.OPTION ABSMOS
.OPTION RELI
.OPTION RELMOS

.OPTION RELQ

Sets the timestep size from iteration to iteration.

Syntax

```
.OPTION RELQ=x
```

Default `10.00m`

Description

Use this option in the timestep algorithm for local truncation error (`LVLTIM=2`). If the capacitor charge calculation in the present iteration exceeds that of the past iteration by a percentage greater than the `RELQ` value, then HSPICE reduces the internal timestep (delta). The default is `0.01`.

See Also

.OPTION LVLTIM

.OPTION RELTOL

Sets the relative error tolerance for voltages from iteration to iteration.

Syntax

```
.OPTION RELTOL=x
```

Default `1e-3`

Description

Use this option to set the relative error tolerance for voltages from iteration to iteration. Min value: 1e-20; Max value: 10.

Use this option with the `ABSV` option to determine voltage convergence. Increasing `x` increases the relative error. This option is the same as the `RELV` option. The `RELI` and `RELVDC` options default to the `RELTOL` value.

See Also

.OPTION ABSV
.OPTION RELI
.OPTION RELV
.OPTION RELVDC

.OPTION RELV

Sets the relative error tolerance for voltages from iteration to iteration.

Syntax

```
.OPTION RELV=x
```

Default `1.00m`

Description

Use this option to set the relative error tolerance for voltages from iteration to iteration.

If voltage or current exceeds the absolute tolerances, a `RELV` test determines convergence. Increasing `x` increases the relative error. You should generally maintain this option at its default value. It conserves simulator charge. For voltages, this option is the same as the `RELTOL` option. The default is `1e-3`. Min value: 1e-20; Max value: 10.

See Also

.OPTION RELTOL

.OPTION RELVAR

Sets the relative voltage change for `LVLTIM=1` or `3` from iteration to iteration.

Syntax

```
.OPTION RELVAR=x
```

Default `300.00m`

Description

Use this option to set the relative voltage change for `LVLTIM=1` or `3` from iteration to iteration.

Use this option with the `ABSVAR` and `DVDT` timestep algorithm. If the node voltage at the current timepoint exceeds the node voltage at the previous timepoint by `RELVAR`, then HSPICE reduces the timestep and calculates a new solution at a new timepoint. The default is `0.30`, or 30 percent.

For additional information, see "DVDT Dynamic Timestep" in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.OPTION ABSVAR
.OPTION DVDT
.OPTION LVLTIM

.OPTION RELVDC

Sets the relative error tolerance for voltages from iteration to iteration.

Syntax

```
.OPTION RELVDC=x
```

Default `1.00m`

Description

Use this option to set the relative error tolerance for voltages from iteration to iteration.

If voltages or currents exceed their absolute tolerances, the `RELVDC` test determines convergence. Increasing the *x* parameter value increases the relative error. You should generally maintain `RELVDC` at its default value to conserve simulator charge.

See Also

.OPTION RELTOL

.OPTION RESMIN

Specifies the minimum resistance for all resistors.

Syntax

```
.OPTION RESMIN=x
```

Default `10.00u`

Description

Use this option to specify the minimum resistance for all resistors, including parasitic and inductive resistances. The range is `1e-15` to `10` ohms.

.OPTION RISETIME (or) .OPTION RISETI

Specifies the smallest signal risetime to be supported in elements and analyses that are sensitive to frequency bandwidth and time scale constraints.

Syntax

```
.OPTION RISETIME=x
```

Default Calculated automatically (see below)

Description

Use this option to specify the smallest signal risetime to be anticipated when analyzing certain elements that have frequency dependencies. Several HSPICE elements require some knowledge regarding either their maximum frequency of operation, or the minimum signal rise time to be expected. This is particularly true of elements that are described in the frequency domain, yet require time-domain simulation. The `RISETIME` option is used to establish time scale and frequency scale information needed for inverse Fourier transform and convolution calculations.

In the W-element (transmission line) model, `RISETIME` is used to determine the maximum signal frequency to be taken into account for frequency dependencies such as skin effect, and dielectric loss (non-zero Rs or Gd).

In the S-element (scattering-parameter) based model, the reciprocal of `RISETIME` sets the maximum signal frequency (`FMAX`) value used for the S-parameter analysis.

In the U-element (lumped transmission line) model, `RISETIME` is used to set the number of lumps according to the equation:

$$\#\text{lumps} = MIN\left[20,\ 1 + 20 \cdot \left(\frac{TDeff}{RISETIME}\right)\right]$$

where, TD_{eff} is the end-to-end delay in a transmission line.

When needed, HSPICE automatically calculates a default value for `RISETIME` as follows:

- 25% of the `tstep` value specified with the `.TRAN` command.
- The time corresponding to a 90-degree phase shift for the highest frequency specified in `SIN`, `SFFM`, and `AM` sources.
- The smallest delay time, rise time, fall time, or time increment used in `PULSE`, `EXP`, and `PWL` sources.

See Also

.TRAN
.OPTION WACC
.OPTION WDELAYOPT

.OPTION RITOL

Sets the minimum ratio value for the (real/imaginary) or (imaginary/real) parts of the poles or zeros.

Syntax

```
.OPTION RITOL=x
```

Default `1.0e-2`

Description

Use this option to set the minimum ratio value for the (real/imaginary) or (imaginary/real) parts of the poles or zeros. Use the `RITOL` option as follows. if: $|X_{\text{imag}}| \leq RITOL \cdot |X_{\text{real}}|$, then $X_{\text{imag}} = 0$. If $|X_{\text{real}}| \leq RITOL \cdot |X_{\text{imag}}|$, then $X_{\text{real}} = 0$.

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION LSCAL
.OPTION PZABS
.OPTION PZTOL
.PZ

.OPTION RMAX

Sets the TSTEP multiplier, which controls the maximum value for the internal timestep delta fore HSPICE/HSPICE RF.

Syntax

```
.OPTION RMAX=x
```

Default 5

Description

Use this option to set the TSTEP multiplier, which controls the maximum value (DELMAX) for the delta of the internal timestep:

```
DELMAX=TSTEP x RMAX
```

- The default is 5 if DVDT is 4 and LVLTIM is 1.
- Otherwise, the default is 2.

Min value: 1e-9; Max value: 1e+9. The RMAX value cannot be smaller than RMIN.

See the "Timestep Control for Accuracy" section in the *HSPICE User Guide: Simulation and Analysis*.

See Also

.OPTION DELMAX
.OPTION DVDT
.OPTION LVLTIM

.OPTION RMIN

Sets the minimum value of delta (internal timestep).

Syntax

```
.OPTION RMIN=x
```

Default `1.00n`

Description

Use this option to set the minimum value of delta (internal timestep). An internal timestep smaller than `RMIN` x `TSTEP`, terminates the transient analysis, and reports an internal “timestep too small” error. If the circuit does not converge in `IMAX` iterations, delta decreases by the amount you set in the `FT` option. The default is `1.0e-9`. Min value: 1e-15.

See Also

.OPTION FT
.OPTION IMAX

.OPTION RUNLVL

Controls runtime speed and simulation accuracy.

Syntax

```
.OPTION RUNLVL= 0|1|2|3|4|5|6
```

Default: `3`

Description

Use this option to control runtime speed and simulation accuracy. Higher values of `RUNLVL` result in higher accuracy and longer simulation runtimes, while lower values result in lower accuracy and faster simulation runtimes.

For HSPICE:

The `RUNLVL` option setting controls the scaling of all simulator tolerances simultaneously, affecting timestep control, transient analysis convergence, and model bypass tolerances all at once. Higher values of `RUNLVL` result in smaller timestep sizes and could result in more Newton-Raphson iterations to meet stricter error tolerances. `RUNLVL` settings affect transient analysis only.

`RUNLVL` can be set to `0` (to disable) `1`, `2`, `3`, `4`, `5`, or `6`:

- `1`: Lowest simulation runtime
- `2`: More accurate than `RUNLVL=1` and faster than `RUNLVL=3`
- `3`: Default value, similar to HSPICE's original default mode
- `4`: More accurate than `RUNLVL=3` and faster than `RUNLVL=5`
- `5` or `6`: Corresponds to HSPICE's standard accurate mode for most circuits:
  - `5` is similar to the standard accurate mode in HSPICE
  - `6` has the highest accuracy

If `RUNLVL` is specified in the netlist without a value, the value is the default, `3`.

If `.OPTION ACCURATE` is specified in the netlist together with a value of `RUNLVL` greater than 0, the value of `RUNLVL` is limited to 5 or 6; specifying a specifying a RUNLVL value of 1, 2, 3, or 4 defaults to 5.

If `.OPTION RUNLVL` is *NOT* specified, there is an order dependency with GEAR and ACCURATE options, as follows:

`.option ACCURATE method=GEAR` -> ACCURATE is not in use
`.option method=GEAR ACCURATE` -> GEAR + ACCURATE effects

With RUNLVL, if GEAR is used, GEAR only determines the numeric integration method; anything else is controlled by RUNLVL; there is no order dependency with RUNLVL and GEAR.

Since there is no order dependency with RUNLVL and GEAR, or RUNLVL and ACCURATE, then:

```
.OPTION ACCURATE method=GEAR RUNLVL
```

is equivalent to

```
.OPTION method=GEAR ACCURATE RUNLVL
```

The RUNLVL option interacts with other options as follows:

- Regardless of its position in the netlist, RUNLVL ignores the following step control-related options which are replaced by automated algorithms:

  LVLTIM DVDT FT FAST TRTOL ABSVAR

  RELVAR RELQ CHGTOL DVTR IMIN ITL3

- See the notes to the table below for discussion of options ACCURATE and BYPASS in relation to RUNLVL if it is specified in the netlist.
- The tstep value specified with the .TRAN command affects timestep control when a RUNLVL option is used. Timestep values larger than tstep*RMAX use a tighter timestep control tolerance.

If RUNLVL is invoked, you can disable it using the following procedure:

1. Re-invoke the $installdir/bin/config program and unselect the .OPTION RUNLVL setting in the *hspice.ini* which disables it for the whole group of simulation jobs.
2. Copy $installdir/hspice.ini to your HOME directory and customize it by adding .option runlvl=0, which disables it for all of your simulation jobs.
3. Add .option runlvl=0 to your current simulation job. (If you are using SimIF with the RUNLVL option, move the slider to the left to 0 to turn off the runlvl setting.)

To learn more about the initialization file, refer to the *HSPICE User Guide: Simulation and Analysis*, Chapter 2, Setup and Simulation, "Initialization File (hspice.ini)." For information on how RUNLVL values affect other options, see the following section, and also see RUNLVL=N and RUNLVL, ACCURATE, FAST, GEAR method in Appendix B of this manual.

For HSPICE RF:

While HSPICE RF supports `.OPTION RUNLV`, this option is most compatible with HSPICE. For HSPICE RF, the `SIM_ACCURACY` option gives you a more continuous range of settings. You can use `.OPTION RUNLVL` to control runtime speed and simulation accuracy. Higher values of `RUNLVL` result in higher accuracy and longer simulation runtimes, while lower values result in lower accuracy and faster simulation runtimes.

`.OPTION RUNLVL` maps to `.OPTION SIM_ACCURACY` as follows:

- RUNLVL=1: SIM_ACCURACY=0.5
- RUNLVL=2: SIM_ACCURACY=0.75
- RUNLVL=3: SIM_ACCURACY=1
- RUNLVL=4: SIM_ACCURACY=5
- RUNLVL=5: SIM_ACCURACY=10
- RUNLVL=6: SIM_ACCURACY=100

Interactions Between .OPTION RUNLVL and Other Options

Since the latest algorithm invoked by RUNLVL sets the timestep and error tolerance internally, many transient error tolerance and timestep control options are no longer valid; furthermore, to assure the most efficiency of the new RUNLVL algorithm, you should let the new engine manage everything itself. Options that are recommended not to tune are listed in the table, as well.

Note:

Once RUNLV is set, it does not = 0.

Option	Default value without RUNLVL	Default value with RUNLVL=3	User definition ignored	Recommend not to tune
ABSV/VNTOL	50u	50u		x
ABSVAR	500m	500m	x	
ACCURATE [a]	0	0		
BYPASS [a]	2	2 for RUNLVL=1-6		
CHGTOL	1.0f	1.0f	x	
DI	100	100		x

Option	Default value without RUNLVL	Default value with RUNLVL=3	User definition ignored	Recommend not to tune
DVDT	4	4	x	
DVTR	1.0k	1.0k	x	
FAST [b]	0	0	x	
FS	250m	250m		x
FT	250m	250m	x	
IMIN/ITL3	3	3	x	
LVLTIM	1	4	x	
METHOD [c]	TRAP	TRAP		
RELQ	10m	10m	x	
RELTOL	1.0m	1.0m		x
RELV	1.0m	1.0m		x
RELVAR	300.0m	300.0m	x	
RMAX	5	5	x	
RMIN	1.0n	1.0n		x
TRTOL	7	7	x	

a. ACCURATE and BYPASS notes:
1. If .option ACCURATE is set, then the RUNLVL value is limited to 5 or 6. Specifying a RUNLVL less than 5 results in a simulation at RUNLVL=5. When both ACCURATE and RUNLVL are set, the RUNLVL algorithm will be used.

2. When RUNLVL is set, BYPASS is set to 2. Users can redefine the BYPASS value by setting .option BYPASS=value; this behavior is independent of the order of RUNLVL and BYPASS;

b. The FAST option is disabled by the RUNLVL option; setting the RUNLVL value to 1 is comparable to setting the FAST option.

c. RUNLVL can work with METHOD=GEAR; in cases where GEAR only determines the numeric integration method during transient analysis, the other options that were previously set by GEAR (when there is no RUNLVL) now are determined by the RUNLVL mode. This behavior is independent of the order of RUNLVL and METHOD. See below.

The interactions of RUNLVL and GEAR are shown in the table below.

Option	GEAR without RUNLVL	GEAR with RUNLVL=3
BYPASS	0	2
BYTOL	50u	100u
LVLTIM	2	Disabled by runlvl
MAXORD	2	3 for RUNLVL=6 2 for RUNLVL=1-5
MBYPASS	1	2
RMAX	2	Disabled by runlvl

See Also

.OPTION ACCURATE
.OPTION BYPASS
.OPTION DVDT
.OPTION LVLTIM
.OPTION METHOD
.OPTION RELTOL
.TRAN
.OPTION SIM_ACCURACY (RF)

.OPTION SAVEHB

Saves the final-state variable values from an HB simulation.

Syntax

```
.OPTION SAVEHB='filename'
```

Description

Use this option to save the final state (that is, the no-sweep point or the steady state of the first sweep point) variable values from an HB simulation to the specified file.

This file can be loaded as the starting point for another simulation by using a `LOADHB` option.

See Also

.HB
.OPTION LOADHB

.OPTION SAVESNINIT

Saves the operating point at the end of Shooting Newton initialization (sninit).

Syntax

```
.OPTION SAVESNINIT="filename"
```

Description

Use this option to save an operating point file at the end of a SN initialization for use as initial conditions for another Shooting Newton analysis. For more information, see SN Steady-State Time Domain Analysis in the *HSPICE User Guide: RF Analysis*.

See Also

.SN

.OPTION LOADSNINIT

.OPTION SAVESNINIT

.OPTION SNACCURACY

.OPTION SNMAXITER

.OPTION SCALE

Sets the element scaling factor for HSPICE/HSPICE RF.

Syntax

```
.OPTION SCALE=x
```

Default `1.00`

Description

Use this option to scale geometric element instance parameters whose default unit is meters. You can also use this option with `.OPTION GEOSHRINK` to scale an element even more finely. The effective scaling factor is the product of the two parameters; HSPICE will use `scale*geoshrink` to scale the parameters/dimensions.

In HSPICE, the possible geometrical instance parameters include width, length, or area for both passive and active devices, in addition to the commonly known MOSFET parameters such as AS, AD, PS, PD, and so on.

- For active elements, the geometric parameters are:

  Diode — W, L, Area
  JFET/MESFET — W, L, Area
  MOS — W, L, AS, AD, PS, PD

- For passive elements having values calculated as a function geometry, the geometric parameters are:

  Resistor — W, L
  Capacitor — W, L

See Also

.OPTION GEOSHRINK
.OPTION CMIUSRFLAG

.OPTION SCALM

Sets the model scaling factor.

Syntax

```
.OPTION SCALM=x
```

Default

1

Description

Use this option to set the scaling factor defined in a .MODEL command for an element. See the HSPICE Elements and Device Models Manual for parameters that this option scales. For MOSFET devices, this option is ignored in Level 49 and higher model levels. See the HSPICE Reference Manual: MOSFET Models for levels available to the SCALM option.

See Also

.MODEL

.OPTION SEARCH

Automatically accesses a library or individual library files.

Syntax

```
.OPTION SEARCH='directory_path' [path_name]
```

Description

Use this option to auto-access a library, or, using `path_name`, to search for library (**.lib*) files. The commands `.LIB` `.INC`, and `.LOAD` each searches for the file. In addition, HSPICE supports `.OPTION SEARCH` for `.VEC` commands. The path can be given as '/remote/home1/aa' or as '../'

Example

```
.OPTION SEARCH='$installdir/parts/vendor'
```

This example searches for models in the `vendor` subdirectory, under the `$installdir/parts` installation directory (see Figure 13). The `parts` directory contains the DDL subdirectories.

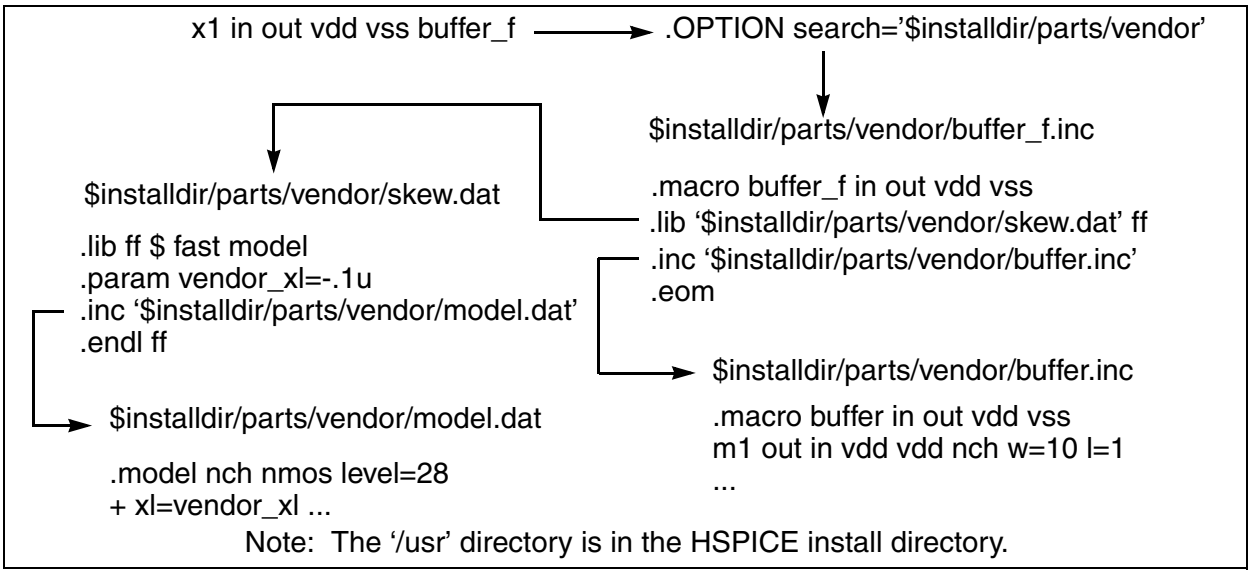

Figure 13 Vendor Library Usage

See Also:

.INCLUDE
.LIB
.LOAD

.OPTION SEED

Specifies the starting seed for the random-number generator in Monte Carlo analysis.

Syntax

```
.OPTION SEED=x | 'random'
```

Default

`1`

Description

Use this option to specify the starting seed for the random-number generator in HSPICE Monte Carlo analysis. The minimum value is 1; the maximum value of is 259200 of SEED. If `SEED='random'`, HSPICE assigns a random number between 1 and 259200 according to the system clock and prints it in the *.lis* file for you to debug. `.OPTION SEED` is supported by HSPICE and it does not exist in the RF flow which uses only the traditional Monte Carlo functionality.

See Also

.OPTION RANDGEN

.OPTION SIM_ACCURACY

Sets and modifies the size of timesteps.

Syntax

`.OPTION SIM_ACCURACY=`*value*

Default Conditional, see below

Description

Use this option to set and modify the size of timesteps. This option applies to all modes and tightens all tolerances, such as: Newton-Raphson tolerance, local truncation error, and other errors.

The *value* must be a positive number. The default is 1. If you specify `.OPTION ACCURATE`, the default value is 10; you can use `.option sim_accuracy=10` instead of `.option accurate`. They are interchangeable. You can set `.option sim_accuracy=10` if you have not set previous `sim_accuracy` settings that are 10 or greater or have previously set `.option accurate`.

To set global accuracy, use `.OPTION SIM_ACCURACY=`*n*, where *n* is a number greater than 0.

You can apply different accuracy settings to different blocks or time intervals. The syntax to set accuracy on a block, instance, or time interval is similar to the settings used for a power supply.

Example

This example sets accuracy to 3 for the XNAND1 and XNAND2 instances and 4 for all instances of the INV subcircuit. Globally, the accuracy is 2. If accuracy settings conflict, then HSPICE RF uses the higher accuracy value. At 12.0ns before the end of the simulation, the global accuracy level is 5. Because this is higher than 2, 3, or 4, it overrides all previous settings.

```
.OPTION SIM_ACCURACY=2
.OPTION SIM_ACCURACY=3 | XNAND1 XNAND2
.OPTION SIM_ACCURACY=4 | @INV
.OPTION SIM_ACCURACY=5 | 12.0n
.OPTION SIM_ACCURACY=5 | 20n
.OPTION SIM_ACCURACY=3 | 40ns
.OPTION SIM_ACCURACY=5 | 20ns 3 | 35ns 7 | 50ns
```

See Also

.OPTION FFT_ACCURATE
.OPTION ACCURATE

.OPTION SIM_DELTAI

Sets the selection criteria for current waveforms in WDB and NW format.

Syntax

```
.OPTION SIM_DELTAI=value
```

Default `0` amps

Description

Use this option to set the selection criteria for RF current waveforms in WDB and NW format (see "Eliminating Current Datapoints" in the *HSPICE User Guide: RF Analysis).*The *value* parameter specifies the amount of change. NW format is not supported by Monte Carlo analysis.

Example

In this example, at the *n* timestep, HSPICE RF saves only datapoints that change by more than 0 amps from previous values at the *n-1* timestep.

```
.OPTION SIM_DELTAI = 0amps
```

See Also

.OPTION SIM_DELTAV

.OPTION SIM_DELTAV

Sets the selection criteria for current waveforms in WDB and NW format.

Syntax

```
.OPTION SIM_DELTAV=value
```

Default 1mv

Description

Sets the selection criteria for RF current waveforms in WDB and NW format (see "Eliminating Voltage Datapoints" in the *HSPICE User Guide: RF Analysis).* NW format is not supported by Monte Carlo analysis.

The *value* parameter specifies the amount of change.

Example

In this example, at the *n* timestep, HSPICE RF saves only datapoints that change by more than 1 mV from their previous values at the *n-1* timestep.

```
.OPTION SIM_DELTAV = 1mv
```

See Also

.OPTION SIM_DELTAI

.OPTION SIM_DSPF

Runs simulation with standard DSPF expansion of all nets from one or more DSPF files.

Syntax

```
.OPTION SIM_DSPF="[scope] dspf_filename"
```

Description

Use this option to run simulation with standard DSPF expansion of all nets from one or more DSPF files.

`scope` can be a subcircuit definition or an instance. If you do not specify `scope`, it defaults to the top-level definition.

You can repeat this option to include more DSPF files.

This option can accelerate simulation by more than 100%. You can further reduce total CPU time by including the `.OPTION SIM_LA` in the netlist.

For designs of 5K transistors or more, including `.OPTION SIM_DSPF_ACTIVE` in your netlist to expand only active nodes also provides a performance gain.

Note:

HSPICE RF requires both a DSPF file and an ideal extracted netlist. Only flat DSPF files are supported; hierarchy commands, such as `.SUBCKT` and .x1 are ignored.

For additional information, see "Post-Layout Back-Annotation" in the *HSPICE User Guide: RF Analysis*.

Example 1

In this example, the parasitics in the DSPF file are mapped into the hierarchical ideal netlist.

```
$  models
.MODEL p pmos
.MODEL n nmos

.INCLUDE add4.dspf
.OPTION SIM_DSPF="add4.dspf"
.VEC "dspf_adder.vec"
.TRAN 1n 5u
vdd vdd 0 3.3
.OPTION POST
.END
```

The SIM_DSPF option accelerates the simulation by more than 100%. By using the SIM_LA option at the same time, you can further reduce the total CPU time:

```
$  models
.MODEL p pmos
.MODEL n nmos
.INCLUDE add4.dspf
.OPTION SIM_DSPF="add4.dspf"
.OPTION SIM_LA=PACT
.VEC "dspf_adder.vec"
.TRAN 1n 5u
vdd vdd 0 3.3
.OPTION POST
.END
```

Example 2

In this example, the x1.spf DSPF file is back-annotated to the x1 top-level instance. It also back-annotates the inv.spf DSPF file to the inv subcircuit.

```
.OPTION SIM_DSPF = "x1 x1.spf"
.OPTION SIM_DSPF = "inv inv.spf"
```

See Also

.OPTION SIM_LA
.OPTION SIM_DSPF_ACTIVE
.OPTION SIM_DSPF_SCALEC
.OPTION SIM_DSPF_SCALER
.OPTION SIM_SPEF

.OPTION SIM_DSPF_ACTIVE

Runs simulation with selective DSPF expansion of active nets from one or more DSPF files.

Syntax

```
.OPTION SIM_DSPF_ACTIVE="active_node"
```

Description

Use this option to run simulation with selective DSPF expansion of active nets from one or more DSPF files. HSPICE RF performs a preliminary verification run to determine the activity of the nodes and generates two ASCII files: *active_node*.rc and *active_node*.rcxt. These files save all active node information in both Star-RC and Star-RCXT formats. If an *active_node* file is not generated from the preliminary run, no nets are expanded. Active nets are added to the file as they are identified in the subsequent transient simulation. A second simulation run using the same file and option causes only the nets listed in the *active_node* file to be expanded. It is possible that activity changes are due to timing changes caused by expansion of the active nets. In this case, additional nets are listed in the *active_node* file and a warning is issued.

HSPICE RF uses the *active_node* file and the DSPF file with the ideal netlist to expand only the active portions of the circuit. If a net is latent, then HSPICE RF does not expand it, which saves memory and CPU time.

For additional information, see “Selective Post-Layout Flow” in the *HSPICE User Guide: RF Analysis*.

Example

In the following example, an active net in which the tolerance of the voltage change is greater than 0.5V is saved to both the active.rc and active.rcxt files. Based on these files, HSPICE RF back-annotates only the active parasitics from x1.spf and s2.spf to the x1 and x2 top-level instances.

```
.OPTION SIM_DSPF = "x1 x1.spf"
.OPTION SIM_DSPF = "x2 x2.spf"
.OPTION SIM_DSPF_ACTIVE = "active"
.OPTION SIM_DSPF_VTOL = 0.5V
```

See Also

.OPTION SIM_DSPF
.OPTION SIM_DSPF_MAX_ITER
.OPTION SIM_DSPF_VTOL
.OPTION SIM_SPEF_ACTIVE

.OPTION SIM_DSPF_INSERROR

Skips unmatched instances.

Syntax

```
.OPTION SIM_DSPF_INSERROR=ON | OFF
```

Default `OFF`

Description

Use this option to skip unmatched instances.

- ON: Skips unmatched instances
- OFF: Does not skip unmatched instances.

For additional information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

.OPTION SIM_DSPF_LUMPCAPS

Connects a lumped capacitor with a value equal to the net capacitance for instances missing in the hierarchical netlist.

Syntax

```
.OPTION SIM_DSPF_LUMPCAPS=ON | OFF
```

Default `ON`

Description

Use this option to connect a lumped capacitor with a value equal to the net capacitance for instances missing in the hierarchical netlist.

- `ON` (default): Adds lumped capacitance while ignoring other net contents
- `OFF`: Uses net contents

For additional information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

.OPTION SIM_DSPF_MAX_ITER

Specifies the maximum number of simulation runs for the second selective DSPF expansion pass.

Syntax

```
.OPTION SIM_DSPF_MAX_ITER=value
```

Default 1

Description

Use this option to specify the maximum number of simulation runs for the second selective DSPF expansion pass.

The *value* parameter specifies the number of iterations for the second simulation run.

Some of the latent nets might turn active after the first iteration of the second simulation run. In this case:

- Resimulate the netlist to ensure the accuracy of the post-layout simulation.
- Use this option to set the maximum number of iterations for the second run. If the active_node remains the same after the second simulation run, HSPICE RF ignores these options.

For details, see "Selective Post-Layout Flow" *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_DSPF_ACTIVE
.OPTION SIM_DSPF_VTOL

.OPTION SIM_DSPF_RAIL

Controls whether power-net parasitics are back-annotated

Syntax

```
.OPTION SIM_DSPF_RAIL=ON | OFF
```

Default `OFF`

Description

Use this option to control whether power-net parasitics are back-annotated.

- `OFF`: Do not back-annotate nets in a power rail
- `ON`: Back-annotate nets in a power rail

For additional information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

.OPTION SIM_DSPF_SCALEC

Scales the capacitance values in a DSPF file for a standard DSPF expansion flow.

Syntax

```
.OPTION SIM_DSPF_SCALEC=scaleC
```

Description

Use this option to scale the capacitance values in a DSPF file for a standard DSPF expansion flow.

The *scaleC* parameter specifies the scale factor.

For additional information, see “Additional Post-Layout Options” in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA
.OPTION SIM_DSPF_ACTIVE

.OPTION SIM_DSPF_SCALER

Scales the resistance values in a DSPF file for a standard DSPF expansion flow.

Syntax

```
.OPTION SIM_DSPF_SCALER=scaleR
```

Description

Use this option to scale the resistance values in a DSPF file for a standard DSPF expansion flow.

The *scaleR* specifies the scale factor.

For additional information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA
.OPTION SIM_DSPF_ACTIVE

.OPTION SIM_DSPF_VTOL

Specifies multiple DSPF active thresholds.

Syntax

```
.OPTION SIM_DSPF_VTOL="value | scope1 scope2 ...
+ scopen"
```

Default `0.1V`

Description

Use this option to specify multiple DSPF active thresholds.

- The *value* parameter specifies the tolerance of voltage change. This value should be relatively small compared to the operating range of the circuit or smaller than the supply voltage.
- *scopen* can be a subcircuit definition that uses a prefix of "@" or a subcircuit instance.

HSPICE RF performs a second simulation run by using the active_node file, the DSPF, and the hierarchical LVS ideal netlist to back-annotate only active portions of the circuit. If a net is latent, HSPICE RF does not expand the net. This saves simulation runtime and memory.

By default, HSPICE RF performs only one iteration of the second simulation run. Use the `SIM_DSPF_MAX_ITER` option to change this setting.

For additional information, see "Selective Post-Layout Flow" in the *HSPICE User Guide: RF Analysis*.

Example 1

In this example, the first line sets the sensitivity voltage to 0.01V. Subcircuit definition *snsamp* and the subcircuit instance xvco have full parasitics if their nodes move more than 0.01V during active nodes generation. In the second line, *xand* and *xff* are less sensitive than the default, indicating that they are not sensitive to parasitics.

```
.OPTION SIM_DSPF_VTOL="0.01 | @snsamp xvco"
.OPTION SIM_DSPF_VTOL="0.25 | xand xff"
```

Example 2

In this example, the sense amp circuit uses full parasitics if their nodes move more than 0.01V during active-node generation. The inv subcircuit definition is less sensitive than the default so the nodes are less sensitive to the parasitics.

```
.OPTION SIM_DSPF = "inv inv.spf"
.OPTION SIM_DSPF = "senseamp senseamp.spf"
.OPTION SIM_DSPF_ACTIVE = "activenet"
.OPTION SIM_DSPF_VTOL = "0.15 | @inv"
.OPTION SIM_DSPF_VTOL = "0.01 | @senseamp"
```

See Also

.OPTION SIM_DSPF_ACTIVE
.OPTION SIM_DSPF_MAX_ITER

.OPTION SIM_LA

Activates linear matrix (RC) reduction for HSPICE/HSPICE RF.

Syntax

```
.OPTION SIM_LA=PACT|PI| 0 | 1 | 2 ]
```

Default `1 or PACT`

Description

Use this option to activate linear matrix reduction. `SIM_LA` does not reduce a node used by any analysis command, such as `.PROBE`, `.MEASURE`, and so on

This option accelerates the simulation of circuits that include large linear RC networks by reducing all matrixes that represent RC networks.

- `0` turns off `SIM_LA`
- `1` is the equivalent of `PACT`, which selects the Pole Analysis via Congruence Transforms (PACT) algorithm to reduce RC networks in a well-conditioned manner, while preserving network stability.
- `2` invokes the `PI` algorithm to create PI models of the RC networks.

  Note:

  `SIM_LA` does not reduce a node used by any analysis command, such as `.PROBE`, `.MEASURE`, and so on.

- If `sim_la` is not specified in the input file, the **.lis* returns `sim_la=0`.
- If `sim_la` is specified with no value or `sim_la=pact`, the **.lis* file returns `sim_la=1`.
- If `sim_la=pi`, the **.lis* file returns `sim_la=2`.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: Simulation and Analysis* or "Linear Acceleration" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_DSPF
.OPTION LA_FREQ
.OPTION LA_MAXR
.OPTION LA_MINC
.OPTION SIM_LA_MINMODE
.OPTION LA_TIME
.OPTION LA_TOL

.OPTION SIM_LA_FREQ

Specifies the upper frequency for which accuracy must be preserved.

Syntax

```
.OPTION SIM_LA_FREQ=value
```

Default 1GHz

Description

Use this option to specify the upper frequency for which accuracy must be preserved. The *value* parameter specifies the upper frequency for which the PACT algorithm must preserve accuracy. If *value* is 0, the algorithm drops all capacitors because only DC is of interest.

The maximum frequency required for accurate reduction depends on both the technology of the circuit and the time scale of interest. In general, the faster the circuit, the higher the maximum frequency. For additional information, see "Linear Acceleration" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA
.OPTION SIM_LA_TIME

.OPTION SIM_LA_MAXR

Specifies the maximum resistance for linear matrix reduction.

Syntax

```
.OPTION SIM_LA_MAXR=value
```

Default `1e15` ohms

Description

Use this option to specify the maximum resistance for linear matrix reduction. The *value* parameter specifies the maximum resistance preserved in the reduction. The linear matrix reduction process assumes that any resistor greater than *value* has an infinite resistance and drops the resistor after reduction completes. For additional information, see "Linear Acceleration" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA

.OPTION SIM_LA_MINC

Specifies the minimum capacitance for linear matrix reduction.

Syntax

```
.OPTION SIM_LA_MINC=value
```

Default `1e-16` farads

Description

Use this option to specify the minimum capacitance for linear matrix reduction.

The *value* parameter specifies the minimum capacitance preserved in the reduction.

The linear matrix reduction process lumps any capacitor smaller than *value* to ground after the reduction completes.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA

.OPTION SIM_LA_MINMODE

Reduces the number of nodes instead of the number of elements.

Syntax

```
.OPTION SIM_LA_MINMODE=ON | OFF
```

Default `OFF`

Description

Use this option to reduce the number of nodes instead of the number of elements.

- `ON`: Reduces the number of nodes
- `OFF`: Does not reduce the number of nodes.

For additional information, see "Linear Acceleration" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA

.OPTION SIM_LA_TIME

Specifies the minimum time for which accuracy must be preserved.

Syntax

`.OPTION SIM_LA_TIME=`*`value`*

Default `1` ns.

Example

For a circuit having a typical rise time of 1ns, either set the maximum frequency to 1 GHz, or set the minimum switching time to 1ns:

```
.OPTION SIM_LA_FREQ=1GHz
-or-
.OPTION SIM_LA_TIME=1ns
```

However, if spikes occur in 0.1ns, HSPICE RF does not accurately simulate them. To capture the behavior of the spikes, use:

```
.OPTION SIM_LA_FREQ=10GHz
-or-
.OPTION SIM_LA_TIME=0.1ns
```

Description

Use this option to specify the minimum time for which accuracy must be preserved.

The *value* parameter specifies the minimum switching time for which the PACT algorithm preserves accuracy.

Waveforms that occur more rapidly than the minimum switching time are not accurately represented.

This option is simply an alternative to `.OPTION SIM_LA_FREQ`. The default is equivalent to setting `SIM_LA_FREQ=1GHz`.

Note:

Higher frequencies (smaller times) increase accuracy, but only up to the minimum time step used in HSPICE RF.

For additional information, see “Linear Acceleration” in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA
.OPTION SIM_LA_FREQ

.OPTION SIM_LA_TOL

Specifies the error tolerance for the PACT algorithm.

Syntax

```
.OPTION SIM_LA_TOL=value
```

Default `0.05`ns.

Description

Use this option to specify the error tolerance for the PACT algorithm.

The *value* parameter must specify a real number between 0.0 and 1.0.

For additional information, see “Linear Acceleration” in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_LA

.OPTION SIM_ORDER

Controls the amount of Backward-Euler (BE) method to mix with the Trapezoidal (TRAP) method for hybrid integration.

Syntax

```
.OPTION SIM_ORDER=x
```

Default `1.9`

Description

Use this option to control the amount of Backward-Euler (BE) method to mix with the Trapezoidal (TRAP) method for hybrid integration.

The *x* parameter must specify a real number between 1.0 and 2.0.

- `SIM_ORDER=1.0` selects BE
- `SIM_ORDER=2.0` selects TRAP.

Note:

`.OPTION SIM_ORDER` has precedence over `.OPTION SIM_TRAP`.

A higher order is more accurate, especially with inductors (such as crystal oscillators), which need `SIM_ORDER=2.0`. A lower order has more damping.

This option affects time stepping when you set `.OPTION METHOD` to `TRAP` or `TRAPGEAR`.

Example

This example causes a mixture of 10% Gear-2 and 90% BE-trapezoidal hybrid integration. The BE-trapezoidal part is 10% BE.

```
.option sim_order=1.9
```

See Also

.OPTION METHOD
.OPTION SIM_TRAP

.OPTION SIM_OSC_DETECT_TOL

Specifies the tolerance for detecting numerical oscillations.

Syntax

```
.OPTION SIM_OSC_DETECT_TOL=value
```

Default `10^8`

Description

Use this option to specify the tolerance for detecting numerical oscillations. If HSPICE RF detects numerical oscillations, it inserts Backward-Euler (BE) steps. Smaller values of this tolerance result in fewer BE steps.

See Also

.OPTION METHOD

.OPTION SIM_POSTAT

Limits waveform output to nodes in the specified subcircuit instance.

Syntax

```
.OPTION SIM_POSTAT=instance
```

Description

Use this option to limit waveform output to nodes in the specified subcircuit instance only.

Example

The following example outputs X1.X4; see Figure 14.

```
.OPTION SIM_POSTAT=X1.X4
```

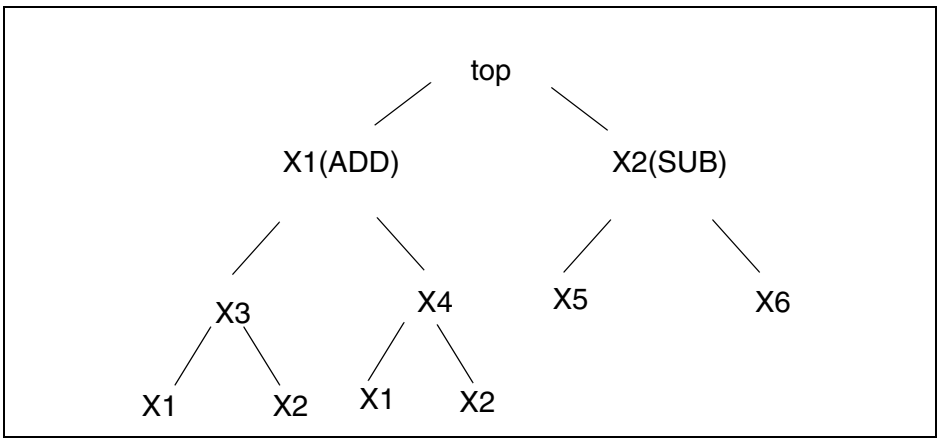

Figure 14 Node Hierarchy

See Also

.OPTION SIM_POSTSKIP
.OPTION SIM_POSTTOP

.OPTION SIM_POSTDOWN

Limits waveform output to nodes in the specified subcircuit instance and their children.

Syntax

```
.OPTION SIM_POSTDOWN=instance
```

Description

Use this option with `.OPTION SIM_POSTTOP` and it takes precedence over `.OPTION SIM_POSTSKIP`.

You can either use wildcards or set the option multiple times to specify multiple instances.

Example

The following example outputs top, X1, X1.X4, X1.X4.X1, X1.X4.X2, and X2. (See Figure 14 on page 562.)

```
.OPTION SIM_POSTTOP=2
.OPTION SIM_POSTDOWN=X1.X4
```

See Also

.OPTION SIM_POSTAT
.OPTION SIM_POSTSKIP
.OPTION SIM_POSTTOP

.OPTION SIM_POSTSCOPE

Specifies the signal types to probe from within a scope.

Syntax

```
.OPTION SIM_POSTSCOPE= net | port | all
```

Default `net`

Description

Use this option to specify the signal types to probe from within a scope.

- `net`: Outputs only nets in the scope
- `port`: Outputs both nets and ports
- `all`: Outputs nets, ports, and global variables.

See Also

.OPTION POST
.OPTION SIM_POSTSKIP
.OPTION SIM_POSTTOP

.OPTION SIM_POSTSKIP

Causes the SIM_POSTTOP option to skip *subckt_definition* instances.

Syntax

```
.OPTION SIM_POSTSKIP=subckt_definition
```

Description

Use this option to cause the SIM_POSTTOP option to skip any instances and their children that are defined by the *subckt_definition* parameter. To specify more than one subcircuit definition, issue this option once for each definition you want to skip.

Example

The following example outputs top, and skips X2. X1 because they are instances of the ADD subcircuit. (See Figure 14 on page 562.)

```
.OPTION SIM_POSTTOP=2
.OPTION SIM_POSTSKIP=ADD
```

See Also

.OPTION SIM_POSTTOP

.OPTION SIM_POSTTOP

Limits data written to your waveform file to data from only the top *n* level nodes.

Syntax

```
.OPTION SIM_POSTTOP=n
```

Default

3

Description

Limits the data written to your waveform file to data from only the top *n* level nodes.

This option outputs instances from up to n levels deep.

- SIM_POSTTOP=3: Outputs instances from 3 levels deep
- SIM_POSTTOP=1: Outputs instances from only the top-level signals.

If you specify the PROBE option without specifying a SIM_POSTTOP option HSPICE RF sets the SIM_POSTTOP=0. HSPICE RF outputs all levels if you do not specify either the PROBE option or a SIM_POSTTOP option.

Note:

You must specify the POST option to enable the waveform display interface.

Example 1

The following example outputs top, X1, and X2. (See Figure 14 on page 562.)

```
.OPTION SIM_POSTTOP=2
```

Example 2

The following example outputs top, X1, X2,and X4, X1and X2. (See Figure 14 on page 562.)

```
.OPTION SIM_POSTTOP=2
.OPTION SIM_POSTDOWN=X1.X4
```

See Also

.OPTION POST
.OPTION PROBE
.OPTION SIM_POSTSKIP

.OPTION SIM_POWER_ANALYSIS

Prints a list of signals matching the tolerance setting at a specified point in time.

Syntax

```
.OPTION SIM_POWER_ANALYSIS="time_point tol"
.OPTION SIM_POWER_ANALYSIS="bottom time_point tol"
```

Arguments

Parameter	Description
time_point	Time when HSPICE RF detects signals where the port current is larger than the tolerance value.
tol	Tolerance value for the signal defined in the .POWER command.
bottom	Signal at the lowest hierarchy level, also called a *leaf* subcircuit.

Description

Use this option to print a list of signals matching the tolerance (`tol`) setting at a specified point in time.

The first syntax produces a list of signals that consume more current than `tol` at `time point`, in this format:

```
*** time=< time point > threshold=< tol > ***
VDD=value
X13.VDD=value
X13.X1.VDD=value
X14.VDD=value
X14.X1.VDD=value
```

The second syntax produces the list of lowest-level signals, known as leaf subcircuits that consume more than `tol` at `time point`. The output is similar to this:

```
*** time=< time point > threshold=< tol > ***
X13.X1.VDD=value
X14.X1.VDD=value
```

For additional information, see "Power Analysis Output Format" in the *HSPICE User Guide: RF Analysis*.

Example

In this example, print the names of leaf subcircuits that use more than 100uA at 100ns into the simulation are printed.

```
.OPTION SIM_POWER_ANALYSIS=“bottom 100ns 100ua”
.POWER VDD
```

See Also

.POWER

.OPTION SIM_POWER_TOP

Controls the number of hierarchy levels on which power analysis is performed.

Syntax

```
.OPTION SIM_POWER_TOP=value
```

Description

Use this option to control the number of hierarchy levels on which power analysis is performed.

By default, power analysis is performed on the top levels of hierarchy.

Example

In the following example, HSPICE RF produces `.POWER` command results for top-level and first-level subcircuits (the subcircuit children of the top-level subcircuits).

```
.OPTION SIM_POWER_TOP=2
```

See Also

.POWER

.OPTION SIM_POWERDC_ACCURACY

Increases the accuracy of operating point calculations for POWERDC analysis.

Syntax

```
.OPTION SIM_POWERDC_ACCURACY=value
```

Description

Use this option to increase the accuracy of operating point calculations for POWERDC analysis.

A higher *value* results in greater accuracy, but more time to complete the calculation.

See Also

.POWERDC
.OPTION SIM_POWERDC_HSPICE

.OPTION SIM_POWERDC_HSPICE

Increases the accuracy of operating point calculations for POWERDC analysis.

Syntax

```
.OPTION SIM_POWERDC_HSPICE
```

Description

Use this option to increase the accuracy of operating point calculations for POWERDC analysis.

See Also

.POWERDC
.OPTION SIM_POWERDC_ACCURACY

.OPTION SIM_POWERPOST

Controls power analysis waveform dumping.

Syntax

```
.OPTION SIM_POWERPOST=ON|OFF
```

Default OFF

Description

Use this option to enable or disable power analysis waveform dumping.

See Also

.POWER

.OPTION SIM_POWERSTART

Specifies a default start time for measuring signals during simulation.

Syntax

```
.OPTION SIM_POWERSTART=time
```

Description

Use this option with a `.POWER` command to specify a default start time for measuring signals during simulation. This default time applies to all signals that do not have their own `FROM` measurement time. This option together with the `.OPTION SIM_POWERSTOP` control the power measurement scope for an entire simulation.

Example

In this example, the scope for simulating the x1.in signal is from 10 ps to 90 ps.

```
.OPTION SIM_POWERSTART=10ps
.OPTION SIM_POWERSTOP=90ps
.power x1.in
```

See Also

.OPTION SIM_POWERSTOP
.OPTION SIM_POWERSTART

.OPTION SIM_POWERSTOP

Specifies a default stop time for measuring signals during simulation.

Syntax

`.OPTION SIM_POWERSTOP=`*`time`*

Description

Use this option with a `.POWER` command to specify a default stop time for measuring signals during simulation. This default time applies to all signals that do not have their own TO measurement time. This option together with the `.OPTION SIM_POWERSTART` control the power measurement scope for an entire simulation.

See Also

.OPTION SIM_POWERSTART
.POWER

.OPTION SIM_SPEF

Runs simulation with SPEF expansion of all nets from one or more SPEF files.

Syntax

```
.OPTION SIM_SPEF="spec_filename"
```

Description

Use this option to run simulation with SPEF expansion of all nets from one or more SPEF files.

You can repeat this option to include more SPEF files.

For additional information, see "Post-Layout Back-Annotation" in the *HSPICE User Guide: RF Analysis*.

Example

In this example, the *senseamp.spf* SPEF file is back-annotated to the sense amp circuit.

```
.OPTION SIM_SPEF = "senseamp.spf"
```

See Also

.OPTION SIM_SPEF_ACTIVE
.OPTION SIM_SPEF_SCALEC
.OPTION SIM_SPEF_SCALER

.OPTION SIM_SPEF_ACTIVE

Runs simulation with selective SPEF expansion of active nets from one or more DSPF files.

Syntax

```
.OPTION SIM_SPEF_ACTIVE="active_node"
```

Description

Use this option to run simulation with selective SPEF expansion of active nets from one or more DSPF files.

HSPICE RF performs a preliminary verification run to determine the activity of the nodes and generates two ASCII files: *active_node*.rc and *active_node*.rcxt. These files save all active node information in both Star-RC and Star-RCXT formats.

If an *active_node* file is not generated from the preliminary run, no nets are expanded. Active nets are added to the file as they are identified in the subsequent transient simulation. A second simulation run using the same file and option causes only the nets listed in the *active_node* file to be expanded. It is possible that activity changes are due to timing changes caused by expansion of the active nets. In this case, additional nets are listed in the *active_node* file and a warning is issued.

By default, a node is considered active if the voltage varies by more than 0.1 V. You can use the `SIM_SPEF_VTOL` option to change this value.

HSPICE RF uses the *active_node* file and the DSPF file with the ideal netlist to expand only the active portions of the circuit. If a net is latent, then HSPICE RF does not expand it, which saves memory and CPU time.

For additional information, see “Selective Post-Layout Flow” in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_SPEF_VTOL

.OPTION SIM_SPEF_INSERROR

Skips unmatched instances.

Syntax

```
.OPTION SIM_SPEF_INSERROR=ON | OFF
```

Default `OFF`

Description

Use this option to skip unmatched instances.

- ON: Skips unmatched instances.
- OFF: Does not skip unmatched instances.

For more information, see “Additional Post-Layout Options” in the *HSPICE User Guide: RF Analysis*.

.OPTION SIM_SPEF_LUMPCAPS

Connects a lumped capacitor with a value equal to the net capacitance for instances missing in the hierarchical netlist.

Syntax

```
.OPTION SIM_SPEF_LUMPCAPS=ON | OFF
```

Default `ON`

Description

Use this option to connect a lumped capacitor with a value equal to the net capacitance for instances missing in the hierarchical netlist.

- `ON`: Adds lumped capacitance while ignoring other net contents.
- `OFF`: Uses net contents.

For additional information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

.OPTION SIM_SPEF_MAX_ITER

Specifies the maximum number of simulation runs for the second selective SPEF expansion pass.

Syntax

```
.OPTION SIM_SPEF_MAX_ITER=value
```

Default

`1`

Description

Use this option to specify the maximum number of simulation runs for the second selective SPEF expansion pass.

The *value* parameter specifies the number of iterations for the second simulation run.

Some of the latent nets might turn active after the first iteration of the second simulation run. In this case:

- Re simulate the netlist to ensure the accuracy of the post-layout simulation.
- Use this option to set the maximum number of iterations for the second run. If the active_node remains the same after the second simulation run, HSPICE RF ignores these options.

For additional information, see “Selective Post-Layout Flow” in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_SPEF_ACTIVE
.OPTION SIM_SPEF_VTOL

.OPTION SIM_SPEF_PARVALUE

Interprets triplet format *float*:*float*:*float* values in SPEF files as best:average:worst.

Syntax

```
.OPTION SIM_SPEF_PARVALUE=1|2|3
```

Default `2`

Description

Use this option to interpret triplet format *float*:*float*:*float* values in SPEF files as best:average:worst.

- `SIM_SPEF_PARVALUE = 1`: Use best.
- `SIM_SPEF_PARVALUE = 2`: Use average.
- `SIM_SPEF_PARVALUE = 3`: Use worst.

For further information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

.OPTION SIM_SPEF_RAIL

Controls whether power-net parasitics are back-annotated.

Syntax

```
.OPTION SIM_SPEF_RAIL=ON | OFF
```

Default OFF

Description

Use this option to control whether power-net parasitics are back-annotated.

- OFF: Do not back-annotate nets in a power rail.
- ON: Back-annotate nets in a power rail .

For further information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

.OPTION SIM_SPEF_SCALEC

Scales the capacitance values in a SPEF file for a standard SPEF expansion flow.

Syntax

```
.OPTION SIM_SPEF_SCALEC=scaleC
```

Description

Use this option to scale the capacitance values in a SPEF file for a standard SPEF expansion flow.

The *scaleC* parameter specifies the scale factor.

See “Additional Post-Layout Options” in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_SPEF_ACTIVE

.OPTION SIM_SPEF_SCALER

Scales the resistance values in a SPEF file for a standard SPEF expansion flow.

Syntax

```
.OPTION SIM_SPEF_SCALER=scaleR
```

Description

Use this option to scale the resistance values in a SPEF file for a standard SPEF expansion flow.

The *scaleR* parameter specifies the scale factor.

For more information, see "Additional Post-Layout Options" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_SPEF_ACTIVE

.OPTION SIM_SPEF_VTOL

Specifies multiple SPEF active thresholds.

Syntax

```
.OPTION SIM_SPEF_VTOL="value | scope1 scope2 ...
+ scopen"
```

Default `0.1V`

Description

Use this option to specify multiple SPEF active thresholds.

- The *value* parameter specifies the tolerance of voltage change. This value should be relatively small compared to the operating range of the circuit, or smaller than the supply voltage.
- The *scopen* parameter can be a subcircuit definition that uses a prefix of "@" or a subcircuit instance.

HSPICE RF performs a second simulation run by using the active_node file, the SPEF, and the hierarchical LVS ideal netlist to back-annotate only active portions of the circuit. If a net is latent, then HSPICE RF does not expand the net. This saves simulation runtime and memory.

By default, HSPICE RF performs only one iteration of the second simulation run. Use the `SIM_SPEF_MAX_ITER` option to change it.

For additional information, see "Selective Post-Layout Flow" in the *HSPICE User Guide: RF Analysis*.

See Also

.OPTION SIM_SPEF_ACTIVE
.OPTION SIM_SPEF_MAX_ITER

.OPTION SIM_TG_THETA

Controls the amount of second-order Gear method to mix with Trapezoidal integration for the hybrid TRAPGEAR method.

Syntax

```
.OPTION SIM_TG_THETA=x
```

Default

0

Description

Use this option to control the amount of second-order Gear (Gear-2) method to mix with Trapezoidal (TRAP) integration for the hybrid TRAPGEAR method.

The *value* parameter must specify a value between 0.0 and 1.0. The default is 0.1.

- `SIM_TG_THETA=0` selects TRAP without Gear-2.
- `SIM_TG_THETA=1` selects pure Gear-2.

See Also

.OPTION METHOD

.OPTION SIM_TRAP

Changes the default `SIM_TG_THETA=0` so that `METHOD=TRAPGEAR` acts like `METHOD=TRAP`.

Syntax

```
.OPTION SIM_TRAP=x
```

Default

`0.1`

Description

Use this option to change the default `SIM_TG_THETA=0` so that `METHOD=TRAPGEAR` acts like `METHOD=TRAP`.

The *x* parameter must specify a value between 0.0 and 1.0.

See Also

.OPTION METHOD
.OPTION SIM_TG_THETA

.OPTION SLOPETOL

Specifies the minimum value for breakpoint table entries in a piecewise linear (PWL) analysis.

Syntax

```
.OPTION SLOPETOL=x
```

Default `0.75`

Description

Use this option to specify the minimum value for breakpoint table entries in a piecewise linear (PWL) analysis. If the difference in the slopes of two consecutive PWL segments is less than the `SLOPETOL` value, HSPICE RF ignores the breakpoint for the point between the segments. Min value: 0; Max value: 2.

.OPTION SNACCURACY

Sets and modifies the size of timesteps.

Syntax

`.OPTION SNACCURACY=`*`integer`*

Default `10`

Description

Use this option to set and modify the size of timesteps. Larger values of `snaccuracy` result in a more accurate solution but might require more time points. Because Shooting-Newton must store derivative information at every time point, the memory requirements might be significant if the number of time points is very large.

The maximum integer value is 50.

For additional information, see SN Steady-State Time Domain Analysis in the *HSPICE User Guide: RF Analysis.*

See Also

.OPTION SIM_ACCURACY

.OPTION SNMAXITER

.OPTION SNMAXITER

Sets the maximum number of iterations for a Shooting Newton analysis.

Syntax

```
.OPTION SNMAXITER=integer
```

Default

`40`

Description

Use this option to limit the number of SN iterations. For more information, see Steady-State Shooting Newton Analysis in the *HSPICE User Guide: RF Analysis*.

.OPTION SPMODEL

Disables the previous .OPTION VAMODEL.

Syntax

```
.OPTION SPMODEL [= name]
```

Description

Use this option to disable a previously issued VAMODEL option. In this option, the name is the cell name that uses a SPICE definition. Each SPMODEL option can take no more than one name. Multiple names need multiple SPMODEL options.

Example 1

```
.OPTION SPMODEL
```

This example disables the previous .OPTION VAMODEL but has no effect on the other VAMODEL options if they are specified for the individual cells. For example, if .OPTION VAMODEL=vco has been set, the vco cell uses the Verilog-A definition whenever it is available until .OPTION SPMODEL=vco disables it.

Example 2

```
.option spmodel=chargepump
```

This example disables the previous .OPTION VAMODEL=chargepump, which causes all instantiations of chargepump to now use the subcircuit definition again.

See Also

.OPTION VAMODEL

.OPTION STATFL

Controls whether HSPICE creates a *.st0* file.

Syntax

```
.OPTION STATFL=0|1
```

Default `0`

Description

Use this option to control whether HSPICE creates a *.st0* file.

- `STATFL=0` Outputs a *.st0* file.
- `STATFL=1` Suppresses the *.st0* file.

.OPTION SYMB

Uses a symbolic operating point algorithm to get initial guesses before calculating operating points.

Syntax

```
.OPTION SYMB=0|1
```

Default 0

Description

Use this option to calculate the operating point. When SYMB is set to 1, HSPICE operates with a symbolic operating point algorithm to get initial guesses before calculating operating points. SYMB assumes the circuit is digital and assigns a low/high state to all nodes that set a reasonable initial voltage guess. This option improves DC convergence for oscillators, logic, and mixed-signal circuits.

.OPTION SYMB does not have any effect on the transient analysis if you set UIC in the .TRAN command.

.OPTION TIMERES

Sets the minimum separation between breakpoint values for the breakpoint table.

Syntax

```
.OPTION TIMERES=x
```

Default 1ps

Description

Use this option to set the minimum separation between breakpoint values for the breakpoint table. If two breakpoints are closer together in time than the TIMERES value, HSPICE enters only one of them in the breakpoint table.

.OPTION TMIFLAG

Invokes the TSMC Model Interface (TMI) flow.

Syntax

```
.OPTION TMIFLAG
```

Description

Use this option to invoke the TMI flow using proprietary TSMC model files and compiled libraries. Both the technology and API are jointly developed by Synopsys and TSMC. The TMI is a compact model with additional instance parameters and equations to support TSMC's extension of the standard BSIM4 model. Modeling API code is written in C and available in a compiled format for HSPICE and HSIM to link to during the simulation. TMI-required settings to invoke the flow and the location of a *.so* file are set by TSMC. The API also performs automatic platform selection on the *.so* file. Both HSPICE and HSIM provide the tool binaries and support the same *.so* file. To point to a TMI *.so* file location use the `.OPTION TMIPATH` command.

To map an instance name starting with "x" to "m" use `.OPTION MACMOD=2|3`.

Use the existing HSPICE and HSIM commands to run the simulation. (Contact Synopsys Technical Support for further information.)

See Also

.OPTION TMIPATH
.OPTION MACMOD

.OPTION TMIPATH

Points to a TMI **.so* (compiled library) file location.

Syntax

```
.OPTION TMIPATH='tmifilename_dir'
```

Description

Use this option to point to a TSCM Model Interface (TMI) **.so* file location. The path must be enclosed in single quotation marks.

Example

```
.option tmipath='tmi_v0d03_dir'
```

See Also

.OPTION TMIFLAG

.OPTION TNOM

Sets the reference temperature for the simulation.

Syntax

```
.OPTION TNOM=x
```

Default

25°C

Description

Use this option to set the reference temperature for the HSPICE RF simulation. At this temperature, component derating is zero.

Note:

The reference temperature defaults to the analysis temperature if you do not explicitly specify a reference temperature.

See Also

.TEMP (or) .TEMPERATURE

.OPTION TRANFORHB

Forces HB analysis to recognize or ignore specific V/I sources.

Syntax

```
.OPTION TRANFORHB=x
```

Description

This option forces HB analysis to recognize or ignore specific V/I sources.

- `TRANFORHB=1`: Forces HB analysis to recognize V/I sources that include SIN, PULSE, VMRF, and PWL transient descriptions, and to use them in analysis. However, if the source also has an HB description, analysis uses the HB description instead.
- `TRANFORHB=0`: Forces HB to ignore transient descriptions of V/I sources and to use only HB descriptions.

To override this option, specify `TRANFORHB` in the source description.

See Also

.HB

.OPTION TRTOL

Estimates the amount of error introduced when the timestep algorithm truncates the Taylor series expansion.

Syntax

```
.OPTION TRTOL=x
```

Default 7.00

Description

Use this option timestep algorithm for local truncation error (LVLTIM=2). HSPICE multiplies TRTOL by the internal timestep, which is generated by the timestep algorithm for the local truncation error. TRTOL reduces simulation time and maintains accuracy. It estimates the amount of error introduced when the algorithm truncates the Taylor series expansion. This error reflects the minimum timestep to reduce simulation time and maintain accuracy.

The range of TRTOL is 0.01 to 100; typical values are 1 to 10. If you set TRTOL to 1 (the minimum value), HSPICE uses a very small timestep. As you increase the TRTOL setting, the timestep size increases.

See Also

.OPTION LVLTIM

.OPTION UNWRAP

Displays phase results for AC analysis in unwrapped form.

Syntax

```
.OPTION UNWRAP=0|1
```

Description

Use this option to display phase results for AC analysis in unwrapped form (with a continuous phase plot).HSPICE uses these results to accurately calculate group delay. HSPICE also uses unwrapped phase results to compute group delay, even if you do not set `UNWRAP`. By default, HSPICE calculates the unwrapped phase first and then converts it to wrapped phase.

The convention is to normalize the phase output from -180 degrees to +180 degrees. A phase of -181 degrees is the same as a phase of +179 degrees.

Below is an example to illustrate how HSPICE wraps the phase.

Example

Default Method (Without)

```
Freq              Phase
3.16228k  -->   -167.7243
3.98107k  -->    178.7844
```

If you use .OPTION UNWRAP = 1

```
3.16228k  -->    -167.7243
3.98107k  -->    -181.2156
```

If the phase value goes beyond -180, then it wraps to a positive value. At the frequency 3.98107kHz the actual value is -181.2156, but by default, it is wrapped to +178.7844.

HSPICE does the following calculation to wrap the phase:

```
 -181.2156
 +180.0000
----------
   -1.2156
 +180.0000
   -1.2156
----------
  178.7844
```

.OPTION VAMODEL

Specifies that *name* is the cell name that uses a Verilog-A definition rather than the subcircuit definition when both exist (for use in HSPICE with Verilog-A).

Syntax

```
.OPTION VAMODEL [=name]
```

Default 0

Description

Use this option to specify that *name* is the cell name that uses a Verilog-A definition rather than the subcircuit definition when both exist. Each VAMODEL option can take no more than one name. Multiple names need multiple VAMODEL options.

If a name is not provided for the VAMODEL option, HSPICE uses the Verilog-A definition whenever it is available. The VAMODEL option works on cell-based instances only. Instance-based overriding is not allowed.

Example 1

The following example specifies a Verilog-A definition for all instantiations of the cell vco.

```
.option vamodel=vco
```

Example 2

The following example specifies a Verilog-A definition for all instantiations of the vco and chargepump cells.

```
.option vamodel=vco vamodel=chargepump
```

Example 3

The following example instructs HSPICE to always use the Verilog-A definition whenever it is available.

```
.option vamodel
```

.OPTION VERIFY

Duplicates the LIST option.

Syntax

```
.OPTION VERIFY=x
```

Default 0

Description

Use this option as an alias for the LIST option.

See Also

.OPTION LIST

.OPTION VFLOOR

Sets the minimum voltage to print in the output listing.

Syntax

```
.OPTION VFLOOR=x
```

Default

`500.00n`

Description

Use this option to set the minimum voltage to print in the output listing. All voltages lower than `VFLOOR` print as 0. Affects only the output listing; `VNTOL` (`ABSV`) sets the minimum voltage to use in a simulation.

See Also

.OPTION ABSV
.OPTION VNTOL

.OPTION VNTOL

Duplicates the ABSV option.

Syntax

```
.OPTION VNTOL=x
```

Default 50uV

Description

Use this option as an alias for the ABSV option. Default value: 5e-05; Min value: 0; Max value: 10.

See Also

.OPTION ABSV

.OPTION WACC

Activates the dynamic step control algorithm for a W element transient analysis.

Syntax

```
.OPTION WACC=x
```

Default 0

Description

Use this option to activate the dynamic step control algorithm for a W-element transient analysis. `WACC` is a non-negative real value that can be set between 0.0 and 10.0.

When WACC is zero, the conventional static time step control method is used. Larger WACC values result in less restriction in time point intervals (therefore faster simulation), while smaller values result in denser time points with higher accuracy.

Since the 2006.09 release, positive WACC is selected by default to activate the dynamic time step control. HSPICE automatically finds the optimum WACC value based on the netlist properties such as transmission line system delay, risetime, and transient command configurations. Since the W-elements in the netlist might have different properties, each has its own WACC values. If a user -specified positive WACC value is found in the netlist, HSPICE uses the user-defined WACC value for all the W-elements in the netlist. If the user-specified WACC is larger than the automatic estimation, HSPICE outputs a warning message.

The following refers to HSPICE only: For cases containing IBIS, PKG, EBD, or ICM blocks, HSPICE turns WACC off automatically. If you want to use the dynamic time step control algorithm for IBIS-related cases, you must set it explicitly in the netlist. For example:

```
.option WACC      $ Make HSPICE use automatically generated WACC
                    value for each W element
```

or

```
.option WACC=value      $ Use this value for all the W elements
```

See Also

Using Dynamic Time-Step Control in the *HSPICE User Guide: Signal Integrity*.

.OPTION WARNLIMIT (or) .OPTION WARNLIM

Limits how many times certain warnings appear in the output listing.

Syntax

```
.OPTION WARNLIMIT=n
```

Default 1

Description

Use this option to limit how many times the same warning appears in the output listing. This reduces the output listing file size. The *n* parameter specifies the maximum number of warnings for each warning type.

This limit applies to the following warning messages:

- MOSFET has negative conductance.
- Node conductance is zero.
- Saturation current is too small.
- Inductance or capacitance is too large.

See Also

.OPTION NOWARN

.OPTION WDELAYOPT

Globally applies the DELAYOPT keyword to a W-element transient analysis.

Syntax

```
.OPTION WDELAYOPT=[0|1|2|3]
```

Default 0

Description

Use this option as a global option which applies to all W-elements in a netlist. .OPTION WDELAYOPT can be overridden by the DELAYOPT keyword for a specified W-element.

- In cases where WDELAYOPT is set in the .OPTION and the DELAYOPT keyword is not specially set for Wxxx, the WDELAYOPT keyword is auto-set for Wxxx.
- In cases where the DELAYOPT keyword is already set for Wxxx, .OPTION WDELAYOPT is overridden for the Wxxx.
- In cases where neither .OPTION WDELAYOPT nor the DELAYOPT keyword is set, the DELAYOPT keyword defaults to 0.

.OPTION WDELAYOPT helps construct a W-element transient (recursive convolution) model with a higher level of accuracy. By specifying this option, you can add the DELAYOPT keyword to the W-element instance line.

You can use DELAYOPT=0|1|2 to deactivate, activate, and automatically determine, respectively.

Use DELAYOPT=3 to achieve a level of accuracy up to a tens of GHz operation and involve harmonics up to THz order. With this option, line length limits are removed, which frees the simulation from segmenting and allows independence in the behavior of the RISETIME option setting. A setting of WDELAYOPT=3 automatically detects whether or not frequency-dependent phenomena need to be recorded, which makes it identical to the DELAYOPT=0 setting if it produces a high enough accuracy.

See Use DELAYOPT Keyword for Higher Frequency Ranges in the *HSPICE User Guide: Signal Integrity*

See Also

.OPTION WINCLUDEGDIMAG
.OPTION RISETIME (or) .OPTION RISETI

.OPTION WDF

Enables HSPICE to produce waveform files in WDF format.

Syntax

```
.OPTION WDF=0|1
```

Default `0`

Description

Use this option to enable HSPICE to produce waveform files in WDF format. The WDF (Waveform Data File) format is the former Sandwork (acquired by Synopsys in 2007) proprietary waveform storage format. The WDF format compresses analog and logic waveform data, and facilitates fast waveform access for large data files. The compression scheme can be lossy or lossless (default). For the 2008.03 release, HSPICE only supports lossless compression.

- `.option WDF=0`—Disables this option
- `.option WDF` or `.option WDF=1`—Enables HSPICE to produce the waveform file in WDF format

For the WDF waveform file, HSPICE automatically appends *_wdf* into the output file root name to specify that it is in WDF format. The file names appear as: **_wdf.tr#, *_wdf.sw#, or *_wdf.ac#*.

For example, the WDF waveform output file will be named: *design_wdf.tr0*.

The WDF format is not available to HSPICE RF at this time.

.OPTION WINCLUDEGDIMAG

Globally activates the complex dielectric loss model in W-element analysis.

Syntax

```
.OPTION WINCLUDEGIMAG=[YES|NO]
```

Default NO

Description

Use this option as a global option to activate the complex dielectric loss model for all W-elements a netlist by introducing an imaginary term of the skin effect to be considered. If WINCLUDEGDIMAG=YESand there is no wp input, the W-element regards the Gd matrix as the conventional model and then automatically extracts constants for the complex dielectric model.
The .OPTION WINCLUDEGIMAG operates with the .OPTION WDELAYOPT option.

- In cases where WINCLUDEGDIMAG is set in the .OPTION and the INCLUDEGDIMAG keyword is not specially set for Wxxx, the INCLUDEGDIMAG is auto-set for Wxxx.
- In cases where the INCLUDEGDIMAG keyword is already set for Wxxx, .OPTION WINCLUDEGDIMAG is overridden for the Wxxx.
- In cases where neither .OPTION WINCLUDEGDIMAG nor the INCLUDEGDIMAG keyword is set, the INCLUDEGDIMAG keyword defaults to NO.

For details about the INCLUDEGDIMAG keyword, see Fitting Procedure Triggered by INCLUDEGDIMAG Keyword in the *HSPICE USER GUIDE: Signal Integrity*.

See Also

.OPTION WDELAYOPT
.OPTION RISETIME (or) .OPTION RISETI

.OPTION WL

Reverses the order of the VSIZE MOS element.

Syntax

```
.OPTION WL=0|1
```

Default 0

Description

Use this option to reverse the order of the MOS element VSIZE. The default order is length-width; this option changes the order to width-length.

.OPTION WNFLAG

Controls whether bin is selected based on w or w/nf.

Syntax

```
.OPTION WNFLAG=[0|1]
```

Default `1`

Description

Use this option to control whether HSPICE selects the bin based on the total device width (`WNFLAG=0`) or based on the width of one finger of a multifingered device (`WNFLAG=1`).

For devices which are using a BSIM4 model, an element parameter `wnflag=[0|1]` can be set, with the same effect as the option, and this element parameter overrides then the option setting on an element basis.

Example

For All Levels:

```
.option wnflag
M1 out in vdd vdd pmos w=10u l=1u nf=5
```

For BSIM4 models only:

```
M1 out in vdd vdd pmos w=10u l=1u nf=5 wnflag=1
```

.OPTION XDTEMP

Defines how HSPICE interprets the DTEMP parameter.

Syntax

```
.OPTION XDTEMP=0|1
```

Default 0(user-defined-parameter)

Description

Use this option to define how HSPICE interprets the DTEMP parameter, where *value* is either:

- 0: Indicates a user-defined parameter, or
- 1: Indicates a temperature difference parameter.

If you set .OPTION XDTEMP to 1, HSPICE adds the DTEMP value in the subcircuit call command to all elements within the subcircuit that use the DTEMP keyword syntax. The DTEMP parameter is cumulative throughout the design hierarchy.

Example

```
.OPTION XDTEMP
X1 2 0 SUB1 DTEMP=2
.SUBCKT SUB1 A B
R1 A B 1K DTEMP=3
C1 A B 1P
X2 A B sub2 DTEMP=4
.ENDS
.SUBCKT SUB2 A B
R2 A B 1K
.ENDS
```

In this example:

X1 sets a temperature difference (2 degrees Celsius) between the elements within the subcircuit SUB1.

X2 (a subcircuit instance of X1) sets a temperature difference by the DTEMP value of both X1 and X2 (2+4=6 degrees Celsius) between the elements within the SUB2 subcircuit. The DTEMP value of each element in this example is:

```
Elements DTEMP Value (Celsius)
X1 2
X1.R1 2+3 =5
X1.C1 2
X2 2+4=6
X2.R2 6
```

.OPTION (X0R,X0I)

The first of three complex starting-trial points in the Muller algorithm used in Pole/Zero analysis.

Syntax

```
.OPTION (X0R,X0I)= x,x
```

Default `X0R=-1.23456e6  X0I=0.0`

Description

Use this option in Pole/Zero analysis if you need to change scale factors and modify the initial Muller points, (X0R, X0I), (X1R, X1I) and (X2R, X2I). HSPICE multiplies these initial points, and FMAX, by FSCAL.

Scale factors must satisfy the following relations:

$$GSCAL = CSCAL \cdot FSCAL$$

$$GSCAL = \frac{1}{LSCAL \cdot FSCAL}$$

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION ITLPZ
.OPTION LSCAL
.OPTION PZABS
.OPTION PZTOL
.PZ

.OPTION (X1R,X1I)

The second of three complex starting-trial points in the Muller algorithm used in Pole/Zero analysis.

Syntax

```
.OPTION (X1R,X1I)= x,x
```

Default `X1R=1.23456e5  X1I=0.0`

Description

Use this option in Pole/Zero analysis if you need to change scale factors and modify the initial Muller points, (X0R, X0I), (X1R, X1I) and (X2R, X2I). HSPICE multiplies these initial points, and FMAX, by FSCAL.

Scale factors must satisfy the following relations:

$$GSCAL = CSCAL \cdot FSCAL$$

$$GSCAL = \frac{1}{LSCAL \cdot FSCAL}$$

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION ITLPZ
.OPTION LSCAL
.OPTION PZABS
.OPTION PZTOL
.PZ

.OPTION (X2R,X21)

The third of three complex starting-trial points in the Muller algorithm used in Pole/Zero analysis.

Syntax

```
.OPTION (X2R,X2I)= x,x
```

Default `X2R=+1.23456e6 X2I=0.0`

Description

Use this option in Pole/Zero analysis if you need to change scale factors and modify the initial Muller points, (X0R, X0I), (X1R, X1I) and (X2R, X2I). HSPICE multiplies these initial points, and FMAX, by FSCAL.

Scale factors must satisfy the following relations:

$$GSCAL = CSCAL \cdot FSCAL$$

$$GSCAL = \frac{1}{LSCAL \cdot FSCAL}$$

See Also

.OPTION CSCAL
.OPTION FMAX
.OPTION FSCAL
.OPTION GSCAL
.OPTION ITLPZ
.OPTION LSCAL
.OPTION PZABS
.OPTION PZTOL
.PZ

.VARIATION Block Control Options

The following options can be applied when doing .VARIATION analysis. Note that no leading period is allowed with variation control options:

- Option Normal_Limit=Value
  Limits the range of the Normal distributions. The default value is 4, that is, numbers in the range +/- 4 σ are generated. The range allowed is 0.1 to 6.0.

  This option allows a foundry to limit the perturbations to parameter ranges where a model is still valid.

- Option Ignore_Variation_Block=Yes
  Ignores the Variation Block and executes earlier style variations (traditional Monte Carlo analysis). By default, the contents of the variation block are executed and other definitions (AGAUSS, GAUSS, AUNIF, UNIF, LOT, and DEV) are ignored. Previous methods of specifying variations on parameters and models are not compatible with the Variation Block. By default, the contents of the Variation Block are used and all other specifications are ignored. Thus no changes are required in existing netlists other than adding the Variation Block.

- Option Ignore_Local_Variation=Yes
  Excludes effects of local variations in simulation. Default is No.

- Option Ignore_Global_Variation=Yes
  Excludes effects of global variations in simulation. Default is No.

- Option Ignore_Spatial_Variation=Yes
  Excludes effects of spatial variations in simulation. Default is No.

- Option Ignore_Interconnect_Variation=Yes
  Excludes effects of interconnect variations in simulation. Default is No. (See Interconnect Variation in Star-RCXT with the HSPICE Flow.)

- Option Output_Sigma_Value=Value
  Use to specify the sigma value of the results of Monte Carlo, DCMatch, and ACMatch analyses. Default is 1, range is 1 to 10. Note that this option only changes the output listings and that the input sigma is not affected.

- `Option Vary_Only Subckts=SubcktList`
  Use either this option to limit variation to the specified subcircuits or the one below, but not both. Actual subcircuit names are specified here (not the hierarchical names).
- `Option Do_Not_Vary Subckts=SubcktList`
  Excludes variation on the specified subcircuits. Use either this option to limit variation to the specified subcircuits or the one above, *but not both*. Actual subcircuit names are specified here (not the hierarchical names).

Monte Carlo-Specific Options Using the Variation Block

Options for the deprecated Monte Carlo style are ignored when simulations based on the Variation Block are executed.

The following Monte Carlo-specific options can be specified in the first section of the Variation Block:

- `Option Random_Generator = [Default | MSG]`

  Specifies the random number generator used in Variation Block based Monte Carlo analysis. `Random_Generator=MSG` invokes the generator from previous releases. `Random_Generator=Default` uses a longer cycle generator.
- `Option Stream =[x | Random | Default]`

  Specifies an integer stream number for random number generator (only for Variation Block). The minimum value of x is 1, the maximum value of x is 20; If `Stream=Random`, HSPICE creates a random stream number between 1 and 20 according to the system clock, and prints it in the *.lis* file for the user to debug. If `Stream=Default`, then it is equivalent to `Stream=1`.
- Option `Normal_Limit=Value`

  Limits the range for the numbers generated by the random number generator for Normal distributions. The default value is 4, i.e., numbers in the range +/- 4 σ are generated. The range allowed is 0.1 to 6.0.
- `Option Output_Sigma_Value=Value`

  Specifies the sigma value of the results. Default is 1, range is 1 to 10. Note that the input sigma is not affected.

- `Option Print_Only Subckts=SubcktList`
  Use either this option to limit output in the *.mc#* file to the specified subcircuits or the one below, *but not both*. Actual subcircuit names are specified here (not the hierarchical names).
- `Option Do_Not_Print Subckts=SubcktList`

  Use either this option to exclude output from the specified subcircuits to the *.mc#* file or the one above, *but not both*. Actual subcircuit names are specified here (not the hierarchical names).

See Also

Analyzing Variability and Using the Variation Block
Monte Carlo Analysis Using the Variation Block Flow

.DESIGN_EXPLORATION Block Control Options

The following options can be applied when doing `.DESIGN_EXPLORATION` analysis. Note that no leading period is allowed with variation control options:

- `Option Explore_only Subckts= SubcktList`
  This command is executed hierarchically—the specified subcircuits and all instantiated subcircuits and elements underneath are affected. Thus, if an inverter with name `INV1` is placed in a digital control block called `DIGITAL` and in an analog block `ANALOG`, and `Option Explore_only Subckts = ANALOG`, then the perturbations only affect the `INV1` in the analog block. You must create a new inverter, `INV1analog`, with the new device sizes.
- `Option Do_not_explore Subckts= SubcktList`
  Excludes listed subcircuits.
- `Option Export=yes|no`
  If yes, exports extraction data and runs one simulation with the original netlist. If no (default), runs a simulation with Exploration data.
- `Option Exploration_method=external Block_name=Block_name`
  The `Block_name` is the same as the name specified in the `.DATA` block; HSPICE will sweep the row content with the *`EXCommand`*`explore`.
- `Option Ignore_exploration= yes|no`
  (Default=no) HSPICE ignores the content in the design_exploration block, when `Ignore_exploration=yes`.
- `Option Secondary_param= yes|no`
  (Default=no) If `Secondary_param=yes`, HSPICE exports the MOSFET secondary instance parameters to a **.mex* file (created when `option export=yes`), and also permits the secondary parameters to be imported as a column header in the `.DATA` block (`option export=no`).

See Also

.DESIGN_EXPLORATION
Exploration Block

4

Digital Vector File Commands

Contains an alphabetical listing of the commands you can use in a digital vector file.

You can use the following HSPICE/HSPICE RF commands in a digital vector file.

ENABLE	TDELAY	VIL
IO	TFALL	VNAME
ODELAY	TRISE	VOH
OUT or OUTZ	TRIZ	VOL
PERIOD	TSKIP	VREF
RADIX	TUNIT	VTH
SLOPE	VIH	

ENABLE

Specifies the controlling signal(s) for bidirectional signals.

Syntax

ENABLE *controlling_signalname* [*mask*]

Arguments

Argument	Description
controlling_signalname	Controlling signal for bidirectional signals. Must be an input signal with a radix of 1. The bidirectional signals become output when the controlling signal is at state 1 (or high). To reverse this default control logic, start the control signal name with a tilde (~).
mask	Defines the bidirectional signals to which ENABLE applies.

Description

Use this command to specify the controlling signal(s) for bidirectional signals. All bidirectional signals require an ENABLE command. If you specify more than one ENABLE command, the last command overrides the previous command and HSPICE issues a warning message:

```
[Warning]:[line 6] resetting enable signal to WENB for bit 'XYZ'
```

Example

```
radix 144
io ibb
vname a x[[3:0]] y[[3:0]]
enable a 0 F 0
enable ~a 0 0 F
```

In this example, the *x* and *y* signals are bidirectional as defined by the *b* in the *io* line.

- The first enable command indicates that *x* (as defined by the position of *F*) becomes output when the *a* signal is 1.
- The second enable specifies that the *y* bidirectional bus becomes output when the *a* signal is 0.

IDELAY

Defines an input delay time for bidirectional signals.

Syntax

IDELAY *delay_value* [*mask*]

Arguments

Argument	Description
delay_value	Time delay to apply to the signals.
mask	Signals to which the delay applies. If you do not provide a *mask* value, the delay value applies to all signals.

Description

Use this command to define an input delay time for bidirectional signals relative to the absolute time of each row in the Tabular Data section. HSPICE ignores IDELAY settings on output signals and issues a warning message.

You can specify more than one TDELAY, IDELAY, or ODELAY command.

- If you apply more than one TDELAY (IDELAY, ODELAY) command to a signal, the last command overrides the previous commands and HSPICE or HSPICE RF issues a warning.
- If you do not specify the signal delays in a TDELAY, IDELAY, or ODELAY command, HSPICE or HSPICE RF defaults to zero.

Example

```
RADIX 1 1 4 1234 11111111
IO i i o iiib iiiiiiii
VNAME V1 V2 VX[[3:0]] V4 V5[[1:0]] V6[[0:2]] V7[[0:3]]
+ V8 V9 V10 V11 V12 V13 V14 V15
TDELAY 1.0
TDELAY -1.2 0 1 F 0000 00000000
TDELAY 1.5 0 0 0 1370 00000000
IDELAY 2.0 0 0 0 000F 00000000
ODELAY 3.0 0 0 0 000F 00000000
```

This example does not specify the TUNIT command so HSPICE or HSPICE RF uses the default, ns, as the time unit for this example. The first TDELAY command indicates that all signals have the same delay time of 1.0ns.

Subsequent TDELAY, IDELAY, or ODELAY commands overrule the delay time of some signals.

- The delay time for the V2 and Vx signals is -1.2.
- The delay time for the V4, V5[0:1], and V6[0:2] signals is 1.5.
- The input delay time for the V7[0:3] signals is 2.0, and the output delay time is 3.0.

See Also

ODELAY
TDELAY
TUNIT

IO

Defines the type for each vector: input, bidirectional, output, or unused.

Syntax

```
IO I | O | B | U     [I | O | B | U ...]
```

Arguments

Argument	Description
i	Input that HSPICE uses to stimulate the circuit.
o	Expected output that HSPICE compares with the simulated outputs.
b	Bidirectional vector.
u	Unused vector that HSPICE ignores.

Description

Use this command to define the type for each vector. The line starts with the `IO` keyword followed by a string of i, b, o, or u definitions. These definitions indicate whether each corresponding vector is an input (i), bidirectional (b), output (o), or unused (u) vector.

- If you do not specify the `IO` command, HSPICE or HSPICE RF assumes that all signals are input signals.
- If you define more than one `IO` command, the last command overrides previous commands.

Example

```
io i i i bbbb iiiioouu
```

ODELAY

Defines an output delay time for bidirectional signals.

Syntax

```
ODELAY delay_value [mask]
```

Arguments

Argument	Description
delay_value	Time delay to apply to the signals.
mask	Signals to which the delay applies. If you do not provide a *mask* value, the delay value applies to all signals.

Description

Use this command to define an output delay time for bidirectional signals relative to the absolute time of each row in the Tabular Data section.

HSPICE ignores ODELAY settings on input signals and issues a warning message.

You can specify more than one TDELAY, IDELAY, or ODELAY command.

- If you apply more than one TDELAY (IDELAY, ODELAY) command to a signal, the last command overrides the previous commands and HSPICE issues a warning.
- If you do not specify the signal delays in a TDELAY, IDELAY, or ODELAY command, HSPICE defaults to zero.

Example

```
RADIX 1 1 4 1234 11111111
IO i i o iiib iiiiiiii
VNAME V1 V2 VX[[3:0]] V4 V5[[1:0]] V6[[0:2]] V7[[0:3]]
+ V8 V9 V10 V11 V12 V13 V14 V15
TDELAY 1.0
TDELAY -1.2 0 1 F 0000 00000000
TDELAY 1.5 0 0 0 1370 00000000
IDELAY 2.0 0 0 0 000F 00000000
ODELAY 3.0 0 0 0 000F 00000000
```

This example does not specify the TUNIT command so HSPICE or HSPICE RF uses the default, ns, as the time unit for this example. The first TDELAY command indicates that all signals have the same delay time of 1.0ns.

Subsequent TDELAY, IDELAY, or ODELAY commands overrule the delay time of some signals.

- The delay time for the V2 and Vx signals is -1.2.
- The delay time for the V4, V5[0:1], and V6[0:2] signals is 1.5.
- The input delay time for the V7[0:3] signals is 2.0 and the output delay time is 3.0.

See Also

IDELAY
TDELAY
TUNIT

OUT or OUTZ

Specifies output resistance for each signal for which the mask applies. OUT and OUTZ are equivalent.

Syntax

OUT *output_resistance* [*mask*]

Arguments

Argument	Description
output_resistance	Output resistance for an input signal. The default is 0.
mask	Signals to which the output resistance applies. If you do not provide a *mask* value, the output resistance value applies to all input signals.

Description

The OUT and OUTZ keywords are equivalent: use these commands to specify output resistance for each signal (for which the mask applies). OUT or OUTZ applies to input signals only.

- If you do not specify the output resistance of a signal in an OUT (or OUTZ) command, HSPICE uses the default (zero).
- If you specify more than one OUT (or OUTZ) command for a signal, the last command overrides the previous commands and HSPICE issues a warning message.

The OUT (or OUTZ) commands have no effect on the expected output signals.

Example

```
OUT 15.1
OUT 150 1 1 1 0000 00000000
OUTZ 50.5 0 0 0 137F 00000000
```

The first OUT command in this example creates a 15.1 ohm resistor to place in series with all vector inputs. The next OUT command sets the resistance to 150 ohms for vectors 1 to 3. The OUTZ command changes the resistance to 50.5 ohms for vectors 4 through 7.

PERIOD

Defines the time interval for the Tabular Data section.

Syntax

`PERIOD` *`time_interval`*

Arguments

Argument	Description
time_interval	Time interval for the Tabular Data.

Description

Use this command to define the time interval for the Tabular Data section. You do not need to specify the absolute time at every time point. If you use a `PERIOD` command without the `TSKIP` command, the Tabular Data section contains only signal values, not absolute times. The `TUNIT` command defines the time unit of the `PERIOD`.

Example

```
radix 1111 1111
period 10
1000 1000
1100 1100
1010 1001
```

- The first row of the tabular data (1000 1000) is at time 0ns.
- The second row (1100 1100) is at 10ns.
- The third row (1010 1001) is at 20ns.

See Also

TSKIP
TUNIT

RADIX

Specifies the number of bits associated with each vector.

Syntax

```
RADIX number_of_bits [number_of_bits...]
```

Arguments

Argument	Description
number_of_bits	Specifies the number of bits in one vector in the digital vector file. You must include a separate *number_of_bits* argument in the RADIX command for each vector listed in the file.

Table 2 Valid Values for the RADIX command

# bits	Radix	Number System	Valid Digits
1	2	Binary	0, 1
2	4	–	0 – 3
3	8	Octal	0 – 7
4	16	Hexadecimal	0 – F

Description

Use this command to specify the number of bits associated with each vector. Valid values for the number of bits range from 1 to 4.

A digital vector file must contain only one `RADIX` command and it must be the first non-comment line in the file.

Example

```
; start of Vector Pattern Definition section
RADIX 1 1 4 1234 1111 1111
VNAME A B C[[3:0]] I9 I[[8:7]] I[[6:4]] I[[3:0]] O7 O6 O5 O4
+ O3 O2 O1 O0
IO I I I IIII OOOO OOOO
```

This example illustrates two 1-bit signals followed by a 4-bit signal, followed by one each 1-bit, 2-bit, 3-bit, and 4-bit signals, and finally eight 1-bit signals.

SLOPE

Specifies the rise/fall time for the input signal.

Syntax

SLOPE [*input_rise_time* | *input_fall_time*] [*mask*]

Arguments

Argument	Description
input_rise_time	Rise time of the input signal.
input_fall_time	Fall time of the input signal.
mask	Name of a signal to which the SLOPE command applies. If you do not specify a *mask* value, the SLOPE command applies to all signals.

Description

Use this command to specify the rise/fall time for the input signal. Use the TUNIT command to define the time unit for this command.

- If you do not specify the SLOPE command, the default slope value is 0.1 ns.
- If you specify more than one SLOPE command, the last command overrides the previous commands and HSPICE or HSPICE RF issues a warning message.

The SLOPE command has no effect on the expected output signals. You can specify the optional TRISE and TFALL commands to overrule the rise time and fall time of a signal.

Example 1

In the following example, the rising and falling times of all signals are 1.2 ns.

```
SLOPE 1.2
```

Example 2

In the following example, the rising/falling time is 1.1 ns for the first, second, sixth, and seventh signals.

```
SLOPE 1.1 1100 0110
```

See Also

TFALL
TRISE
TUNIT

TDELAY

Defines the delay time for both input and output signals in the Tabular Data section.

Syntax

TDELAY *delay_value* [*mask*]

Arguments

Argument	Description
delay_value	Time delay to apply to the signals.
mask	Signals to which the delay applies. If you do not provide a *mask* value, the delay value applies to all signals.

Description

Use this command to define the delay time of both input and output signals relative to the absolute time of each row in the Tabular Data section.

You can specify more than one TDELAY, IDELAY, or ODELAY command.

- If you apply more than one TDELAY (IDELAY, ODELAY) command to a signal, the last command overrides the previous commands and HSPICE or HSPICE RF issues a warning.
- If you do not specify the signal delays in a TDELAY, IDELAY, or ODELAY command, HSPICE or HSPICE RF defaults to zero.

Example

```
RADIX 1 1 4 1234 11111111
IO i i o iiib iiiiiiii
VNAME V1 V2 VX[[3:0]] V4 V5[[1:0]] V6[[0:2]] V7[[0:3]]
+ V8 V9 V10 V11 V12 V13 V14 V15
TDELAY 1.0
TDELAY -1.2 0 1 F 0000 00000000
TDELAY 1.5 0 0 0 1370 00000000
IDELAY 2.0 0 0 0 000F 00000000
ODELAY 3.0 0 0 0 000F 00000000
```

This example does not specify the TUNIT command so HSPICE or HSPICE RF uses the default, ns, as the time unit for this example. The first TDELAY command indicates that all signals have the same delay time of 1.0ns.

Subsequent TDELAY, IDELAY, or ODELAY commands overrule the delay time of some signals.

- The delay time for the V2 and Vx signals is -1.2.
- The delay time for the V4, V5[0:1], and V6[0:2] signals is 1.5.
- The input delay time for the V7[0:3] signals is 2.0, and the output delay time is 3.0.

See Also

IDELAY
ODELAY
TUNIT

TFALL

Specifies the fall time of each input signal for which the mask applies.

Syntax

TFALL *input_fall_time* [*mask*]

Arguments

Argument	Description
input_fall_time	Fall time of the input signal.
mask	Name of a signal to which the TFALL command applies. If you do not specify a *mask* value, the TFALL command applies to all input signals.

Description

Use this command to specify the fall time of each input signal for which the mask applies. The TUNIT command defines the time unit of TFALL.

- If you do not use any TFALL command to specify the fall time of the signals, HSPICE or HSPICE RF uses the value defined in the *slope* command.
- If you apply more than one TFALL command to a signal, the last command overrides the previous commands and HSPICE or HSPICE RF issues a warning message.

TFALL commands have no effect on the expected output signals.

Example

In the following example, the TFALL command assigns a fall time of 0.5 time units to all vectors.

```
TFALL 0.5
```

In the following example, the TFALL command assigns a fall time of 0.3 time units overriding the older setting of 0.5 to vectors 2, 3, and 4 to 7.

```
TFALL 0.3 0 1 1 137F 00000000
```

In the following example, the TFALL command assigns a fall time of 0.9 time units to vectors 8 through 11.

```
TFALL 0.9 0 0 0 0000 11110000
```

See Also

TRISE
TUNIT

TRISE

Specifies the rise time of each input signal for which the mask applies.

Syntax

```
TRISE input_rise_time [mask]
```

Arguments

Argument	Description
input_rise_time	Rise time of the input signal.
mask	Name of a signal to which the TRISE command applies. If you do not specify a *mask* value, the TRISE command applies to all input signals.

Description

Use this command to specify the rise time of each input signal for which the mask applies. The `TUNIT` command defines the time unit of `TRISE`.

- If you do not use any `TRISE` command to specify the rising time of the signals, HSPICE or HSPICE RF uses the value defined in the *slope* command.
- If you apply more than one `TRISE` command to a signal, the last command overrides the previous commands and HSPICE or HSPICE RF issues a warning message.

`TRISE` commands have no effect on the expected output signals.

Example 1

```
TRISE 0.3
```

In this example, the `TRISE` command assigns a rise time of 0.3 time units to all vectors.

Example 2

```
TRISE 0.5 0 1 1 137F 00000000
```

In this example, the `TRISE` command assigns a rise time of 0.5 time units overriding the older setting of 0.3 in at least some of the bits in vectors 2, 3, and 4 through 7.

Example 3

```
TRISE 0.8 0 0 0 0000 11110000
```

In this example, the `TRISE` command assigns a rise time of 0.8 time units to vectors 8 through 11.

See Also

TFALL
TUNIT

TRIZ

Specifies the output impedance when the signal for which the mask applies is in tristate.

Syntax

```
TRIZ output_impedance [mask]
```

Arguments

Argument	Description
output_impedance	Output impedance of the input signal.
mask	Name of a signal to which the TRIZ command applies. If you do not specify a *mask* value, the TRIZ command applies to all input signals.

Description

Use this command to specify the output impedance when the signal (for which the mask applies) is in *tristate*; `TRIZ` applies only to the input signals.

- If you do not specify the tristate impedance of a signal, in a `TRIZ` command, HSPICE or HSPICE RF assumes 1000M.
- If you apply more than one `TRIZ` command to a signal, the last command overrides the previous commands and HSPICE or HSPICE RF issues a warning.

`TRIZ` commands have no effect on the expected output signals.

Example

```
TRIZ 15.1Meg
TRIZ 150Meg 1 1 1 0000 00000000
TRIZ 50.5Meg 0 0 0 137F 00000000
```

- The first `TRIZ` command sets the high impedance resistance globally at 15.1 Mohms.
- The second `TRIZ` command increases the value to 150 Mohms for vectors 1 to 3.
- The last `TRIZ` command increases the value to 50.5 Mohms for vectors 4 through 7.

TSKIP

Causes HSPICE to ignore the absolute time field in the tabular data.

Syntax

```
TSKIP absolute_time tabular_data ...
```

Arguments

Argument	Description
absolute_time	Absolute time.
tabular_data	Data captured at *absolute_time*.

Description

Use this command to cause HSPICE to ignore the absolute time field in the tabular data. You can then keep, but ignore, the absolute time field for each row in the tabular data when you use the `.PERIOD` command.

You might do this, for example, if for testing reasons the absolute times are not perfectly periodic. Another reason might be that a path in the circuit does not meet timing, but you might still use it as part of a test bench. Initially, HSPICE writes to the vector file using absolute time. After you fix the circuit, you might want to use periodic data.

Example

```
radix 1111 1111
period 10
tskip
11.0 1000 1000
20.0 1100 1100
33.0 1010 1001
```

HSPICE or HSPICE RF ignores the absolute times 11.0, 20.0 and 33.0, but HSPICE does process the tabular data on the same lines as those absolute times.

See Also

PERIOD

TUNIT

Defines the time unit for PERIOD, TDELAY,IDELAY, ODELAY, SLOPE, TRISE, TFALL, and absolute time.

Syntax

```
TUNIT [fs|ps|ns|us|ms]
```

Arguments

Argument	Description
fs	femtosecond
ps	picosecond
ns	nanosecond (this is the default)
us	microsecond
ms	millisecond

Description

Use this command to define the time unit in the digital vector file for PERIOD, TDELAY, IDELAY, ODELAY, SLOPE, TRISE, TFALL, and absolute time.

- If you do not specify the `TUNIT` command, the default time unit value is ns.
- If you define more than one `TUNIT` command, the last command overrides the previous command.

Example

```
TUNIT ns
11.0 1000 1000
20.0 1100 1100
33.0 1010 1001
```

The `TUNIT` command in this example specifies that the absolute times in the Tabular Data section are 11.0ns, 20.0ns, and 33.0ns.

See Also

IDELAY
ODELAY
PERIOD
SLOPE

TDELAY
TFALL
TRISE

VIH

Specifies the logic-high voltage for each input signal to which the mask applies.

Syntax

```
VIH logic-high_voltage [mask]
```

Arguments

Argument	Description
logic-high_voltage	Logic-high voltage for an input signal. The default is 3.3.
mask	Name of a signal to which the VIH command applies. If you do not specify a *mask* value, the VIH command applies to all input signals.

Description

Use this command to specify the logic-high voltage for each input signal to which the mask applies.

- If you do not specify the logic high voltage of the signals in a `VIH` command, HSPICE assumes 3.3.
- If you use more than one `VIH` command for a signal, the last command overrides previous commands and HSPICE issues a warning.

`VIH` commands have no effect on the expected output signals.

Example

```
VIH 5.0
VIH 3.5 0 0 0 0000 11111111
```

- The first `VIH` command sets all input vectors to 5V when they are high.
- The last `VIH` command changes the logic-high voltage from 5V to 3.5V for the last eight vectors.

See Also

VIL
VOH
VOL
VTH

VIL

Specifies the logic-low voltage for each input signal to which the mask applies.

Syntax

```
VIL logic-low_voltage [mask]
```

Arguments

Argument	Description
logic-low_voltage	Logic-low voltage for an input signal. The default is 0.0.
mask	Name of a signal to which the VIL command applies. If you do not specify a *mask* value, the VIL command applies to all input signals.

Description

Use this command to specify the logic-low voltage for each input signal to which the mask applies.

- If you do not specify the logic-low voltage of the signals in a `VIL` command, HSPICE or HSPICE RF assumes 0.0.
- If you use more than one `VIL` command for a signal, the last command overrides previous commands and HSPICE issues a warning.

`VIL` commands have no effect on the expected output signals.

Example

```
VIL 0.0
VIL 0.5 0 0 0 0000 11111111
```

- The first `VIL` command sets the logic-low voltage to 0V for all vectors.
- The second `VIL` command changes the logic-low voltage to 0.5V for the last eight vectors.

See Also

VIH
VOH
VOL
VTH

VNAME

Defines the name of each vector.

Syntax

```
VNAME vector_name [[starting_index:ending_index]]
```

Arguments

Argument	Description
vector_name	Name of the vector, or range of vectors.
starting_index	First bit in a range of vector names.
ending_index	Last bit in a range of vector names. You can associate a single name with multiple bits (such as bus notation). The opening and closing brackets and the colon are required; they indicate that this is a range. The vector name must correlate with the number of bits available. You can nest the bus definition inside other grouping symbols, such as { }, ( ), [ ], and so on. The bus indices expand in the specified order

Description

Use this command to define the name of each vector. If you do not specify VNAME, HSPICE or HSPICE RF assigns a default name to each signal: V1, V2, V3, and so on. If you define more than one VNAME command, the last command overrides the previous command.

Example 1

```
RADIX 1 1 1 1 1 1 1 1 1 1 1 1
VNAME V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12
```

Example 2

```
VNAME a[[0:3]]
```

This example represents a0, a1, a2, and a3, in that order. HSPICE or HSPICE RF does not reverse the order to make a3 the first bit.

The bit order is MSB:LSB, which means most significant bit to least significant bit. For example, you can represent a 5-bit bus such as: {a4 a3 a2 a1 a0}, using this notation: a[[4:0]]. The high bit is a4, which represents 2^4. It is the largest value and therefore is the MSB.

Example 3

```
RADIX 2 4
VNAME VA[[0:1]] VB[[4:1]]
```

HSPICE or HSPICE RF generates voltage sources with the following names:

```
VA0 VA1 VB4 VB3 VB2 VB1
```

- *VA0* and *VB4* are the MSBs.
- *VA1* and *VB1* are the LSBs.

Example 4

```
VNAME VA[[0:1]] VB<[4:1]>
```

HSPICE or HSPICE RF generates voltage sources with the following names:

```
VA[0] VA[1] VB<4> VB<3> VB<2> VB<1>
```

Example 5

```
VNAME VA[[2:2]]
```

This example specifies a single bit of a bus. This range creates a voltage source named:

```
VA[2]
```

Example 6

```
RADIX 444444
VNAME A[[0:23]]
```

This example generates signals named A0, A1, A2, ... A23.

VOH

Specifies the logic-high threshold voltage for each output signal to which the mask applies.

Syntax

VOH *logic-high_threshold_voltage* [*mask*]

Arguments

Argument	Description
logic-high_threshold_voltage	Logic-high threshold voltage for an output vector. The default is 2.66.
mask	Name of a signal to which the VOH command applies. If you do not specify a *mask* value, the VOH command applies to all output signals.

Description

Use this command to specify the logic-high threshold voltage for each output signal to which the mask applies.

- If you do not specify the logic-high threshold voltage in a VOH command, HSPICE assumes 2.64.
- If you apply more than one VOH command to a signal, the last command overrides the previous commands and HSPICE issues a warning.

VOH commands have no effect on input signals.

Example

```
VOH 4.75
VOH 4.5 1 1 1 137F 00000000
VOH 3.5 0 0 0 0000 11111111
```

- The first line tries to set a logic-high threshold output voltage of 4.75V, but it is redundant.
- The second line changes the voltage level to 4.5V for the first seven vectors.
- The last line changes the last eight vectors to a 3.5V logic-high threshold output.

These second and third lines completely override the first VOH command.

If you do not define either VOH or VOL, HSPICE or HSPICE RF uses VTH (default or defined).

See Also

VIH
VIL
VOL
VTH

VOL

Specifies the logic-low threshold voltage for each output signal to which the mask applies.

Syntax

VOL *logic-low_threshold_voltage* [*mask*]

Arguments

Argument	Description
logic-low_voltage	Logic-low threshold voltage for an output vector. The default is 0.64.
mask	Name of a signal to which the VOL command applies. If you do not specify a *mask* value, the VOL command applies to all output signals.

Description

Use this command to specify the logic-low threshold voltage for each output signal to which the mask applies.

- If you do not specify the logic-low threshold voltage in a VOL command, HSPICE assumes 0.66.
- If you apply more than one VOL command to a signal, the last command overrides the previous commands and HSPICE issues a warning.

Example

```
VOL 0.0
VOL 0.2 0 0 0 137F 00000000
VOL 0.5 1 1 1 0000 00000000
```

- The first VOL command sets the logic-low threshold output to 0V.
- The second VOL command sets the output voltage to 0.2V for the fourth through seventh vectors.
- The last command increases the voltage further to 0.5V for the first three vectors.

These second and third lines completely override the first VOL command.

If you do not define either VOH or VOL, HSPICE or HSPICE RF uses VTH (default or defined).

See Also

VIH
VIL
VOH
VTH

VREF

Specifies the name of the reference voltage for each input vector to which the mask applies.

Syntax

```
VREF reference_voltage
```

Arguments

Argument	Description
reference_voltage	Reference voltage for each input vector. The default is 0.

Description

Use this command to specify the name of the reference voltage for each input vector to which the mask applies. Similar to the `TDELAY` command, the `VREF` command applies only to input signals.

- If you do not specify the reference voltage name of the signals in a `VREF` command, HSPICE assumes 0.
- If you apply more than one `VREF` command, the last command overrides the previous commands and HSPICE issues a warning.

`VREF` commands have no effect on the output signals.

Example

```
VNAME v1 v2 v3 v4 v5[[1:0]] v6[[2:0]] v7[[0:3]] v8 v9 v10
VREF 0
VREF 0 111 137F 000
VREF vss 0 0 0 0000 111
```

When HSPICE or HSPICE RF implements these commands into the netlist, the voltage source realizes *v1*:

```
v1 V1 0 pwl(......)
```

as well as *v2*, *v3*, *v4*, *v5*, *v6*, and *v7*.

However, *v8* is realized by

```
V8 V8 vss pwl(......)
```

v9 and *v10* use a syntax similar to v8.

See Also

TDELAY

VTH

Specifies the logic threshold voltage for each output signal to which the mask applies.

Syntax

VTH *logic-threshold_voltage*

Arguments

Argument	Description
logic-threshold_voltage	Logic-threshold voltage for an output vector. The default is 1.65.

Description

Use this command to specify the logic threshold voltage for each output signal to which the mask applies. It is similar to the TDELAY command. The threshold voltage determines the logic state of output signals for comparison with the expected output signals.

- If you do not specify the threshold voltage of the signals in a VTH command, HSPICE assumes 1.65.
- If you apply more than one VTH command to a signal, the last command overrides the previous commands and HSPICE or HSPICE RF issues a warning.

VTH commands have no effect on the input signals.

Example

```
VTH 1.75
VTH 2.5 1 1 1 137F 00000000
VTH 1.75 0 0 0 0000 11111111
```

- The first VTH command sets the logic threshold voltage at 1.75V.
- The next line changes that threshold to 2.5V for the first 7 vectors.
- The last line changes that threshold to 1.75V for the last 8 vectors.

All of these examples apply the same vector pattern and both output and input control commands, so the vectors are all bidirectional.

See Also

TDELAY
VIH
VIL
VOH
VOL

A

Obsolete Commands and Options

Describes the obsolete or rarely used HSPICE commands.

The following commands and options are included for completeness only. More efficient functionality and commands are available. The obsolete commands and options are:

- .GRAPH
- .MODEL Command for .GRAPH
- .NET
- .PLOT
- .WIDTH
- .OPTION ALT999 or ALT9999
- .OPTION BKPSIZ
- .OPTION CDS
- .OPTION CO
- .OPTION H9007
- .OPTION MEASSORT
- .OPTION MENTOR
- .OPTION MODSRH
- .OPTION PLIM
- .OPTION SDA
- .OPTION TRCON
- .OPTION ZUKEN

.GRAPH

Provides high-resolution plots of HSPICE simulation results.

Note:

This is an obsolete command. You can gain the same functionality by using the .PROBE command.

Syntax

```
.GRAPH antype <MODEL=mname> <unam1=> ov1,
+ <unam2=>ov2 ... <unamn=>ovn (plo,phi)
```

Arguments

Argument	Description
antype	Type of analysis for the specified plots (outputs). Analysis types are: DC, AC, TRAN, NOISE, or DISTO.
mname	Plot model name, referenced in the .GRAPH command. Use .GRAPH and its plot name to create high-resolution plots directly from HSPICE.
unam1...	You can define output names, which correspond to the ov1 ov2 ... output variables (*unam1 unam2 ...*), and use them as labels, instead of output variables for a high resolution graphic output.
ov1 ...	Output variables to print. Can be voltage, current, or element template variables from a different type of analysis. You can also use algebraic expressions as output variables, but you must define them inside the PAR( ) command.
plo, phi	Lower and upper plot limits. Set the plot limits only at the end of the .GRAPH command.

Example

```
.GRAPH DC cgb=lx18(m1) cgd=lx19(m1)
+ cgs=lx20(m1)
.GRAPH DC MODEL=plotbjt
+ model_ib=i2(q1)    meas_ib=par(ib)
+ model_ic=i1(q1)    meas_ic=par(ic)
+ model_beta=par('i1(q1)/i2(q1)')
+ meas_beta=par('par(ic)/par(ib)')(1e-10,1e-1)
.MODEL plotbjt PLOT MONO=1 YSCAL=2 XSCAL=2
+ XMIN=1e-8 XMAX=1e-1
```

Description

Use this command when you need high-resolution plots of HSPICE simulation results.

Each `.GRAPH` command creates a new *.gr#* file, where # ranges first from 0 to 9 and then from a to z. You can create up to 10000 graph files.

You can include wildcards in `.GRAPH` commands.

You cannot use `.GRAPH` commands in the Windows version of HSPICE or in HSPICE RF.

.MODEL Command for .GRAPH

For a description of how to use the `.MODEL` command with `.GRAPH`, see the .MODEL command in the *HSPICE Command Reference*.

Table 3 Model Parameters

Name (Alias)	Default	Description
MONO	0.0	Monotonic option. MONO=1 automatically resets the x-axis, if any change occurs in the x direction.
TIC	0.0	Shows tick marks.
FREQ	0.0	Plots symbol frequency. ▪ A value of 0 does not generate plot symbols. ▪ A value of *n* generates a plot symbol every *n* points. This is not the same as the FREQ keyword in element commands (see the Modeling Filters and Networks chapter in the *HSPICE User Guide: Simulation and Analysis*).
XGRID, YGRID	0.0	Set these values to 1.0, to turn on the axis grid lines.
XMIN, *XMA*X	0.0	▪ If XMIN is not equal to XMAX, then XMIN and XMAX determine the x-axis plot limits. ▪ If XMIN equals XMAX, or if you do not set XMIN and XMAX, then HSPICE automatically sets the plot limits. These limits apply to the actual x-axis variable value, regardless of the XSCAL type.

Table 3 Model Parameters (Continued)

Name (Alias)	Default	Description
XSCAL	1.0	Scale for the x-axis. Two common axis scales are: Linear(LIN) (XSCAL=1) Logarithm(LOG) (XSCAL=2)
YMIN, *YMAX*	0.0	▪ If YMIN is not equal to YMAX, then YMIN and YMAX determine the y-axis plot limits. The y-axis limits in the .GRAPH command overrides YMIN and YMAX in the model. ▪ If you do not specify plot limits, HSPICE sets the plot limits. These limits apply to the actual y-axis variable value, regardless of the YSCAL type.
YSCAL	1.0	Scale for the y-axis. Two common axis scales are: Linear(LIN) (XSCAL=1) Logarithm(LOG) (XSCAL=2)

.NET

Computes parameters for impedance, admittance, hybrid, and scattering matrixes.

Syntax

One-Port Network

```
.NET input <RIN=val>
.NET input <val>
```

Two-Port Network

```
.NET output input <ROUT=val> <RIN=val>
```

Arguments

Argument	Description
input	Name of the voltage or current source for AC input.
output	Output port. It can be: ▪ An output voltage, V(n1<,n2>). ▪ An output current, I (source), or I (element).
RIN	Input or source resistance. RIN calculates output impedance, output admittance, and scattering parameters. The default RIN value is 1 ohm.
ROUT	Output or load resistance. ROUT calculates input impedance, admittance, and scattering parameters. The default is 1 ohm.

Example

One-Port Network

```
.NET   VINAC   RIN=50
.NET   IIN   RIN=50
```

Two-Port Network

```
.NET V(10,30)   VINAC   ROUT=75   RIN=50
.NET I(RX)   VINAC   ROUT=75   RIN=50
```

Description

You can the .NET command to compute parameters for:

- Z impedance matrix
- Y admittance matrix
- H hybrid matrix
- S scattering matrix

You can use the .NET command only in conjunction with the .AC command.

HSPICE also computes:

- Input impedance
- Output impedance
- Admittance

This analysis is part of AC small-signal analysis. To run network analysis, specify the frequency sweep for the `.AC` command.

.PLOT

Plots the output values of one or more variables in a selected HSPICE analysis as a low-resolution (ASCII) plot in the output listing file.

Note:

This is an obsolete command. You get the same functionality using the `.PRINT` command.

Syntax

```
.PLOT antype ov1 <(plo1,phi1)> <ov2> <(plo2,phi2)> ...>
```

Arguments

Argument	Description
antype	Type of analysis for the specified plots. Analysis types are: DC, AC, TRAN, NOISE, or DISTO.
ov1 ...	Output variables to plot: voltage, current, or element template variables (HSPICE only; HSPICE RF does not support element template output or .PLOT commands)from a DC, AC, TRAN, NOISE, or DISTO analysis. See the next sections for syntax.
plo1, phi1 ...	Lower and upper plot limits. The plot for each output variable uses the first set of plot limits after the output variable name. Set a new plot limit for each output variable after the first plot limit. For example to plot all output variables that use the same scale, specify one set of plot limits at the end of the .PLOT command. If you set the plot limits to (0,0) HSPICE automatically sets the plot limits.

Example 1

```
.PLOT DC V(4) V(5) V(1) PAR(`I1(Q1)/I2(Q1)')
.PLOT TRAN V(17,5) (2,5) I(VIN) V(17) (1,9)
.PLOT AC VM(5) VM(31,24) VDB(5) VP(5) INOISE
```

- In the first line, PAR plots the ratio of the collector current and the base current for the Q1 transistor.
- In the second line, the VDB output variable plots the AC analysis results (in decibels) for node 5.
- In the third line, the AC plot can include NOISE results and other variables that you specify.

Example 2

```
.PLOT AC ZIN YOUT(P) S11(DB) S12(M) Z11(R)
.PLOT DISTO HD2 HD3(R) SIM2
.PLOT TRAN V(5,3) V(4) (0,5) V(7) (0,10)
.PLOT DC V(1) V(2) (0,0) V(3) V(4) (0,5)
```

In the last line above, HSPICE sets the plot limits for V(1) and V(2), but you specify 0 and 5 volts as the plot limits for V(3) and V(4).

Description

Use this command to plot the output values of one or more variables in a selected HSPICE analysis. Each .PLOT command defines the contents of one plot, which can contain more than one output variable.

If more than one output variable appears on the same plot, HSPICE prints *and* plots the first variable specified. To print out more than one variable, include another .PLOT command. You can include wildcards in .PLOT commands.

.WIDTH

(Obsolete) Specifies the width of the low resolution (ASCII) plot in the listing file.

Syntax

```
.WIDTH OUT={80 |132}
```

Arguments

Argument	Description
OUT	Output print width.

Example

```
.WIDTH OUT=132 $ SPICE compatible style
.OPTION CO=132 $ preferred style
```

Description

Use this command to specify the width of the low resolution (ASCII) plot. Permissible values for OUT are 80 and 132. You can also use .OPTION CO to set the OUT value.

.OPTION ALT999 or ALT9999

Allows the .GRAPH command to create more output files when you run .ALTER simulations.

Syntax

```
.OPTION ALT999
.OPTION ALT9999
```

Description

Use this option to allow the .GRAPH command to create more output files when you run .ALTER simulations.

This option is now obsolete. HSPICE can now generate up to 10,000 unique files without using this option.

.OPTION BKPSIZ

Sets the size of the breakpoint table.

Syntax

```
.OPTION BKPSIZ=x
```

Default 5000

Description

Use this option to set the size of the breakpoint table. This is an obsolete option, provided only for backward-compatibility.

.OPTION CDS

Produces a Cadence WSF (ASCII format) post-analysis file for Opus™.

Syntax

```
.OPTION CDS=x
```

Description

Use this option to produce a Cadence WSF (ASCII format) post-analysis file for Opus™ when CDS=2. This option requires a specific license. The CDS option is the same as the SDA option.

.OPTION CO

(Obsolete) Sets column width for printouts.

Syntax

```
.OPTION CO=<column_width>
```

Arguments

Parameter	Description
column_width	The number of characters in a single line of output.

Example

```
* Narrow print-out (default)
.OPTION CO=80
* Wide print-out
.OPTION CO=132
```

Description

(Obsolete) Use this option to set the column width for printouts. The number of output variables that print on a single line of output is a function of the number of columns.

You can set up to 5 output variables per 80-column output, and up to 8 output variables per 132-column output with 12 characters per column. HSPICE automatically creates additional print commands and tables for all output variables beyond the number that the CO option specifies. The default is 78.

.OPTION H9007

Sets default values for general-control options to correspond to values for HSPICE H9007D.

Syntax

```
.OPTION H9007
```

Default 0

Description

Use this option to set default values for general-control options to correspond to values for HSPICE H9007D. If you set this option, HSPICE does not use the EXPLI model parameter.

.OPTION MEASSORT

Automatically sorts large numbers of .MEASURE commands. (This option is obsolete.)

Syntax

```
.OPTION MEASSORT=x
```

Default 0

Description

Note:

Starting in version 2003.09, this option is obsolete. Measure performance is now order-independent and HSPICE ignores this option.

In versions of HSPICE before 2003.09, to automatically sort large numbers of .MEASURE commands, you could use the .OPTION MEASSORT command.

- .OPTION MEASSORT=0 (default; did not sort .MEASURE commands).
- .OPTION MEASSORT=1 (internally sorted .MEASURE commands).

You needed to set this option to 1 only if you used a large number of .MEASURE commands, where you needed to list similar variables together (to reduce simulation time). For a small number of .MEASURE commands, turning on internal sorting sometimes slowed-down simulation while sorting, compared to not sorting first.

.OPTION MENTOR

Enables the Mentor MSPICE-compatible (ASCII) interface.

Syntax

```
.OPTION MENTOR=0|1|2
```

Default 0

Description

Use this option to enable the Mentor MSPICE-compatible (ASCII) interface. `MENTOR=2` enables that interface. This option requires a specific license.

.OPTION MODSRH

Made obsolete beginning in the 2008.03 release as it increases runtime and costs more memory. Controls whether HSPICE loads or references a model described in a `.MODEL` command, but not used in the netlist.

Syntax

```
.OPTION MODSRH=0|1
```

Default `0`

Description

Use this option to control whether HSPICE loads or references a model described in a `.MODEL` command, but is not used in the netlist.

This option parameter determines if HSPICE reads and loads every model card or all model bins that are present in netlists and model libraries during a simulation run. When this parameter is set to 0, all the model cards in the model libraries will be read into HSPICE even if there are certain models or bins that are not referenced by any elements of the netlists. If this option parameter is not assigned a numerical value or is set to 1, or it is not specified at all, then only those model cards or model bins that are referenced are read into the HSPICE executable for simulation.

Note:

The `.OPTION MODSRH` control must appear before the `.MODEL` definition.

- `MODSRH=0`: all models expanded even if the model described in a `.MODEL` command is not referenced. This was the default prior to Y-2006.03 and restored in A-2008.03.
- `MODSRH=1`: only referenced models are expanded. This option shortens simulation runtime when the netlist references many models, but no element in the netlist calls those models. This option increased read-in time. This was the default after Y-2006.03 until 2008.03.

Example

In this example, the input file automatically searches t6.inc for the nch model, but it is not loaded.

```
example.sp:
.option post modsrh=1
xi1 net8 b c t6
xi0 a b net8 t6
v1 a 0 pulse 3.3 0.0 10E-6 1E-9 1E-9
+ 25E-6 50E-6
v2 b 0 2
v3 c 0 3
.model nch nmos level=49 version=3.2
.end
```

See Also

.MODEL

.OPTION PLIM

Specifies plot size limits for current and voltage plots.

Syntax

```
.OPTION PLIM
```

Default 0

Description

Use this option to specify plot size limits for current and voltage plots:

- Finds a common plot limit and plots all variables on one graph at the same scale.
- Enables SPICE-type plots, which create a separate scale and axis for each plot variable.

This option does not affect postprocessing of graph data.

.OPTION SDA

Produces a Cadence WSF (ASCII format) post-analysis file for Opus™.

Syntax

```
.OPTION SDA=x
```

Default 0

Description

Use this option to produce a Cadence WSF (ASCII format) post-analysis file for Opus™. Set `SDA=2` to produce this file. This option requires a specific license. The `SDA` is the same as the `CDS` option.

See Also

.OPTION CDS

.OPTION TRCON

Controls the speed of some special circuits.

Syntax

```
.OPTION TRCON=-1|0|1
```

Default `0`

Description

Use this option to control the speed of some special circuits. For some large nonlinear circuits with large TSTOP/TSTEP values, analysis might run for an excessively long time. In this case, HSPICE might automatically set a new and bigger RMAX value to speed up the analysis for primary reference. In most cases, however, HSPICE does not activate this type of auto-speedup process.

For autospeedup to occur, all three of the following conditions must occur:

- N1 (Number of Nodes) > 1,000
- N2 (TSTOP/TSTEP) >= 10,000
- N3 (Total Number of Diode, BJTs, JFETs and MOSFETs) > 300

Autospeedup is most likely to occur if the circuit also meets either of the following conditions:

- N2 >= 1e+8 and N3 > 500, or
- N2 >= 2e+5 and N3 > 1e+4
- `TRCON=3`: enable auto-speedup only. HSPICE invokes auto-speed up if:
  - there are more than 1000 nodes, or
  - there are more than 300 active devices, or
  - Tstop/Tstep (as defined in .TRAN) > 1e8.

When auto-speedup is active, RMAX increases, and HSPICE can take larger timesteps.

- `TRCON=2`: enables auto-convergence only.
  - HSPICE invokes auto-convergence if you use the default integration method (trapezoidal), and if HPSICE fails to converge, an “internal timestep too small” error is issued.
  - Auto-convergence sets `METHOD=gear`, `LVLTIM=2`, and starts the transient simulation again from `time=0`.
- `TRCON=1`: enables both auto-convergence and auto-speedup.
- `TRCON=0`: disables both auto-convergence and auto-speedup (default).
- `TRCON=-1`: same as `TRCON=0`.

`TRCON` also controls the automatic convergence process (autoconvergence) as well as the automatic speedup (autospeedup) processes in HSPICE. HSPICE also uses autoconvergence in DC analysis if the Newton-Raphson (N-R) method fails to converge.

If the circuit fails to converge using the trapezoidal (TRAP) numerical integration method (for example because of trapezoidal oscillation), HSPICE uses the GEAR method and LTE timestep algorithm to run the transient analysis again from `time=0`. This process is called autoconvergence.

Autoconvergence sets options to their default values before the second try:

```
METHOD=GEAR, LVLTIM=2, MBYPASS=1.0,
+ BYPASS=0.0, SLOPETOL=0.5,
+ BYTOL= min{mbypas*vntol and reltol}
```

`RMAX=2.0` if it was 5.0 in the first run; otherwise RMAX does not change.

.OPTION ZUKEN

Enables or disables the Zuken interface.

Syntax

```
.OPTION ZUKEN=x
```

Description

Use this option to enable or disable the Zuken interface.

- If x is `2`, the interface is enabled.
- If x is `1` (default), the interface is disabled.

.OPTION ZUKEN

B

How Options Affect other Options

Describes the effects of specifying control options on other options in the netlist.

The following options either impact or are impacted by the specifying of other .OPTION parameters:

- GEAR Method
- ACCURATE
- FAST
- GEAR Method, ACCURATE
- ACCURATE, GEAR Method
- ACCURATE, FAST
- GEAR Method, FAST
- GEAR Method, ACCURATE, FAST
- RUNLVL=N
- RUNLVL, ACCURATE, FAST, GEAR method
- DVDT=1,2,3
- LVLTIM=0,2,3
- KCLTEST
- BRIEF
- Option Notes
- Finding the Golden Reference for Options

GEAR Method

Specifying .OPTION METHOD=GEAR sets the values of other options as follows:

- BYPASS = 0
- BYTOL = 50u
- DVDT = 3
- LVLTIM = 2
- MBYPASS = 1.0
- METHOD = 2
- RMAX = 2.0
- SLOPETOL = 500m

ACCURATE

Specifying the ACCURATE option sets the values of other options as follows:

- ABSVAR = 0.2
- ACCURATE =1
- BYPASS = 0
- DVDT = 2
- FFT_ACCU = 1
- FT = 0.2
- LVLTIM = 3
- RELMOS = 0.01
- RELVAR = 0.2

FAST

Specifying the FAST option sets the values of other options as follows:

- BYTOL = 50u
- DVDT = 3
- BYPASS = 0
- DVDT = 2
- FAST = 1
- MBYPASS = 1.0
- RMAX = 2.0
- SLOPETOL = 500m

GEAR Method, ACCURATE

Specifying .OPTION METHOD=GEAR first with the ACCURATE option sets the values of other options as follows:

- ABSVAR = 0.2
- ACCURATE =1
- BYPASS = 0
- BYTOL = 50u
- DVDT = 2
- FFT_ACCU = 1
- FT = 0.2
- LVLTIM = 3
- MBYPASS = 1.0
- METHOD = 2
- RELMOS = 0.01
- RELVAR = 0.2
- RMAX = 2
- SLOPETOL = 500m

Note:

When GEAR is specified first, DVDT=2 and LVLTIM=3.

ACCURATE, GEAR Method

Specifying the ACCURATE option first in with.OPTION METHOD=GEAR sets the values of other options as follows:

- ABSVAR = 0.2
- ACCURATE =1
- BYPASS = 0
- BYTOL = 50u
- DVDT = 3
- FFT_ACCU = 1
- FT = 0.2
- LVLTIM = 2
- MBYPASS = 1.0
- METHOD = 2
- RELMOS = 0.01
- RELVAR = 0.2
- RMAX = 2
- SLOPETOL = 500m

Note:

When ACCURATE is specified before the GEAR method, then DVDT=2, LVLTIM=3.

ACCURATE, FAST

Specifying the ACCURATE option with the FAST option sets the values of other options as follows:

- ABSVAR = 0.2
- ACCURATE =1
- BYPASS = 0
- BYTOL = 50u
- DVDT = 2

- FAST = 1
- FFT_ACCU = 1
- FT = 0.2
- LVLTIM = 3
- MBYPASS = 1.0
- RELMOS = 0.01
- RELVAR = 0.2
- RMAX = 2
- SLOPETOL = 500m

Note:

The ACCURATE and FAST options are order-independent.

GEAR Method, FAST

Specifying .OPTION METHOD=GEAR in combination with the FAST option sets the values of other options as follows:

- BYTOL = 50u
- DVDT = 3
- FAST = 1
- LVLTIM = 2
- MBYPASS = 2
- METHOD = 0.01
- RMAX = 2
- SLOPETOL = 500m

Note:

The METHOD=GEAR and FAST options are order-independent.

GEAR Method, ACCURATE, FAST

Specifying .OPTION METHOD=GEAR first in combination with the ACCURATE and FAST options sets the values of other options as follows:

- ABSVAR = 0.2
- ACCURATE =1
- BYPASS = 0
- BYTOL = 50u
- DVDT = 2
- FAST = 1
- FFT_ACCU = 1
- FT = 0.2
- LVLTIM = 3
- METHOD = 2
- MBYPASS = 1.0
- RELMOS = 0.01
- RELVAR = 0.2
- RMAX = 2
- SLOPETOL = 500m

Note:

If GEAR is specified first, then DVDT=2 LVLTIM=3. Otherwise, the METHOD=GEAR, ACCURATE, and FAST options are order-independent.

RUNLVL=N

Specifying the RUNLVL option with any legal numeric value sets the following options:

- BYPASS = 2
- DVDT = 3
- LVLTIM = 4
- RUNLVL = N
- SLOPETOL = 500m

RUNLVL, ACCURATE, FAST, GEAR method

Specifying the options RUNLVL, ACCURATE, and FAST with METHOD=GEAR is order-independent:

- RUNLVL option (LVLTIM = 4) is always on
- GEAR method is always selected
- RUNLVL = 5 is always selected
- FAST has no effect on RUNLVL

DVDT=1,2,3

Specifying the DVDT option= 1,2,3 sets the following options:

- BYPASS = 0
- BYTOL = 50u
- MBYPASS = 1.0
- RMAX = 2
- SLOPETOL = 500m

LVLTIM=0,2,3

Specifying the LVLTIM option= 1,2,3 sets the following options:

- BYPASS = 0
- BYTOL = 50u
- MBYPASS = 1.0
- RMAX = 2
- SLOPETOL = 500m

These options are order-independent.

Note:

The DVDT value is ignored if LVLTIM = 2

KCLTEST

Specifying the KCLTEST option sets the following options:

- ABSTOL = 1u
- RELI = 1u

KCLTEST is order-dependent with ABSTOL and RELI.

BRIEF

Specifying the BRIEF option resets the following options to their defaults:

- NODE
- LIST
- OPTS

and sets the NOMOD option.

The BRIEF option is order-dependent with the affected options. If option BRIEF is specified after NODE, LIST, OPTS, and NOMOD, then it resets them. If option BRIEF is specified before NODE, LIST, OPTS, and NOMOD, then those options overwrite whatever values option BRIEF may have set.

Option Notes

- ABSTOL aliases ABSI
- VNTOL aliases ABSV
- If ABSVDC is not set, VNTOL sets it
- DCTRAN aliases CONVERGE
- GMIN does not overwrite GMINDC, nor does GMINDC overwrite GMIN
- RELH only takes effect when ABSH is non-zero
- RELTOL aliases RELV
- RELVDC defaults to RELTOL
- If RELTOL < BYTOL, BYTOL = RELTOL
- RELVAR applies to LVLTIM = 1 or 3 only

- CHGTOL, RELQ & TRTOL are the only error tolerance options for LVLTIM = 2 (LTE)
- The DVDT algorithm works with LVLTIM = 1 and 3

RUNLVL Option Notes

If RUNLVL is invoked, you can disable it by:

- Adding .OPTION RUNLVL=0 to your current simulation job.
- Copying $installdir/hspice.ini to your HOME directory and customize it by adding .OPTION RUNLVL=0, which disables it for all of your simulation jobs.
- Re-invoking the $installdir/bin/config program and deselecting the option runlvl setting in box 'hspice.ini' which disables it for the whole group of simulation jobs.

If RUNLVL is invoked, some options are ignored or automatically set:

Options below are automatically set (user setting will overwrite them):

- If runlvl=6, then .option bypass=0
- If runlvl=1|2|3|4|5, then .option bypass=2
- The following options are ignored; they are replaced by automated algorithms: lvltim, dvdt, ft, fast, trtol, absvar, relvar, relq, chgtol, dvtr, imin, itl3, rmax

If RUNLVL is invoked, actual values of options used by HSPICE are:

- runlvl= 3
- bypass= 2
- mbypass= 2.00
- bytol= 100.00u
- bdfatol=1e-3
- bdfrtol=1e-3

Finding the Golden Reference for Options

When trying to determine the acceptable trade-off between HSPICE accuracy and transient analysis simulation performance, it is important to first establish a reference value for the measurements you are using to evaluate the

performance (speed and accuracy) of a given HSPICE configuration. There are multiple ways to configure HSPICE for higher accuracy. The following is a good starting point that you might want to modify for your specific application:

```
.OPTION RUNLVL=6 ACCURATE KCLTEST DELMAX=a_small_value
```

The options are described as follows:

Options	Description
RUNLVL	Invokes the RUNLVL algorithm and sets tolerances to their tightest values. Refer to .OPTION RUNLVL for more details:
ACCURATE	Sets even more HSPICE OPTIONs to tighter tolerances. (See .OPTION ACCURATE for details.
BACKGROUND	Activates Kirchhoff's Current Law testing for every circuit node. (See .OPTION KCLTEST for details.
DELMAX	Sets the largest timestep that HSPICE is allowed to take. It should be set to the smallest value (1ps, for example) that still allows the simulation to finish in a reasonable amount of time. Typically, it should be set approximately `1/(20*highest-frequency-activity-in-the-circuit)` Warning: This option can create very large *tr0* files. Be careful to only probe the needed nodes (use `.OPTION PROBE` combined with `.PROBE`). See .OPTION DELMAX for details.

Index

Symbols

(X0R, X0I) option 612
(X1R, X1I) option 613
(X2R, X2I) option 614

A

ABSH option 298
ABSI option 299, 432
ABSMOS option 300, 432
ABSTOL option 301
ABSV option 302
ABSVAR option 303
ABSVDC option 304
AC analysis
  magnitude 307
  optimization 19
  output 307
  phase 307
.AC command 19
  external data 55
ACCT option 305
ACCURATE option 306
  combined with FAST option 674
  combined with FAST option and GEAR method 675
  combined with GEAR option 673, 674
  plus FAST and RUNLVL options and METHOD=GEAR 677
.ACMATCH command 22
ACOUT option 307
algorithms
  DVDT 303, 448
  local truncation error 448, 517, 598
  pivoting 496
  timestep control 371
  transient analysis timestep 448
  trapezoidal integration 459
.ALIAS command 26
ALL keyword 203, 233
allows 336
ALT9999 option 662
ALTCC option 308
ALTCHK option 309
alter block commands 15
ALTER cases, multiprocessing 4
.ALTER command 28, 69
Analog Artist interface 508
  See also Artist
Analysis commands 14
analysis, network 660
APPENDALL option 310
.APPENDMODEL 30
arguments, command-line
  hspice 1
  hspicerf 10
arithmetic expression 162
ARTIST option 311
ASCII output 10
ASCII output data 458, 663, 667
ASPEC option 312
AT keyword 156, 160
autoconvergence 352
AUTOSTOP option 313
average nodal voltage, with .MEASURE 164
average value, measuring 165
AVG keyword 164, 176

B

BA_TERMINAL option 317
back annotation
  terminal name 317
BADCHR option 318, 321
BETA keyword 232
.BIASCHK command 31
BIASFILE option 322
BIASINTERVAL option 323
BIASNODE option 324
BIASPARALLEL option 325

BIAWARN option 326
BINPRNT option 327
bisection
  pushout 180
BKPSIZ option 662
BPNMATCHTOL option 328
branch current error 299
breakpoint table, size 662
BRIEF option 203, 206, 329, 443, 472, 482, 488
  effect on other options 678
BSIM4PDS option 330
bus notation 644
BYPASS option 331
BYTOL option 332

C

Cadence
  Opus 663, 667
  WSF format 663, 667
capacitance
  charge tolerance, setting 334
  CSHUNT node-to-ground 343
  table of values 333
capacitor, models 189
CAPTAB option 333
CDS option 662
characterization of models 63
charge tolerance, setting 334
.CHECK EDGE command 37
.CHECK FALL command 39
.CHECK GLOBAL_LEVEL command 40
.CHECK HOLD command 41
.CHECK IRDROP command 43
.CHECK RISE command 45
.CHECK SETUP command 47
.CHECK SLEW command 49
CHGTOL option 334
CLOSE optimization parameter 190
CMIFLAG option 335
CMIPATH option 336
CMIUSRFLAG option 337
CO option 273, 277, 662, 663
column laminated data 59
command-line arguments
  hspice 1
  hspicerf 10
commands
  .AC 19
  .ACMATCH 22
  .ALIAS 26
  .ALTER 28, 69
  alter block 15
  analysis 14
  .APPENDMODEL 30
  .BIASCHK 31
  .CHECK EDGE 37
  .CHECK FALL 39
  .CHECK GLOBAL_LEVEL 40
  .CHECK HOLD 41
  .CHECK IRDROP 43
  .CHECK RISE 45
  .CHECK SETUP 47
  .CHECK SLEW 49
  .CONNECT 51
  .DATA 54
  .DC 61
  .DCMATCH 66
  .DCVOLT 68
  .DEL LIB 69
  .DISTO 74
  .DOUT 76
  .EBD 78
  .ELSE 80
  .ELSEIF 81
  .END 82
  .ENDDATA 83
  .ENDIF 84
  .ENDL 85
  .ENDS 86
  .ENV 87
  .ENVFFT 88
  .ENVOSC 90
  .EOM 91
  .FFT 92
  .FLAT 97
  .FOUR 98
  .FSOPTIONS 99
  .GLOBAL 102
  .GRAPH 656
  .HB 103
  .HBAC 107
  .HBLIN 108
  .HBLSP 110

.HBNOISE 112
.HBOSC 115
.HBXF 120
.HDL 121
.IBIS 124
.IC 128
.ICM 130
.IF 132
.INCLUDE 134
.LAYERSTACK 135
.LIB 137
.LIN 140
.LOAD 144
.LPRINT 146
.MACRO 147
.MALIAS 149
.MATERIAL 151
.MEASURE 153
.MEASURE PHASENOISE 175
.MEASURE PTDNOISE 179
.MEASURE(ACMATCH) 182
.MEASURE(DCMATCH) 183
.MODEL 188
.MOSRA 194
.MOSRAPRINT 197
.NET 658
.NODESET 198
.NOISE 200
.OP 203
.OPTION 205
.PARAM 207
.PAT 211
.PHASENOISE 213
.PKG 216
.PLOT 660
.POWER 218
.POWERDC 220
.PRINT 221
.PROBE 225
.PROTECT 227
.PTDNOISE 228
.PZ 231
.SAVE 233
.SENS 235
.SHAPE 237
.SNFT 246
.SNOSC 251
.SNXF 254
.STATEYE 256
.STIM 258
subcircuit 18
.SUBCKT 263
.SURGE 266
.SWEEPBLOCK 267
.TEMP (or) .TEMPERATURE 269
.TF 271
.TITLE 272
.TRAN 273
.UNPROTECT 281
.VARIATION 282
.VEC 285
Verilog-A 18
.WIDTH 661
Common Simulation Data Format 369
concatenated data files 58
Conditional Block 15
conductance
current source, initialization 387
minimum, setting 388
models 353
MOSFETs 389
negative, logging 368
node-to-ground 393
sweeping 390
.CONNECT command 51
control options
printing 488
setting 206
transient analysis
limit 602
CONVERGE option 338, 354
convergence
for optimization 191
problems
changing integration algorithm 459
CONVERGE option 338, 354
DCON setting 352
decreasing the timestep 381
nonconvergent node listing 352
operating point Debug mode 203
setting DCON 352
steady state 390
CPTIME option 339
CPU time, reducing 473
CROSS keyword 159
CSCAL option 340, 446

CSDF option 341
CSHDC option 342
CSHUNT option 343
current
  ABSMOS floor value for convergence 516
  branch 299
  operating point table 203
CURRENT keyword 203
CUSTCMI option 344
CUT optimization parameter 190
CVTOL option 345

D

-d argument 3
D_IBIS option 346
.DATA command 54, 58
  datanames 56
  external file 54
  for sweep data 55
  inline data 56
data files, disabling printout 329, 482
DATA keyword 19, 55, 61, 274
datanames 56, 260
DC
  analysis
    decade variation 62
    initialization 350
    iteration limit 424
    linear variation 62
    list of points 62
    octave variation 62
  optimization 61
.DC command 61, 63
  external data with .DATA 55
DCAP option 347
DCCAP option 348
DCFOR option 349
DCHOLD option 350
DCIC option 351
.DCMATCH command 66
DCON option 352
DCSTEP option 353
DCTRAN option 354
.DCVOLT command 68, 128
DEBUG keyword 203
DEC keyword 20, 62, 276
DEFAD option 355
DEFAS option 356
DEFL option 357
DEFNRD option 358
DEFNRS option 359
DEFPD option 360
DEFPS option 361
DEFSA option 362
DEFSB option 363
DEFSD option 364
DEFW option 365
.DEL LIB command 69
  with .ALTER 69
  with .LIB 69
delays
  group 599
DELMAX option 366, 526
DELTA internal timestep
  See also timestep
derivative function 170
DERIVATIVE keyword 170
derivatives, measuring 160
DI option 367
DIAGNOSTIC option 368
DIFSIZ optimization parameters 190
DIM2 distortion measure 75
DIM3 distortion measure 75
diode models 189
.DISTO command 74
distortion
  HD2 75
  HD3 75
distortion measures
  DIM2 75
  DIM3 75
DLENCSDF option 369
.DOUT command 76
DV option 352, 370
DVDT
  algorithm 303
  option 371, 448
DVDT option 371
DVDToption
  value e1,2,3 ffect on other options
    .OPTION DVDT
      value 1,2,3 effect on other options
        677

DVTR option 372

E

.EBD command 78
element
  checking, suppression of 473
  OFF parameter 483
.ELSE command 80
.ELSEIF command 81
EM_RECOVERY option 373
ENABLE command 620
Encryption 16
.END command 82
  for multiple HSPICE runs 82
  location 82
.ENDDATA command 83
ENDDATA keyword 54, 57, 58
.ENDIF command 84
.ENDL command 85, 138
.ENDS command 86
.ENV command 87
envelope simulation 87
  FFT on output 88
  oscillator startup, shutdown 90
.ENVFFT command 88
.ENVOSC command 90
.EOM command 91
EPSMIN option 374
equation 162
ERR function 173, 174
ERR1 function 173
ERR2 function 173
ERR3 function 173
error function 173
errors
  branch current 299
  function 174
  internal timestep too small 343, 527
  optimization goal 156
  tolerances
    ABSMOS 300
    branch current 299
    RELMOS 300
example, subcircuit test 147, 264
EXPLI option 375
EXPMAX option 376
expression, arithmetic 162
external data files 56

F

FALL keyword 159
fall time
  verification 39
FAST option 377
  effect on other options 672
FASToption
  combined with ACCURATE option 674
  combined with ACCURATE option and GEAR method 675
  combined with GEAR method 675
  plus ACCURATE and RUNLVL options and METHOD=GEAR 677
.FFT command 92
FFT_ACCURATE option 378
FFTOUT option 379
FIL keyword 56
files
  column lamination 59
  concatenated data files 58
  filenames 56
  hspice.ini 458
  include files 134, 139
  input 2
  multiple simulation runs 82
  output
    version number 2, 11
FIND keyword 160
FIND, using with .MEASURE 158
.FLAT command 97
floating point overflow
  CONVERGE setting 338
  setting GMINDC 389
FMAX option 380
.FOUR command 98
FREQ
  model parameter 657
frequency
  ratio 74
  sweep 21
FROM parameter 174
FS option 232, 381
FSCAL option 382
.FSOPTIONS command 99

FT option 383
functions
  ERR 174
  ERR1 173
  ERR2 173
  ERR3 173
  error 173

G

GDCPATH option 384
GEAR method
  combined with FAST option 675
  effect on options 672
GEAR option
  combined with ACCURATE option 673, 674
  effect on other options 672
GENK option 385
GEOSHRINK option 386
.GLOBAL command 102
global node names 102
GMAX option 387
GMIN option 388, 389
GMINDC option 389
GOAL keyword 165
GRAD optimization parameter 191
GRAMP
  calculation 352
  option 390
.GRAPH command 656
graph data file (Viewlogic format) 369
ground bounce checking 43
group delay, calculating 599
GSCAL option 391
GSHDC option 392
GSHUNT option 393

H

-h argument
  usage information 11
H9007 option 663, 664
harmonic balance analysis 104
harmonic balance noise analysis 114
harmonic balance transfer analysis 120, 254
harmonic balance-based periodic AC analysis 107
harmonic distortion 75
.HB command 103
HB_GIBBS option 400
.HBAC command 107
HBACKRYLOVDIM option 394
HBACKRYLOVITR option 395
HBACTOL option 396
HBCONTINUE option 397
HBFREQABSTOL option 398
HBFREQRELTOL option 399
HBJREUSE option 401
HBJREUSETOL option 402
HBKRYLOVDIM option 403
HBKRYLOVMAXITER option 405
HBKRYLOVTOL option 404
.HBLIN command 108
HBLINESEARCHFAC option 406
.HBLSP command 110
HBMAXITER option 407
HBMAXOSCITER option 408
.HBNOISE command 112
.HBOSC command 115
HBPROBETOL option 409
HBSOLVER option 410
HBTOL option 411
HBTRANFREQSEARCH option 412
HBTRANINIT option 413
HBTRANPTS option 414
HBTRANSTEP option 415
.HBXF command 120
HCI and NBTI analysis 196
HD2 distortion 75
HD3 distortion 75
.HDL command 121
HIER_DELIM option 416
HIER_SCALE option 417
HSPICE
  job statistics report 305
  version
    H9007 compatibility 664
hspice
  arguments 1
  command 1
hspice.ini file 458
hspicerf
  arguments 10
  command 10

-html argument 3

I

-I argument 4
-i argument 2
.IBIS command 124
IBIS commands 16
.IC command 68, 128
  from .SAVE 234
IC keyword 233
IC parameter 68
.ICM command 130
ICSWEEP option 418
IDELAY command 621
.IF command 132
IGNOR keyword 173
IMAX option 419, 427
IMIN option 420, 426
.INCLUDE command 134
include files 134, 139
indepout 260
indepvar 259, 260, 261
inductors, mutual model 189
INGOLD option 421, 455
initial conditions
  saving and reusing 418
initialization 483
inline data 56
inner sweep 58
input
  data
    adding library data 69
    column laminated 59
    concatenated data files 58
    deleting library data 69
    external, with .DATA command 55
    filenames on networks 60
    formats 56, 59
    include files 134
    printing 443
    suppressing printout 443
  file names 2
  netlist file 82
INTEG keyword 164, 169, 176
  used with .MEASURE 164
integral function 168
integration
  backward Euler method 452
interfaces
  Analog Artist 508
  Mentor 665
  MSPICE 665
  ZUKEN 669
intermodulation distortion 75
INTERP option 422
IO command 623
iterations
  limit 424
  maximum number of 428
ITL1 option 424
ITL2 option 425
ITL3 option 426
ITL4 option 427
ITL5 option 428
ITLPTRAN option 429
ITLPZ option 430
ITROPT optimization parameter 191
ITRPRT option 431

J

Jacobian data, printing 487

K

KCLTEST option 432
KCLTESToption
  effect on other options
    .OPTION KCLTEST
      effect on other options 678
keywords
  .AC command parameter 19
  ALL 203, 233
  AT 156, 160
  AVG 164, 176
  BETA 232
  CROSS 159
  CURRENT 203
  DATA 19, 55, 61, 274
  .DATA command parameter 55
  .DC command parameter 61
  DEBUG 203
  DEC 20, 62, 276
  DERIVATIVE 170
  ENDDATA 54, 57, 58

FALL 159
FIL 56
FIND 160
FS 232
IGNOR 173
INTEG 164, 169, 176
LAM 56, 60
LAST 159
LIN 20, 62, 276
MAXFLD 232
.MEASUREMENT command parameter 164, 176
MER 56, 59
MINVAL 173
MODEL 61
MONTE 20, 62, 274
NONE 203, 233
NUMF 232
OCT 20, 62, 276
OPTIMIZE 62
POI 20, 62, 276
PP 164, 176
RESULTS 62
RIN 659
RISE 159
START 275
SWEEP 20, 62, 275
target syntax 156, 160
TO 164, 168, 174
TOL 232
TOP 233
.TRAN command parameter 274
TRIG 155
VOLTAGE 203
WEIGHT 165, 173
weight 165
WHEN 160
Kirchhoff's Current Law (KCL) test 432
KLIM option 433

L

LA_FREQ option 434
LA_MAXR option 435
LA_MINC option 436
LA_TIME option 437
LA_TOL option 438
LAM keyword 56, 60
keywords
LAM 56
laminated data 59
LAST keyword 159
latent devices
excluding 377
.LAYERSTACK command 135
LENNAM option 439
.LIB command 137
call command 138
in .ALTER blocks 138
nesting 138
with .DEL LIB 69
libraries
adding with .LIB 69
building 138
deleting 69
private 227
protecting 227
Library Management 16
LIMPTS option 440
LIMTIM option 441
.LIN command 140
LIN keyword 20, 62, 276
LISLVL option 442
LIST option 443
listing, suppressing 227
.LOAD command 144
LOADHB option 444
LOADSNINIT option 445
local truncation error algorithm 448, 517, 598
.LPRINT command 146
LVLTIM option 448, 598
value 0,2,3 effect on other options 677

M

MACMOD option 449
.MACRO command 147
macros 69
magnetic core models 189
.MALIAS command 149
.MATERIAL command 151
Material Properties 16
matrix
minimum pivot values 500
parameters 659

row/matrix ratio 499
size limitation 498
MAX 164
MAX parameter 164, 176, 190
MAXAMP option 451
MAXFLD keyword 232
maximum value, measuring 165
MAXORD option 452
MBYPASS option 453
MCBRIEF option 454
MEASDGT option 455
MEASFAIL option 456
MEASFILE option 457
MEASOUT option 458
MEASSORT option 664
.MEASURE command 153, 455, 458
average nodal voltage 164
expression 162, 163
propogation delay 155
.MEASURE PHASENOISE 175
.MEASURE(ACMATCH) command 182
.MEASURE(DCMATCH) command 183
measuring average values 165
measuring derivatives 160
Mentor interface 665
MENTOR option 664
MER keyword 56, 59
keywords
MER 56
messages
See also errors, warnings
messages, pivot change 497
METHOD option 459
MIN 164
MIN parameter 164, 176
minimum value, measuring 165
MINVAL keyword 173
.MODEL command 188
CLOSE 190
CUT 190
DEV 192
DIFSIZ 190
distribution 192
GRAD 191
ITROPT 191
keyword 191
LOT 192
MAX 190
model name 188
PARMIN 191
RELIN 191
RELOUT 191
type 189
.MODEL command for .GRAPH 657
MODEL keyword 61
model parameters
.GRAPH command parameters 657
MONO 657
output 657
suppressing printout of 475
TEMP 269
TIC 657
models
BJTs 189
capacitors 189
characterization 63
diode 189
JFETs 189
magnetic core 189
MOSFETs 189
mutual inductors 189
names 188
npn BJT 189
op-amps 189
optimization 189
plot 189
private 227
protecting 227
simulator access 138
types 189
models, diode 189
MODMONTE option 462
MODSRH option 665
MONO model parameter 657
Monte Carlo
AC analysis 19
DC analysis 61
.MODEL parameters 191
time analysis 274
variation block options 616
MONTE keyword 20, 62, 274
MONTECON option 464
.MOSRA command 194
MOSRALIFE option 465

.MOSRAPRINT command 197
MOSRASORT option 466
MSPICE simulator interface 665
-mt argument 4
MTTHRESH option 467
MU option 468
multiprocessing, ALTER cases 4
multithreading, lowering device number threshold 467

N

-n argument 2, 11
namei 259, 260, 261
NBTI and HCI analysis 196
NCFILTER option 469
n-channel, MOSFET's models 189
NCWARN option 470
negative conductance, logging 368
nested library calls 138
.NET comamnd 658
network
  analysis 660
  filenames 60
network analysis 660
NEWTOL option 471
Node Naming 17
NODE option 472
nodes
  cross-reference table 472
  global versus local 102
  printing 472
.NODESET command 198
  from .SAVE 234
NODESET keyword 233
node-to-element list 497
NOELCK option 473
noise
  folding 232
  numerical 343
  sampling 232
.NOISE command 200
NOISEMINFREQ option 474
NOMOD option 475
NONE keyword 203, 233
NOPAGE option 476
NOPIV option 477
NOTOP option 478
NOWARN option 479
npn BJT models 189
npoints 259, 260, 261
NUMDGT option 480
numerical integration algorithms 459
numerical noise 343, 393
NUMERICAL_DERIVATIVES option 481
NUMF keyword 232
NXX option 482

O

-o argument 2
obsolete commands
  .GRAPH (use .PRINT) 656
  .NET (use .LIN) 658
  .PLOT (use .PRINT) 660
  .WIDTH 661
obsolete options
  .OPTION ALT999 or ALT9999 662
  .OPTION BKPSIZ 662
  .OPTION CDS 663
  .OPTION CO 663
  .OPTION H9007 664
  .OPTION MEASSORT 664
  .OPTION MENTOR 665
  .OPTION PLIM 666
  .OPTION TRCON 667
  .OPTION ZUKEN 668
  .SDA 667
OCT keyword 20, 62, 276
ODELAY command 624
OFF option 483
.OP command 203
op-amps model, names 189
operating point
  capacitance 333
  .IC command initialization 68
  restoring 144
  solution 483
  voltage table 203
OPFILE option 484
OPTCON option 485
optimization
  AC analysis 19

DC analysis 61
error function 156
iterations 191
models 189
time
analysis 274
optimization parameter, DIFSIZ 190
OPTIMIZE keyword 62
.OPTION (X0R, X0I) 612
.OPTION (X1R, X1I) 613
.OPTION (X2R, X2I) 614
.OPTION ABSH 298
.OPTION ABSI 299
.OPTION ABSMOS 300
.OPTION ABSTOL 301
.OPTION ABSV 302
.OPTION ABSVAR 303
.OPTION ABSVDC 304
.OPTION ACCT 305
.OPTION ACCURATE 306
combined with FAST option 674
combined with FAST option and GEAR method 675
combined with GEAR option 673, 674
plus FAST and RUNLVL options, METHOD=GEAR 677
.OPTION ACOUT 307
.OPTION ALT9999 662
.OPTION ALTCC 308
.OPTION ALTCHK 309
.OPTION APPENDALL 310
.OPTION ARTIST 311
.OPTION ASPEC 312
.OPTION AUTOSTOP 313
.OPTION BA_TERMINAL 317
.OPTION BADCHR 318, 321
.OPTION BDFATOL 319, 320
.OPTION BIASFILE 322
.OPTION BIASINTERVAL 323
.OPTION BIASNODE 324
.OPTION BIASPARALLEL 325
.OPTION BIAWARN 326
.OPTION BINPRNT 327
.OPTION BKPSIZ 662
.OPTION BPNMATCHTOL 328
.OPTION BRIEF 203, 206, 329, 443, 472, 482, 488
effect on other options 678
.OPTION BSIM4PDS 330
.OPTION BYPASS 331
.OPTION BYTOL 332
.OPTION CAPTAB 333
.OPTION CDS 662
.OPTION CHGTOL 334
.OPTION CMIFLAG 335
.OPTION CMIPATH 336
.OPTION CMIUSRFLAG 337
.OPTION CO 273, 277, 662, 663
.OPTION command 205
.OPTION CONVERGE 338
.OPTION CPTIME 339
.OPTION CSCAL 340, 446
.OPTION CSDF 341
.OPTION CSHDC 342
.OPTION CSHUNT 343
.OPTION CUSTCMI 344
.OPTION CVTOL 345
.OPTION D_IBIS 346
.OPTION DCAP 347
.OPTION DCCAP 348
.OPTION DCFOR 349
.OPTION DCHOLD 350
.OPTION DCIC 351
.OPTION DCON 352
.OPTION DCSTEP 353
.OPTION DCTRAN 354
.OPTION DEFAD 355
.OPTION DEFAS 356
.OPTION DEFL 357
.OPTION DEFNRD 358
.OPTION DEFNRS 359
.OPTION DEFPD 360
.OPTION DEFPS 361
.OPTION DEFSA 362
.OPTION DEFSB 363
.OPTION DEFSD 364
.OPTION DEFW 365
.OPTION DELMAX 366
.OPTION DI 367
.OPTION DIAGNOSTIC 368

.OPTION DLENCSDF 369
.OPTION DV 370
.OPTION DVDT 371
.OPTION DVTR 372
.OPTION EM_RECOVERY 373
.OPTION EPSMIN 374
.OPTION EXPLI 375
.OPTION EXPMAX 376
.OPTION FAST 377
  combined with ACCURATE option 674
  combined with ACCURATE option and GEAR method 675
  combined with GEAR method 675
  effect on other options 672
  plus ACCURATE and RUNLVL options and METHOD=GEAR 677
.OPTION FFT_ACCURATE 378
.OPTION FFTOUT 379
.OPTION FMAX 380
.OPTION FS 381
.OPTION FSCAL 382
.OPTION FT 383
.OPTION GDCPATH 384
.OPTION GEAR
  combined with ACCURATE option 673, 674
  effects on other options 672
.OPTION GENK 385
.OPTION GEOSHRINK 386
.OPTION GMAX 387
.OPTION GMIN 388
.OPTION GMINDC 389
.OPTION GRAMP 390
.OPTION GSCAL 391
.OPTION GSHDC 392
.OPTION GSHUNT 393
.OPTION H9007 663
.OPTION HB_GIBBS 400
.OPTION HBACKRYLOVDIM 394
.OPTION HBACKRYLOVITR 395
.OPTION HBACTOL 396
.OPTION HBCONTINUE 397
.OPTION HBFREQABSTOL 398
.OPTION HBFREQRELTOL 399
.OPTION HBJREUSE 401
.OPTION HBJREUSETOL 402
.OPTION HBKRYLOVDIM 403
.OPTION HBKRYLOVMAXITER 405
.OPTION HBKRYLOVTOL 404
.OPTION HBLINESEARCHFAC 406
.OPTION HBMAXITER 407
.OPTION HBMAXOSCITER 408
.OPTION HBPROBETOL 409
.OPTION HBSOLVER 410
.OPTION HBTOL 411
.OPTION HBTRANFREQSEARCH 412
.OPTION HBTRANINIT 413
.OPTION HBTRANPTS 414
.OPTION HBTRANSTEP 415
.OPTION HIER_DELIM 416
.OPTION HIER_SCALE 417
.OPTION ICSWEEP 418
.OPTION IMAX 419
.OPTION IMIN 420
.OPTION INGOLD 421
.OPTION INTERP 422
.OPTION ITL1 424
.OPTION ITL2 425
.OPTION ITL3 426
.OPTION ITL4 427
.OPTION ITL5 428
.OPTION ITLPTRAN 429
.OPTION ITLPZ 430
.OPTION ITRPRT 431
.OPTION KCLTEST 432
.OPTION KLIM 433
.OPTION LA_FREQ 434
.OPTION LA_MAXR 435
.OPTION LA_MINC 436
.OPTION LA_TIME 437
.OPTION LA_TOL 438
.OPTION LENNAM 439
.OPTION LIMPTS 440
.OPTION LIMTIM 441
.OPTION LISLVL 442
.OPTION LIST 443
.OPTION LOADHB 444
.OPTION LOADSNINIT 445
.OPTION LVLTIM 448
  value 0,2,3 effect on other options 677
.OPTION MACMOD 449
.OPTION MAXAMP 451

.OPTION MAXORD 452
.OPTION MBYPASS 453
.OPTION MCBRIEF 454
.OPTION MEASDGT 455
.OPTION MEASFAIL 456
.OPTION MEASFILE 457
.OPTION MEASOUT 458
.OPTION MEASSORT 664
.OPTION MENTOR 664
.OPTION METHOD 459
.OPTION METHOD=GEAR
  combined with FAST option 675
  effects on other options 672
.OPTION MODMONTE 462
.OPTION MODSRH 665
.OPTION MONTECON 464
.OPTION MTTHRESH 467
.OPTION MU 468
.OPTION NCFILTER 469
.OPTION NCWARN 470
.OPTION NEWTOL 471
.OPTION NODE 472
.OPTION NOELCK 473
.OPTION NOISEMINFREQ 474
.OPTION NOMOD 475
.OPTION NOPAGE 476
.OPTION NOPIV 477
.OPTION NOTOP 478
.OPTION NOWARN 479
.OPTION NUMDGT 480
.OPTION NUMERICAL_DERIVATIVES 481
.OPTION NXX 482
.OPTION OFF 483
.OPTION OPFILE 484
.OPTION OPTCON 485
.OPTION OPTLST 487
.OPTION OPTS 488
.OPTION PARHIER 489
.OPTION PATHNUM 490, 493
.OPTION PHASENOISEAMPM 495
.OPTION PHASENOISEKRYLOVDIM 491
.OPTION PHASENOISEKRYLOVITER 492
.OPTION PHNOISELORENTZ 494
.OPTION PIVOT 496
.OPTION PIVREF 498
.OPTION PIVREL 499
.OPTION PIVTOL 500
.OPTION PLIM 666
.OPTION POST 501
.OPTION POST_VERSION 504
.OPTION POSTLVL 503
.OPTION POSTTOP 506
.OPTION PROBE 507
.OPTION PSF 508
.OPTION PURETP 509
.OPTION PUTMEAS 510
.OPTION PZABS 511
.OPTION PZTOL 512
.OPTION RANDGEN 513
.OPTION RELH 514
.OPTION RELI 515
.OPTION RELMOS 516
.OPTION RELQ 517
.OPTION RELTOL 518
.OPTION RELV 519
.OPTION RELVAR 520
.OPTION RELVDC 521
.OPTION RESMIN 522
.OPTION RISETIME 523
.OPTION RITOL 525
.OPTION RMAX 526
.OPTION RMIN 527
.OPTION RUNLVL 528
  N value effect on other options 676
.OPTION SAVEHB 533
.OPTION SAVESNINIT 534
.OPTION SCALE 535
.OPTION SCALM 536
.OPTION SDA 666
.OPTION SEARCH 537
.OPTION SEED 538
.OPTION SIM_ACCURACY 539
.OPTION SIM_DELTAI 540
.OPTION SIM_DELTAV 541
.OPTION SIM_DSPF 542
.OPTION SIM_DSPF_ACTIVE 544
.OPTION SIM_DSPF_INSERROR 545
.OPTION SIM_DSPF_LUMPCAPS 546
.OPTION SIM_DSPF_MAX_ITER 547
.OPTION SIM_DSPF_RAIL 548

.OPTION SIM_DSPF_SCALEC 549
.OPTION SIM_DSPF_SCALER 550
.OPTION SIM_DSPF_VTOL 551
.OPTION SIM_LA 553
.OPTION SIM_LA_FREQ 554
.OPTION SIM_LA_MAXR 555
.OPTION SIM_LA_MINC 556
.OPTION SIM_LA_MINMODE 557
.OPTION SIM_LA_TIME 558
.OPTION SIM_LA_TOL 559
.OPTION SIM_ORDER 560
.OPTION SIM_OSC_DETECT_TOL 561
.OPTION SIM_POSTAT 562
.OPTION SIM_POSTDOWN 563
.OPTION SIM_POSTSCOPE 564
.OPTION SIM_POSTSKIP 565
.OPTION SIM_POSTTOP 566
.OPTION SIM_POWER_ANALYSIS 567
.OPTION SIM_POWER_TOP 569
.OPTION SIM_POWERDC_ACCURACY 570
.OPTION SIM_POWERDC_HSPICE 571
.OPTION SIM_POWERPOST 572
.OPTION SIM_POWERSTART 573
.OPTION SIM_POWERSTOP 574
.OPTION SIM_SPEF 575
.OPTION SIM_SPEF_ACTIVE 576
.OPTION SIM_SPEF_INSERROR 577
.OPTION SIM_SPEF_LUMPCAPS 578
.OPTION SIM_SPEF_MAX_ITER 579
.OPTION SIM_SPEF_PARVALUE 580
.OPTION SIM_SPEF_RAIL 581
.OPTION SIM_SPEF_SCALEC 582
.OPTION SIM_SPEF_SCALER 583
.OPTION SIM_SPEF_VTOL 584
.OPTION SIM_TG_THETA 585
.OPTION SIM_TRAP 586
.OPTION SLOPETOL 587
.OPTION SNACCURACY 588
.OPTION SNMAXITER 589
.OPTION SPMODEL 590
.OPTION STATFL 591
.OPTION SYMB 592
.OPTION TIMERES 593
.OPTION TNOM 596
.OPTION TRANFORHB 597
.OPTION TRCON 667
.OPTION TRTOL 598
.OPTION UNWRAP 599
.OPTION VAMODEL 600
.OPTION VERIFY 601
.OPTION VFLOOR 602
.OPTION VNTOL 603
.OPTION WACC 604
.OPTION WARNLIMIT 605
.OPTION WDELAYOPT 606
.OPTION WDF 607
.OPTION WINCLUDEGDIMAG 608
.OPTION WL 609
.OPTION WNFLAG 610
.OPTION XDTEMP 611
.OPTION ZUKEN 668
options
  BDFATOL 319, 320
  LISLVL 442
  MOSRALIFE 465
  MOSRASORT 466
  NCFILTERr 469
  WDF 607
OPTLST option 487
OPTS option 488
Opus 663, 667
oscillation, eliminating 459
oscillator analysis 118, 215
OUT, OUTZ command 626
outer sweep 58
Output 17
output
  ASCII 10
  data
    format 455, 508
    limiting 422
    significant digits specification 480
    specifying 440
    storing 458
  data, redirecting 7
  files
    reducing size of 605
    version number, specifying 2, 11
  .MEASURE results 153
  plotting 660–661
  printing 222–224
  printout format 421

redirecting 7, 10
variables
printing 431
probing 225
specifying significant digits for 480
ovari 259, 260, 261

P

.PARAM command 207
parameters
AC sweep 19
DC sweep 61
defaults 489
FROM 174
IC 68
inheritance 489
ITROPT optimization 191
matrix 659
names
.MODEL command parameter name 190
simulator access 138
skew, assigning 139
UIC 68, 128
PARHIER option 489
PARMIN optimization parameter 191
.PAT command 211
path names 490
path numbers, printing 490
PATHNUM option 490, 493
p-channel
JFETs models 189
MOSFET's models 189
peak-to-peak value
measuring 164
PERIOD command 627
PERIOD statement 627
periodic pime-dependent noise analysis 230
.PHASENOISE command 213
PHASENOISEAMPM option 495
PHASENOISEKRYLOVDIM option 491
PHASENOISEKRYLOVITER option 492
PHNOISELORENTZ option 494
pivot
algorithm, selecting 496
change message 497
reference 498
PIVOT option 496
PIVREF option 498
PIVREL option 499
PIVTOL option 496, 500
.PKG command 216
PLIM option 666
plot
models 189
value calculation method 307
.PLOT command 660
in .ALTER block 28
pnp BJT models 189
POI keyword 20, 62, 276
pole-zero
(X0R, X0I) option 612
(X1R, X1I) option 613
(X2R, X2I) option 614
CSCAL option 340, 446
FSCAL option 382
GSCAL option 391
PZABS option 511
PZTOL option 512
RITOL option 525
pole-zero analysis
FMAX option 380
maximum iterations 430
polygon, defining 242
POST option 501
POST_VERSION option 504
POSTLVL option 503
POSTTOP option 506
.POWER command 218
power operating point table 203
.POWERDC command 220
power-dependent S parameter extraction 111
PP 164, 169
PP keyword 164, 176
.PRINT command 221
in .ALTER 28
printing
Jacobian data 487
printout
disabling 329, 482
suppressing 227
value calculation method 307
.PROBE command 225
PROBE option 507
propogation delays

measuring 157
with .MEASURE 155
.PROTECT command 227
protecting data 227
PSF option 508
PTDNOISE 230
overview 230
.PTDNOISE command 228
PTDNOISE with .MEASURE command 179
PURETP option 509
pushout bisection 180
PUTMEAS option 510
.PZ command 231
PZABS option 511
PZTOL option 512

R

RADIX scommand 628
RANDGEN option 513
reference temperature 269
RELH option 514
RELI option 432, 515
RELIN optimization parameter 191
RELMOS option 300, 432, 516
RELOUT optimization parameter 191
RELQ option 517
RELTOL option 334
RELTOLoption 518
RELV option 377, 453, 519
RELVAR option 520
RELVDC option 521
resistance 522
RESMIN option 522
RESULTS keyword 62
RF
.MEASURE PTDNOISE 179
RF commands
.SNNOISE 245, 249
RIN keyword 659
Rise 155
rise and fall times 157
RISE keyword 159
rise time
example 45
specify 634, 636
verify 45
RISETIME option 523
RITOL option 525
RMAX option 526
RMIN option 527
RMS keyword 164, 176
ROUT keyword 659
row/matrix ratio 499
RUNLVL option 528
N value effect on other options 676

S

.SAMPLE 232
.SAMPLE command 232
sampling noise 232
.SAVE command 233
SAVEHB option 533
SAVESNINIT option 534
SCALE option 535
SCALM option 536
SDA option 666
SEARCH option 537
SEED option 538
.SENS command 235
Setup 17
.SHAPE command 237
Defining Circles 239
Defining Polygons 240
Defining Rectangles 238
Defining Strip Polygons 242
Shooting Newton syntaxes 243
SIM_ACCURACY option 539
SIM_DSPF option 542
SIM_DSPF_ACTIVE option 544
SIM_DSPF_DELTAI option 540
SIM_DSPF_DELTAV option 541
SIM_DSPF_INSERROR option 545
SIM_DSPF_LUMPCAPS option 546
SIM_DSPF_MAX_ITER option 547
SIM_DSPF_RAIL option 548
SIM_DSPF_SCALEC option 549
SIM_DSPF_SCALER option 550
SIM_DSPF_VTOL option 551
SIM_LA option 553
SIM_LA_FREQ option 554

SIM_LA_MAXR option 555
SIM_LA_MINC option 556
SIM_LA_MINMODE option 557
SIM_LA_TIME option 558
SIM_LA_TOL option 559
SIM_ORDER option 560
SIM_OSC_DETECT_TOL option 561
SIM_POSTAT option 562
SIM_POSTDOWN option 563
SIM_POSTSCOPE option 564
SIM_POSTSKIP option 565
SIM_POSTTOP option 566
SIM_POWER_ANALYSIS option 567
SIM_POWER_TOP option 569
SIM_POWERDC_ACCURACY option 570
SIM_POWERDC_HSPICE option 571
SIM_POWERPOST option 572
SIM_POWERSTART option 573
SIM_POWERSTOP option 574
SIM_SPEF option 575
SIM_SPEF_ACTIVE option 576
SIM_SPEF_INSERROR option 577
SIM_SPEF_LUMPCAPS option 578
SIM_SPEF_MAX_ITER option 579
SIM_SPEF_PARVALUE option 580
SIM_SPEF_RAIL option 581
SIM_SPEF_SCALEC option 582
SIM_SPEF_SCALER option 583
SIM_SPEF_VTOL option 584
SIM_TG_THETA option 585
SIM_TG_TRAP option 586
SIM2 distortion measure 75
simulation
  accuracy 306, 448
  accuracy improvement 371
  multiple analyses, .ALTER command 28
  multiple runs 82
  reducing time 55, 313, 371, 420, 426, 587, 598
  results
    plotting 660–661
    printing 222
    specifying 153
  title 272
Simulation Runs 18
skew, parameters 139
slew rate
  verification 49
SLEW, .CHECK command 49
SLOPE command 630
SLOPETOL option 587
small-signal, DC sensitivity 236
.SN command 243
SNACCURACY option 588
.SNFT command 246
SNMAXITER option 589
.SNNOISE command 245, 249
.SNOSC command 251
.SNXF command 254
source
  AC sweep 19
  DC sweep 61
S-parameter, model type 189
SPICE
  compatibility
    AC output 307
    plot 666
SPMODEL option 590
START keyword 275
statements
  .AC 19
  .ACMATCH 22
  .ALIAS 26
  .ALTER 28, 69
  alter block 15
  .BIASCHK 31
  .CHECK EDGE 37
  .CHECK FALL 39
  .CHECK GLOBAL_LEVEL 40
  .CHECK HOLD 41
  .CHECK IRDROP 43
  .CHECK RISE 45
  .CHECK SETUP 47
  .CHECK SLEW 49
  .CONNECT 51
  .DATA 54
    external file 54
    inline 54
  .DC 61, 63
  .DCMATCH 66
  .DCVOLT 68, 128
  .DEL LIB 69
  .DISTO 74, 75
  .DOUT 76

.EBD 78
.ELSE 80
.ELSEIF 81
.END 82
.ENDDATA 83
.ENDIF 84
.ENDL 85, 138
.ENDS 86, 91
.ENV 87
.ENVFFT 88
.ENVOSC 90
.EOM 91
.FFT 92
.FOUR 98
.FSOPTIONS 99
.GLOBAL 102
.GRAPH 656, 657
.HB 103
.HBAC 107
.HBLIN 108
.HBLSP 110
.HBNOISE 112
.HBOSC 115
.HBXF 120
.HDL 121
.IBIS 124
.IC 68, 128
.ICM 130
.IF 132
.INCLUDE 80, 82, 132, 134, 234
.LAYERSTACK 135
.LIB 137, 138
  nesting 138
.LIN 140
.LOAD 144
.LPRINT 146
.MACRO 147
.MALIAS 149
.MATERIAL 151
.MEASURE 153, 455, 458
.MODEL 188
.MOSRA 194
.MOSRAPRINT 197
.NET 658
.NODESET 198
.NOISE 200
.OP 203, 204
.PARAM 207
.PAT 211
.PERIOD 627
.PHASENOISE 213
.PKG 216
.PLOT 660
.POWER 218
.POWERDC 220
.PRINT 221
.PROBE 225
.PROTECT 227
.PZ 231
.SAMPLE 232
.SAVE 233
.SENS 235
.SHAPE 237
.SNFT 246
.SNOSC 251
.SNXF 254
.STIM 258
.SUBCKT 263
.SURGE 266
.SWEEPBLOCK 267
.TEMP 269
.TF 271
.TITLE 272
.TRAN 273
.UNPROTECT 281
.VARIATION 282
.VEC 285
.WIDTH 661
.STATEYE command 256
STATFL option 591
statistical eye diagram analysis 256
statistics, listing 305
.STIM command 258
subcircuit commands 18
subcircuits
  calling 147, 264
  global versus local nodes 102
  names 147, 263
  node numbers 147, 263
  parameter 86, 91, 147, 263, 264
  printing path numbers 490
  test example 147, 264
.SUBCKT command 263
.SURGE command 266
sweep
  data 58, 458

frequency 21
inner 58
outer 58
SWEEP keyword 20, 62, 275
.SWEEPBLOCK command 267
SYMB option 592

T

Tabular Data section
time interval 627
TARG_SPEC 155
target specification 155
TDELAY command 632
TEMP
keyword 20, 62
model parameter 269
.TEMP (or) .TEMPERATURE command 269
temperature
AC sweep 19
DC sweep 61, 64
derating 269, 270
reference 269
terminal name, back annotation 317
.TF command 271
TFALL command 634
threshold voltage 76
TIC model parameter 657
time 203
See also CPU time
TIMERES option 593
timestep
algorithms 371
calculation for DVDT=3 381
changing size 517
control 381, 520, 598
maximum 419, 427, 526
minimum 420, 426, 527
reversal 303
transient analysis algorithm 448
.TITLE command 272
title for simulation 272
TNOM option 269, 596
TO keyword 164, 168, 174
TOL keyword 232
TOP keyword 233
.TRAN command 273
TRANFORHB option 597
transient analysis
Fourier analysis 98
initial conditions 68, 128
number of iterations 428
TRAP algorithm
See trapezoidal integration
TRCON option 667
TRIG keyword 155
TRIG_SPEC 155
trigger specification 155
TRISE command 634, 636
TRIZ command 638
TRTOL option 598
TSKIP command 639
TSTEP
multiplier 526, 527
option 526, 527
TUNIT command 640

U

UIC
parameter 68, 128
U-lement, transmission line model 189
.UNPROTECT command 281
UNWRAP option 599

V

-v argument
version information 11
VAMODEL option 600
variation block options
Monte Carlo 616
.VARIATION command 282
.VEC command 285
VEC commands
ENABLE 620
IDELAY 621
IO 623
ODELAY 624
OUT, OUTZ 626
PERIOD 627
RADIX 628
SLOPE 630
TDELAY 632
TFALL 634
TRISE 636

TRIZ 638
TSKIP 639
TUNIT 640
VIH 642
VIL 643
VNAME 644
VOH 646
VOL 648
VREF 650
VTH 652
VERIFY option 601
Verilog-A commands 18
version
determining 11
H9007 compatibility 664
VFLOOR option 602
Viewlogic graph data file 369
VIH command 642
VIL command 643
VNAME command 644
VNTOL option 377, 603
VOH command 646, 648
VOL command 648
voltage
initial conditions 68, 128
iteration-to-iteration change 370
logic high 642, 646
logic low 643
logic low threshold 648
maximum change 303
minimum
DC analysis 304
listing 602
transient analysis 302
operating point table 203
tolerance
MBYPASS multiplier 453
value for BYPASS 332
VOLTAGE keyword 203
VREF command 650
VREF statement 650
VTH command 652
VTH statement 652

W

WACC option 604
warnings
limiting repetitions 605
suppressing 479
WARNLIMIT option 605
WDELAYOPT option 606
WEIGHT keyword 165, 173
W-elements transmission line model 189
WHEN keyword 160
WHEN, using with .MEASURE 158
.WIDTH command 661
WINCLUDEGDIMAG option 608
WL option 609
WNFLAG option 610
WSF output data 663, 667

X

XDTEMP option 611
XGRID model parameter 657
XMAX model parameter 657
XMIN model parameter 657
XSCAL model parameter 658

Y

YGRID model parameter 657
YMAX parameter 173, 658
YMIN parameter 173, 658
YSCAL model parameter 658

Z

ZUKEN option 668, 669