

Index

- ABCD parameters
 - cascading of devices, 26
 - transmission line, 180
 - conversion
 - to s-parameters, 60
 - to Z-parameters, 24
 - distributed transmission line, 182
 - series impedance, 19
 - shunt impedance, 20
 - telegrapher's equations, 596–597
- ABCD2S() function, 60
- ABCDparameters
 - definition, 15
- absolute network device
 - Y-parameter, 14
 - Z-parameter, 11
- adaption (waveform), 429–433
 - example, 433–435
 - fractional delay, 429
 - overlapping portions, 430
 - summary, 432
 - trimming, 430–432
- balanced transmission line, 205
 - s-parameters, 207
- bandwidth limiting effects, 360, 361
- base class, 542
- base sample period, calculation property, 566
- base sample rate, calculation property, 566
- baud rate, 570
- behavioral model, 9
- blind equalizer adaption, 585
- block matrix solution, 103
- block weights matrices, 104
- boost, in virtual probing, 351
- cable, linear fit, 516
- calculation properties, 566–568
- calibration, *see* s-parameter measurement, calibration
- Calibration class, 477
 - error terms calculation methods, 478
 - hierarchy, 561
- calibration constants, *see* calibration standards
- calibration kit
 - constants and standards, *see* calibration standards
 - example, 485
 - Rosenberger 2.92 mm, 485
- calibration standards, 479–484
 - load, 482
 - offset, 479–481
 - offset loss, 484
 - open, 483
 - short, 482
- CalibrationConstants class, 480
 - hierarchy, 561
- CalibrationKit class, 486
 - hierarchy, 561
- CalibrationMeasurement class, hierarchy, 561
- CamelCase, 542
- canonical form, 90, 92, 104
- capacitor, 252, 553
- cascading of devices
 - ABCD parameters, 26
 - s-parameters, 69
 - example, 129
 - T-parameters, 69
- causality, 504–508, 569
 - enforcement, 455
- characteristic impedance, 183
 - real-valued, 184
- chirp z-transform, 398, 400, 402
 - CZT() function, 402
 - description, 401–403
 - graphical interpretation, 402
 - resampling with, 403
 - in SignalIntegrity.Lib package organization, 537
- circuit element, 8

- circuit topology, 512
- class, 541
- class inheritance, 541
- class instance, 541
- classic network device, 9
 - reason for using, 15
- common mode
 - definition, 195
 - due to imbalance, 199
 - probe, 573
 - radiated emissions, 199
 - s-parameters, 200
 - susceptibility, 199
- constraints matrix, 329
- container class, 542
- continuous-frequency response, 364
- continuous-time waveform, 359, 362
- conversion
 - ABCD to s-parameters, 60
 - ABCD to Z-parameters, 24
 - mixed-mode, 199
 - network parameters, 21, 22
 - reference impedance, 149
 - s- to ABCD parameters, 61
 - s- to T-parameters, 67
 - s- to Y-parameters, 59
 - s- to Z-parameters, 56
 - in `SignalIntegrity.Lib` package organization, 537
 - T- to s-parameters, 68
 - Y- to ABCD parameters, 24
 - Y- to s-parameters, 57
 - Z- to ABCD parameters, 23
 - Z- to s-parameters, 55
 - Z- to Y-parameters, 23
- conversion function
 - `ABCD2S()`, 60
 - `ReferenceImpedance()`, 149
 - `S2ABCD()`, 62
 - `S2T()`, 76
 - `S2Y()`, 58
 - `S2Z()`, 56
 - `Sp2Sw()`, 63
 - `Sw2Sp()`, 64
 - `T2S()`, 77
 - `Y2S()`, 58
 - `Z2S()`, 55
- convolution
 - definition, 357
 - discrete, 377
 - relationship to DFT, 377
 - time effect, 404–407
 - example, 405
- crosstalk term, 462
 - calibration method, 465
 - in error terms, 465
- current amplifier, 254, 554
 - four-port, 166, 611
 - three-port, 167, 612
 - two-port, 168, 613
- current controlled current source, 160, 253, 554
- current controlled voltage source, 161, 253, 554
- current sensor, 159
- current source
 - current controlled, 160
 - one-port, 153
 - terminated, 154
 - two-port, 152
 - voltage controlled, 162
- cursor tap, 580
- CZT, *see* chirp z-transform
- CZT() function, 402
- data hiding, 541
- DC point
 - extrapolation, 569–570
 - measurement, 487
- de-embedding
 - fixture, 287–289
 - internal nodes, 290–294
 - multiple unknowns, 300–302
 - symbolic example, 308
 - number of ports, 294–295
 - numeric example, 314–318
 - one-port, 283
 - example, 283
 - symbolic example, 304–306
 - two-port, 284–287, 298, 299
 - example, 284
 - symbolic example, 307
 - tip de-embedding, 289–290
- de-emphasis, 585
- Deembedder** class, 303
 - description, 303–304

- DeembedderNumeric** class, 313
 - description, 312
 - hierarchy, 546
- DeembedderNumericParser** class, 313
 - description, 312
 - hierarchy, 549
- DeembedderParser** class, 310
 - description, 310
 - symbolic example, 311
- DeembedderSymbolic** class, 305
 - description, 304
 - examples, 304–310
 - hierarchy, 547
- delay samples, 407
- dependent source
 - current–current, 160
 - current–voltage, 161
 - voltage–current, 162
 - voltage–voltage, 160
- dependent stimuli, 329, 332, 347
- derived class, 542
- Device** class, 219, 220
- device parser
 - class hierarchy, 551–556
 - UML diagram, 552
- device under test, 457
 - calculation, 474–475
- DeviceFactory** class, 249–251
 - MakeDevice()** method, 248
- DeviceParser** class, 248
 - description, 248–256
 - recognized amplifiers, 254
 - recognized calibration standards, 256
 - recognized dependent sources, 252
 - recognized devices, 253
 - recognized transmission lines, 255
- devices, in **SignalIntegrity.Lib** package
 - organization, 537
- DFT, *see* discrete Fourier transform
- differential mode
 - definition, 195
 - s-parameters, 200
- differential signaling, 194, 200
- directional coupler, 252, 553
 - in vector network analyzer, 457
- directivity term, in error terms, 466
- discrete Fourier transform
 - definition, 368
 - even K , 372
 - frequency content view, 370
 - frequency effect view, 369
 - interpretation, 370, 371
 - odd K , 373
 - padding, 383
 - padding and interpolation, 382–384
 - periodicity, 380–381
 - repetition and discreteness, 381
 - scaling, 378
 - explanation, 378–379
- discrete-frequency response, 358, 364
- discrete-time waveform, 360, 362
- dissipation factor, 186, 526
- downsampler, 416
- DUT, *see* device under test
- dynamic range, in the TDR, 494–498
- eigenvalues, 181
- eigenvectors, 181
- electrical length, 179
- emphasis, 585, 589
- end frequency, calculation property, 566–568
- equalizer, 578–580
 - blind adaption, 585
 - feed-forward, 586
 - fitting, 579–581
 - nonlinear, 584
 - inter-symbol interference reduction, 580–581
 - pulse response, 579
 - tapped delay line, 587
 - zero-forcing, 579–581
- EqualizerFitter** class, 584
 - description, 581–585
- error terms, 458
 - crosstalk term, 465
 - directivity term, 466
 - fixture structure, 459–462
 - forward-transmission term, 467
 - load-match term, 469
 - reverse-transmission term, 466
 - source-match term, 466
 - TDR, 495
 - twelve-term model, 459–462
 - VNA, 476
- ErrorTerms** class
 - crosstalk calibration method, 465
 - DUT calculation method, 476
 - EnforceReciprocity()** method, 503

- Fixture()** method, 462
 - hierarchy, 561
 - reflect calibration method, 466
 - thru calibration method, 471
 - transfer thru calibration method, 473
 - unknown thru calibration method, 472
- even mode, 205
 - impedance, 573–575
 - termination, 570
- EvenlySpacedFrequencyList** class, 385
 - hierarchy, 559
- external interference, 199
- eye diagram, 588–589
- EyePattern()** function, 588
 - description, 589
- feed-forward equalizer, 586
 - repetition in frequency, 587
- feedback amplifier
 - four-port, 621
 - two-port, 620
- filter
 - class
 - FirFilter**, 414
 - TransferMatrices**, 438
 - TransferMatricesProcessor**, 437
 - descriptor, *see* filter descriptor
 - fractional delay, 423–428
 - linear, 428–429
 - interpolator, 417
 - order in cascade, 435–436
 - unity upsample factor, 436
 - in **SignalIntegrity.Lib** package organization, 539
 - transfer matrices, processing, 436–439
- filter descriptor, 407–409
 - delay samples, 407
 - of impulse response, 411–412
 - of interpolator, 421
 - order in cascade, 435–436
 - unity upsample factor, 436
 - startup samples, 409
 - upsample factor, 407
- FilterDescriptor** class, 409
 - After()** method, 435
 - Before()** method, 435
 - hierarchy, 559
 - trimming methods, 431
- filtering, 414
- finite impulse response filter, 438
- FirFilter** class, 414
 - hierarchy, 557
 - relationship with **ImpulseResponse**, 559
- fit, in **SignalIntegrity.Lib** package organization, 539
- fixture de-embedding, 287–289
- forward-transmission term, in error terms, 467
- Fourier transform, 358
- fractional delay filter, 423–428
 - class
 - FractionalDelayFilterLinear**, 428
 - FractionalDelayFilterSinX**, 427
 - InterpolatorFractionalDelayFilterLinear**, 428
 - InterpolatorFractionalDelayFilterSinX**, 427
 - linear, 428–429
- FractionalDelayFilterLinear** class, 428
- FractionalDelayFilterSinX** class, 427
- frequency aliasing, 360
- frequency content, 361
- frequency domain, in **SignalIntegrity.Lib** package organization, 539
- frequency list, 384–386
- frequency points, calculation property, 566, 567
- frequency resolution
 - calculation property, 566
 - requirements, 568
- frequency response, 384–386
 - delay, 388–390
 - impulse response duality, 386–388
 - padding, 393–394
 - resampling, 395–398
 - with chirp z-transform, 401–403
 - example, 398–401
 - Nyquist rate point, 388–390
- frequency spacing, 366
- FrequencyContent** class, 375
 - description, 374
 - hierarchy, 557
 - relationship with **Waveform**, 559
 - Values()** method, 377
 - Waveform()** method, 377

FrequencyDomain class, 376, 384
 description, 384
 hierarchy, 557
FrequencyList class, 385
 relationship with **TimeDescriptor**, 559
FrequencyResponse class
 _DelayBy() method, 388
 _FractionalDelayTime() method, 388
 hierarchy, 557
 ImpulseResponse() method, 386
 _Pad() method, 393
 relationship with **ImpulseResponse**, 559
 Resample() method, 396
 ResampleCZT() method, 403
function
 ABCD2S(), 60
 CZT(), 402
 EyePattern(), 588
 description, 589
 MixedModeConverter(), 197
 MixedModeConverterVoltage(), 196
 Mutual(), 52
 Rat(), 397
 ReferenceImpedance(), 149
 ReferenceImpedanceTransformer(), 139
 S2ABCD(), 62
 S2T(), 76
 S2Y(), 58
 S2Z(), 56
 SeriesZ(), 49
 ShuntZ(), 51
 SinX(), 427
 Sp2Sw(), 63
 Sw2Sp(), 64
 T2S(), 77
 Tee(), 55
 TerminationZ(), 86
 TLineFourPort(), 189
 TLineFourPortLossless(), 189
 TLineTwoPort(), 186
 TLineTwoPortLossless(), 186
 TLineTwoPortRLGCAalytic(), 188
 TLineTwoPortRLGCApproximate(), 187
 Y2S(), 58
 ZOKHelper(), 36
 ZOKHelperPW(), 36
 Z2S(), 55
 ZeroForcingEqualizer(), 579–581

GenericFrequencyList class, 385
Gibb’s phenomenon, 568
Git, 534
GitHub, 534
GNU general purpose license, 534
graphical equation method, 122
ground, 252, 553

horizontal offset, 406, 407

ideal transformer, 177, 252, 553, 624
impedance
 characteristic of transmission line, 183
 series
 network parameters, 71
 port reference impedance, 138
 s-parameters, 49
 Y-parameters, 19
 series-shunt, network parameters, 72
 shunt
 ABCD parameters, 20
 network parameters, 72
 s-parameters, 51
 Y-parameters, 20
 Z-parameters, 19
 termination
 network parameters, 71
 Y-parameters, 20
 Z-parameters, 20
impedance profile, 440
 class
 ImpedanceProfile, 451
 ImpedanceProfileWaveform, 452
 PeeledLaunches, 453
 PeeledPortSParameters, 453
 peeling, 448–450
 from s-parameters, 445–448
 series distributed resistance, 455
 in **SignalIntegrity.Lib** package organi-
 zation, 539
 single port impedance, 441
 software, 450–455
 step response, 443–445
 time-domain reflectometer, 440–443
ImpedanceProfile class, 451
ImpedanceProfileWaveform class, 452
impulse response
 delay, 388–390
 filter descriptor of, 411–412

- frequency response duality, 386–388
- length, 569, 575–577
 - accounting for in simulation, 578
 - adjustment, 390–393
 - calculation property, 566
 - proper determination of, 568
 - requirement, 366
- negative time, 390–393
- padding, 393–394
- time sense, 367, 387, 392, 393
- trimming, 394
- impulse train response, 365
- ImpulseResponse** class
 - `DelayBy()` method, 389
 - `FirFilter()` method, 411
 - `_FractionalDelayTime()` method, 389
 - `FrequencyResponse()` method, 387
 - hierarchy, 557
 - `_Pad()` method, 394
 - relationship with **FirFilter**, 559
 - relationship with **FrequencyResponse**, 559
 - `Resample()` method, 398
 - `TrimToThreshold()` method, 395
- incident wave, 88
- independent stimuli, 329, 332
- inductor, 252, 553
 - mutual, 51, 252, 553
- inheritance, 541
- internal nodes, 103
- interpolation, 417–423
 - class
 - InterpolatorFractionalDelayFilter-Linear**, 428
 - InterpolatorFractionalDelayFilter-SinX**, 427
 - InterpolatorLinear**, 428
 - InterpolatorSinX**, 427
 - filter descriptor, 421
 - filter options, 420
 - pulse responses, 422
 - linear, 420
 - sinc, 418
- InterpolatorFractionalDelayFilter-Linear** class, 428
- InterpolatorFractionalDelayFilterSinX** class, 427
- InterpolatorLinear** class, 428
- InterpolatorSinX** class, 427
- inverse, of matrix, 600–601
- Jacobian matrix, 520, 528
- known thru standard, 467
- Laplace transform, 358
- LTeX**, 230–232
- least mean-squared error, 513–515
- Levenberg–Marquardt algorithm, 521
- LevMar** class, 522
 - description, 521–524
 - `Solve()` method, 523
- license, of *SignalIntegrity* software, 534
- linear equations, 513–515
- linear feed-forward equalizer, 586
- linear fit
 - cable, 516
 - least mean-squared error, 513–515
- linear model extraction example, 515–517
- linear regression, 513–515
- linear simulations, 357
- load-match term, in error terms, 469
- load standard, 256, 482, 555
- LoadStandard** class, 482
 - hierarchy, 561
- loopback, 96, 98, 114, 116, 117, 119, 123, 124
- loss tangent, 186, 526
- match, 87
- matrices
 - block, 602–607
 - block weights, 104
 - inverse, 600–601
 - block, 602–607
 - multiplication order, 607
 - power of, 180
 - pseudo-inverse, 600–601
 - in systems of equations, 598–600
 - terminology, 593–594
- meander, 507
- measurement, *see also* s-parameter measurement
 - in **SignalIntegrity.Lib** package organization, 539
- measurement conditions
 - in s-parameter measurements, 458
 - in virtual probing, 332

mixed-mode s-parameters - advice for using, 200 - circuit example, 573 - conversion between single-ended s-parameters, 199 - converter, 252 - standard, 197 - voltage, 196 - device, 553 - equivalent circuits, 571 - function <code>MixedModeConverter()</code>, 197 <code>MixedModeConverterVoltage()</code>, 196 - pi termination, 629 - tee termination, 628 - termination, 209, 630 - example, 572 - thru response, 575–576	node names, 90
<code>MixedModeConverter()</code> function, 197	node ordering, 104
<code>MixedModeConverterVoltage()</code> function, 196	node removal, 109 - example, 118 - graphical equation method, 122 - rules, 110 - for single node, 111
mode conversion, 200	node vector, 89, 91
model - behavioral, 9 - topological, 8	nonlinear fit, 517
model extraction - class <code>LevMar</code>, 522, 523 <code>RLGCFitter</code>, 525, 526 - linear, example, 516 - nonlinear - example, 528–531 - Levenberg–Marquardt, 521 - Newton’s method, 517–521	normalization factor - power, 34–37 - terminology, 594 - transformer, 141
model fitting, 512	number of points, time descriptor, 407
Moore–Penrose pseudo-inverse, 601	numeric system description, UML diagram, 545
multiple port connections, 87	Nyquist rate, 361, 370, 372, 374
<code>Mutual()</code> function, 52	
mutual inductance, 51, 252, 553	odd mode, 206 - impedance, 573 - termination, 570
namespace, 541	offset - calibration standard, 479–481 - loss, 484
netlist, 217	Offset class, 481
network device - absolute, 10 - classic, 9	open, 252, 553
network parameters - of circuits, 16 - conversion, 21, 22	open standard, 256, 483, 555
Newton’s method, 517–521	OpenStandard class, 483 hierarchy, 561
	operational amplifier, 175, 254, 554, 623
	oscilloscope, 358
	output condition, in virtual probing, 332
	overconstrained fit, 513
	overconstrained systems solution, 601–602
	overconstrained thru measurement, 470
	package, 541
	padding - effect in other domain, 383 - and interpolation, 382–384
	PAM-4 - signaling, 570 - waveform, 582
	ParserArgs class, 257 description, 257
	ParserDevice class, 249
	parsers class hierarchy, 549–551

- in **SignalIntegrity.Lib** package organization, 539
- UML diagram, 550
- passivity, 498–500, 569
- PeeledLaunches** class, 453
- PeeledPortSParameters** class, 453
- peeling, impedance profile, 448–450
- periodicity
 - in frequency domain, 368
 - in time domain, 362
- permutation, 94, 103
 - column, 94
 - equation reordering, 95
- pi termination, 629
- Port** class, 219, 220
- port condition, 10
- port connections, 84
- post-cursor tap, 580, 585
- power waves, 29
 - conversion to waves, 40
 - definitions, 39
 - helper function, 36
 - multi-port definitions, 39
 - s-parameters, 62
- power, of matrix, 180
- PRBS, *see* pseudo-random bit sequence
- pre-cursor tap, 580, 585
- primary, of transformer, 176
- private member, of class, 541
- probe compensation, 331, 344
- probe loading, 331
- project file format, 562–564
- propagation constant, 183
- properties, of class member, 542
- pseudo-inverse, of matrix, 600–601
- pseudo-random bit sequence, 581–582
 - generator, pseudo, 582
 - waveform, 582
- pseudo-wave, 30
- pulse response, 577–578
 - equalized, 579
- PulseWaveform** class, 413
- PyPI, *see* Python Packaging Index
- Python Packaging Index, 534
- quadratic convergence, 519
- radiated emissions, 199
- range reduction, 519
- Rat()** function, 396–397
 - in **SignalIntegrity.Lib** package organization, 539
- reciprocity, 501–504
- reference impedance, 569, 594
 - function
 - ReferenceImpedance()**, 149
 - ReferenceImpedanceTransformer()**, 139
 - proper use of, 135
 - in s-parameter file, 81
 - transformation
 - equations, 150
 - tip de-embedding, 149–151
 - tip embedding, 146–148
 - transformer, 138
 - transmission line, 184
 - wave measurement, 143
 - with normalization, 143
- ReferenceImpedance()** function, 149
- ReferenceImpedanceTransformer()** function, 139
- reflect calibration, 466–467
- ReflectCalibrationMeasurement** class,
 - hierarchy, 561
- reflected wave, 88
- resampling, 569
 - Nyquist rate point, 388–390
- resistor, 8, 252, 553
 - series, 553
 - ABCD parameters, 19
 - Z-parameters, 17
 - shunt, 553
- resonance, effect on impulse response
 - length, 568
- response, 357
- reversal property, of waves, 31
- reverse-transmission term, in error terms, 466
- risetime, transmission line, 208
- RLGC transmission line model, 200–209
 - differential, 201
 - uncoupled, 204
 - model fit, 524–528
 - example, 528–531
 - single-ended, 178
- RLGCFitter** class, 525, 526
 - description, 524–528
- Rosenberger calibration constants, 485

- s-parameter file, 553
 - in device parser, 252
- s-parameter measurement
 - calibration, 462–470
 - directivity term, 466
 - error terms, of TDR, 495
 - error terms, of VNA, 476
 - forward-transmission term, 467
 - load-match term, 469
 - overconstrained thru calibration, 470
 - reflect calibration, 466–467
 - reverse-transmission term, 466
 - source-match term, 466
 - summary, 475–479
 - thru calibration, 467–470
 - transfer thru calibration, 473–474
 - unknown thru calibration, 471–472
 - Calibration** class, 477, 478
 - class hierarchy, 561
 - conditions, 458
 - device under test calculation, 474–475
 - ErrorTerms** class
 - crosstalk calibration method, 465
 - DUT calculation method, 476
 - Fixture()** method, 462
 - thru calibration method, 471
 - transfer thru calibration method, 473
 - unknown thru calibration method, 472
 - methods, 457
 - time-domain reflectometer, 487–492
 - UML diagram, 560
- s-parameter network
 - device connection, 83
 - multiple, 87
- s-parameters
 - cascading of devices, 69
 - causality, 504–508, 569
 - circuit node, 54
 - class
 - SParameterFile**, 79
 - SParameters**, 80
 - TDRWaveformToSParameterConverter**, 489–491
 - conversion
 - to ABCD parameters, 61
 - from power wave based, 63
 - to power wave based, 64
 - to T-parameters, 67
 - to Y-parameters, 59
 - to Z-parameters, 56
 - DC point
 - extrapolation, 569–570
 - measurement, 487
 - definition, 41
 - file format, 77
 - frequency spacing requirements, 368
 - function
 - MixedModeConverter()**, 197
 - MixedModeConverterVoltage()**, 196
 - Mutual()**, 52
 - ReferenceImpedanceTransformer()**, 139
 - SeriesZ()**, 49
 - ShuntZ()**, 51
 - Tee()**, 55
 - TerminationZ()**, 86
 - TLineFourPort()**, 189
 - TLineFourPortLossless()**, 189
 - TLineTwoPort()**, 186
 - TLineTwoPortLossless()**, 186
 - TLineTwoPortRLGCAalytic()**, 188
 - TLineTwoPortRLGCApproximate()**, 187
 - ideal termination, 86
 - identity section, 70
 - impedance profile, 445–448
 - measurement, *see* s-parameter measurement
 - method for determining, 46
 - mutual inductance, 53
 - open, 86
 - passivity, 498–500, 569
 - power wave based, 62
 - reciprocity, 501–504
 - series impedance, 49, 71
 - series-shunt impedance, 72
 - short, 86
 - shunt impedance, 51, 71, 72
 - signal-flow representation, 83
 - in **SignalIntegrity.Lib** package organization, 539
 - symbolic, 233
 - examples, 235–238
 - output types, 239, 240
 - symmetry, 501
 - of systems, 98
 - block matrix solution, 103
 - tee, 54
 - transmission line, 183

- balanced, 207
 - uncoupled, 204
- wire, 49
 - port reference impedance, 138
- S2ABCD()** function, 62
- S2T()** function, 76
- S2Y()** function, 58
- S2Z()** function, 56
- sample period, 366
- sample rate, time descriptor, 407
- sampling, 360
- schematic, 217
- secondary, 176
- SeriesZ()** function, 49
- short-open-load-reciprocal calibration, 471–472
- short-open-load-thru calibration, 462–470
- short standard, 256, 482, 555
- ShortStandard** class, 483
 - hierarchy, 561
- ShuntZ()** function, 51
- signal-flow diagram
 - loopback, 96
 - of s-parameter device, 83
 - of s-parameter system, 87
 - from system equation, 91
- SignalIntegrity* software, 534
- SignalIntegrityApp*, 562
 - embedded projects, 572
 - equalization example, 570–589
 - project file format, 562–564
 - user manual, 562
- SignalIntegrityAppHeadless** class, 562, 565–566
- SignalIntegrity.Lib**, 537
 - import, 539
- simulation
 - definition, 260
 - example, 272–281
 - linear, 357
- Simulator** class, 264, 265
 - description, 264–266
- SimulatorNumeric** class, 271
 - description, 271–272
 - hierarchy, 546
- SimulatorNumericParser** class, 271
 - hierarchy, 549
- SimulatorParser** class, 269
 - description, 266–271
 - example, 270
- SimulatorSymbolic** class, 268
 - description, 266
 - example, 267
 - hierarchy, 547
- sinc interpolator, 418
- SineWaveform** class, 413
- SinX()** function, 427
- skin effect, 186, 526
- SOLR, *see* short-open-load-reciprocal calibration
- SOLT, *see* short-open-load-thru calibration
- source-match term, in error terms, 466
- Sp2Sw()** function, 63
- SParameterFile** class, 79
 - description, 77–81
- SParameterManipulation** class
 - EnforceCausality()** method, 505
 - EnforcePassivity()** method, 500
 - EnforceReciprocity()** method, 501
- SParameters** class, 80
 - description, 77–81
 - hierarchy, 557, 561
 - Resample()** method, 399
- spectral density, 374
- splines, in **SignalIntegrity.Lib** package
 - organization, 539
- startup samples, 407, 409
 - removal, 414
- step response, 568
 - impedance profile, 443–445
- StepWaveform** class, 413
- stim, 345–347
 - stimdef*, 329, 347
 - property, 335
- stimuli, 90
- stimulus vector, 89, 91, 104
- subcircuit, 252, 256
 - in **SignalIntegrity.Lib** package organization, 539
- SubCircuit** class, 256
 - description, 256–259
 - example, 258
- superposition, 358
- Sw2Sp()** function, 64
- symbolic
 - device solutions, 608–629

- devices, 547
- s-parameter solutions, 233
 - examples, 235–238
 - output types, 239, 240
- in `SignalIntegrity.Lib` package organization, 539
- Symbolic** class, 229, 230
 - hierarchy, 549
- symbolic system description, UML diagram, 548
- symmetry, 501
- system characteristics matrix, 89, 92, 98
- system description, 217
 - class hierarchy, 542–544
 - example, 226
 - numeric
 - class hierarchy, 544–547
 - UML diagram, 545
 - parsers
 - class hierarchy, 549–551
 - UML diagram, 550
 - in `SignalIntegrity.Lib` package organization, 539
 - symbolic
 - class hierarchy, 547–549
 - UML diagram, 548
 - UML diagram, 543
- system equation, 89
- system identification, 512
- system of equations, 88, 598–600
- system reduction, 109
- SystemDescription** class, 221, 222
 - `AddDevice()` method, 221
 - `AddPort()` method, 221, 225
 - class structure, 218–225
 - `ConnectDevicePort()` method, 221–223
 - hierarchy, 544
 - `NodeVector()` method, 224
 - `StimulusVector()` method, 224
 - `WeightsMatrix()` method, 223–225
- SystemDescriptionParser** class, 241
 - description, 240–242, 251–255
 - examples, 242–245
 - hierarchy, 544
- SystemDescriptionSymbolic** class, 233
 - description, 233–235
 - example, 234
 - hierarchy, 544
- SystemSParameters** class, 224

- SystemSParametersNumeric** class, 247
 - description, 245–256
 - hierarchy, 544, 546
- SystemSParametersNumericParser**
 - class, 248
 - description, 255–256
 - hierarchy, 544, 549
- SystemSParametersSymbolic** class, 236
 - hierarchy, 547
 - output types, 239
 - description, 240
- T-parameters
 - advanced multi-port, 73
 - cascading of devices, 69
 - conversion to s-parameters, 68
 - definition, 65
 - identity section, 69
- `T2S()` function, 77
- tapped delay line equalizer, 587
- TDR, *see* time-domain reflectometer
- TDRWaveformToSParameter-Converter** class, 489
 - `Convert()` method, 490
 - explanation, 489–491
 - hierarchy, 561
- tee, 252
 - in voltage sense element, 158
 - multiple device port connections, 87
 - s-parameters of, 54
 - termination, 628
- `Tee()` function, 55
- telegrapher’s equations, 178, 596, 597
 - ABCD parameters, 596–597
 - derivation, 595–596
- termination, 20, 553
 - mixed-mode, 209, 210, 630
 - pi, 212, 629
 - s-parameters, 86
 - tee, 211, 628
- `TerminationZ()` function, 86
- Thévenin equivalent source, 156
- thru, 252, 553
- thru calibration, 467–470
 - overconstrained, 470
 - unknown, 471–472
- thru standard, 256, 555
- ThruCalibrationMeasurement** class,
 - hierarchy, 561

- ThruStandard** class, 482
 - hierarchy, 561
- time aliasing, 364
- time descriptor, 407
 - division
 - by filter descriptor, 410
 - by time descriptor, 410
 - horizontal offset, 407
 - multiplication with filter descriptor, 409
 - number of points, 407
 - sample rate, 407
- time domain, 568
 - in **SignalIntegrity.Lib** package organization, 539
- time interleaving, 415
- time invariance, 358
- time points, calculation property, 566
- time sense, of impulse response, 367, 387, 392, 393
- time-domain reflectometer
 - class
 - TDRWaveformToSParameterConverter**, 489, 490
 - TDRWaveformToSParameterConverter** explanation, 489–491
 - comparison with vector network analyzer, 487
 - impedance profile, 440–443
 - s-parameter measurement, 487–492
 - calibration, 492–494
 - dynamic range, 494–498
 - error terms, 495
 - wavelet denoising, 508–511
- TimeDescriptor** class, 408
 - hierarchy, 557, 559
 - relationship with **FrequencyList**, 559
- tip de-embedding, 289–290
 - reference impedance transformation, 149–151
- tip embedding, 108, 109
 - reference impedance transformation, 146–148
- TLineDifferentialRLGC** class, 209
- TLineDifferentialRLGCApproximate** class, 201
- TLineDifferentialRLGCBalanced** class, 207
- TLineDifferentialRLGCUncoupled** class, 204
- TLineFourPort()** function, 189
- TLineFourPortLossless()** function, 189
- TLineLossless** class, 187
- TLineTwoPort()** function, 186
- TLineTwoPortLossless()** function, 186
- TLineTwoPortRLGC** class, 187
- TLineTwoPortRLGCAalytic()** function, 188
- TLineTwoPortRLGCApproximate()** function, 187
- topological model, 8
- topology, 512
- Touchstone format, 77
- transconductance amplifier, 254, 554
 - four-port, 169, 614
 - three-port, 170, 615
 - two-port, 171, 616
- transfer matrices
 - frequency response example, 275
 - impulse response example, 276
 - processing, 436–439
- transfer thru calibration, 473–474
- TransferMatrices** class, 272, 438
 - hierarchy, 557
- TransferMatricesProcessor** class, 437
- transformer, 252, 553
 - ideal, 177, 624
- transient simulation, 357
- transistor, 176, 627
- transmission line, 255, 554
 - balanced, 205
 - s-parameters, 207
- class
 - TLineDifferentialRLGC**, 209
 - TLineDifferentialRLGCApproximate**, 201
 - TLineDifferentialRLGCBalanced**, 207
 - TLineDifferentialRLGCUncoupled**, 204
 - TLineLossless**, 187
 - TLineTwoPortRLGC**, 187
- differential, 200, 201
 - even-mode analysis, 205
 - odd-mode analysis, 206
 - software model, 209
 - uncoupled, 204
- energy storage, 194
- four-port software models, 188
- function

- `TLineFourPort()`, 189
- `TLineFourPortLossless()`, 189
- `TLineTwoPort()`, 186
- `TLineTwoPortLossless()`, 186
- `TLineTwoPortRLGCAalytic()`, 188
- `TLineTwoPortRLGCApproximate()`, 187
- risetime, 208
- simulation example, 191
- two-port software models, 186
- transresistance amplifier, 254, 554
 - four-port, 172, 617
 - three-port, 173, 618
 - two-port, 174, 619
- traveling wave, 30
- turns ratio, 176
- twelve-term error model, 459–462
- UML diagram, *see* universal modeling language diagram
- unittest, 537
- universal modeling language diagram, 217, 541–542
 - device parser, 552
 - measurement, 560
 - numeric system description solution, 545
 - parsers, 550
 - symbolic system description solution, 548
 - system description, 543
 - waveform processing, 558
- unknown thru measurement, 471–472
- upsample factor, 407
- upsampling, 415–417
- user sample period, calculation property, 566
- user sample rate, calculation property, 566, 567
- vector network analyzer, 358
 - calibration, *see* s-parameter measurement, calibration
 - comparison with time-domain reflectometer, 487
- virtual circuit, 332
- virtual probing
 - concept, 319
 - degrees of freedom, 327
 - dependent stimuli, 329
 - description, 319–323
 - general equation, 329–331
 - with multiple inputs/outputs, 323–326
 - requirements, 347
 - stimdef*, 329
 - virtual circuit, 331
- VirtualProbe** class, 333
 - description, 334
- VirtualProbeNumeric** class, 343
 - description, 342–344
 - hierarchy, 546
- VirtualProbeNumericParser** class, 343
 - example, 344–353
 - hierarchy, 549
- VirtualProbeParser** class, 338
 - description, 337–339
 - `_ProcessLines()` method, 338–339
 - symbolic example, 339–342
- VirtualProbeSymbolic** class, 334
 - description, 335
 - examples, 335–337
 - hierarchy, 547
- VNA, *see* vector network analyzer
- voltage amplifier, 254, 554
 - four-port, 163, 608
 - three-port, 164, 609
 - two-port, 165, 610
- voltage controlled current source, 162, 253, 554
- voltage controlled voltage source, 160, 253, 553
- voltage extraction matrix, 324, 330
 - definition, 321
- voltage sensor
 - absolute, 157
 - differential, 158
- voltage source
 - current controlled, 161
 - terminated, 157
 - two-port, 155
 - voltage controlled, 160
- wave
 - conversion to power wave, 40
 - definitions, 37
 - helper function, 36
 - incident, 88
 - multi-port definitions, 38
 - power relationship, 34–37
 - power wave, 29
 - multi-port definitions, 39

- pseudo-wave, 30
- reflected, 88
- required form, 34
- reversal property, 31
- s-parameter requirement, 33
- traveling wave, 30
- waveform
 - adaption, 429–433
 - example, 433–435
 - fractional delay, 429
 - overlapping portions, 430
 - summary, 432
 - trimming, 430–432
 - class
 - ImpedanceProfileWaveform**, 452
 - PulseWaveform**, 413
 - SineWaveform**, 413
 - StepWaveform**, 413
 - Waveform**, 412
 - downsampler, 416
 - filtering, 414
 - overlapping portions, 430
 - processing
 - class hierarchy, 557–561
 - UML diagram, 558
 - in **SignalIntegrity.Lib** package organization, 539
 - time descriptor, *see* time descriptor
 - trimming, 430–432
 - upsampling, 415–417
- Waveform** class, 412
 - Adapt()** method, 433
 - file read/write methods, 413
 - hierarchy, 557
 - multiplication method, 414
 - relationship with **FrequencyContent**, 559
- WaveformProcessor** class
 - hierarchy, 557
- WaveformTrimmer** class, 431
- wavelet
 - Daubechies, 510
 - denoising, 508–511
 - in **SignalIntegrity.Lib** package organization, 539
- Wavelet** class, 510
- WaveletDaubechies4** class, 510
- WaveletDenoiser** class, 511
- WavePulser 40iX, 314, 509
- weights matrix, 89–92, 96, 104
- wire
 - s-parameters, 49
 - port reference impedance, 138
 - transmission line, 185
- XtalkCalibrationMeasurement** class,
 - hierarchy, 561
- Y-parameters
 - cascading of devices, 28
 - conversion
 - to ABCD parameters, 24
 - to s-parameters, 57
 - definition, 13
 - mutual inductance, 53
 - series impedance, 19
 - shunt impedance, 20
 - termination, 20
- Y2S() function, 58
- Z-parameters, 9
 - cascading of devices, 27
 - conversion
 - to ABCD parameters, 23
 - to s-parameters, 55
 - to Y-parameters, 23
 - definition, 12
 - series impedance, 17
 - shunt impedance, 19
 - termination, 20
- z-transform, 358
- Z0 vs Z₀, 594
- Z0KHelper() function, 36
- Z0KHelperPW() function, 36
- Z2S() function, 55
- zero padding, 383
- zero-forcing equalizer, 579–581
- ZeroForcingEqualizer() function, 579–581