

List of Examples

Example 1-1	Driving the APB pins	17
Example 1-2	A task to drive the APB pins	18
Example 1-3	Low-level Verilog test	18
Example 1-4	Basic transactor code	22
Example 2-1	Using the logic type	28
Example 2-2	Signed data types	28
Example 2-3	Checking for four-state values	29
Example 2-4	Declaring fixed-size arrays	29
Example 2-5	Declaring and using multidimensional arrays	29
Example 2-6	Unpacked array declarations	30
Example 2-7	Initializing an array	30
Example 2-8	Using arrays with for and foreach loops	31
Example 2-9	Initialize and step through a multidimensional array	31
Example 2-10	Output from printing multidimensional array values	31
Example 2-11	Array copy and compare operations	32
Example 2-12	Using word and bit subscripts together	33
Example 2-13	Packed array declaration and usage	33
Example 2-14	Declaration for mixed packed/unpacked array	34
Example 2-15	Using dynamic arrays	35
Example 2-16	Using a dynamic array for an uncounted list	35
Example 2-17	Queue operations	36
Example 2-18	Declaring, initializing, and using associative arrays	38
Example 2-19	Using an associative array with a string index	39
Example 2-20	Creating the sum of an array	40
Example 2-21	Array locator methods: min, max, unique	41
Example 2-22	Array locator methods: find	41

Example 2-23	Array locator methods	42
Example 2-24	User-defined type-macro in Verilog	45
Example 2-25	User-defined type in SystemVerilog	45
Example 2-26	Definition of uint	45
Example 2-27	Creating a single pixel type	46
Example 2-28	The pixel struct	46
Example 2-29	Using typedef to create a union	47
Example 2-30	Packed structure	47
Example 2-31	A simple enumerated type	48
Example 2-32	Enumerated types	48
Example 2-33	Specifying enumerated values	48
Example 2-34	Incorrectly specifying enumerated values	49
Example 2-35	Correctly specifying enumerated values	49
Example 2-36	Stepping through all enumerated members	50
Example 2-37	Assignments between integers and enumerated types	50
Example 2-38	Declaring a const variable	51
Example 2-39	String methods	52
Example 2-40	Expression width depends on context	53
Example 2-41	Disabling implicit nets with 'default_nettype none	53
Example 3-1	New procedural statements and operators	55
Example 3-2	Using break and continue while reading a file	56
Example 3-3	Ignoring a function's return value	56
Example 3-4	Void function for debug	57
Example 3-5	Simple task without begin...end	57
Example 3-6	Verilog-1995 routine arguments	58
Example 3-7	C-style routine arguments	58
Example 3-8	Verbose Verilog-style routine arguments	58
Example 3-9	Routine arguments with sticky types	58
Example 3-10	Passing arrays using ref and const	59
Example 3-11	Using ref across threads	60
Example 3-12	Function with default argument values	61
Example 3-13	Using default argument values	61
Example 3-14	Original task header	61
Example 3-15	Task header with additional array argument	61
Example 3-16	Task header with additional array argument	62
Example 3-17	Return in a task	62
Example 3-18	Return in a function	62
Example 3-19	Specifying automatic storage in program blocks	63
Example 3-20	Static initialization bug	64

Example 3-21	Static initialization fix: use automatic	64
Example 3-22	Time literals and \$timeformat	65
Example 4-1	Simple BusTran class	69
Example 4-2	Declaring and using a handle	71
Example 4-3	Simple use-defined new function	72
Example 4-4	A new function with arguments	72
Example 4-5	Calling the right new function	73
Example 4-6	Allocating multiple objects	74
Example 4-7	Creating multiple objects	75
Example 4-8	Using variables and routines in an object	76
Example 4-9	Class with a static variable	77
Example 4-10	Initializing a static variable in a task	78
Example 4-11	Routines in the class	79
Example 4-12	Out-of-block routine declarations	80
Example 4-13	Out-of-body task missing class name	81
Example 4-14	Name scope	82
Example 4-15	Class uses wrong variable	83
Example 4-16	Using this to refer to class variable	83
Example 4-17	Bug using shared program variable	84
Example 4-18	Statistics class declaration	85
Example 4-19	Encapsulating the Statistics class	86
Example 4-20	Using a typedef class statement	87
Example 4-21	Passing objects	88
Example 4-22	Bad packet creator task, missing ref on handle	89
Example 4-23	Good packet creator task with ref on handle	89
Example 4-24	Bad generator creates only one object	90
Example 4-25	Good generator creates many objects	90
Example 4-26	Using an array of handles	91
Example 4-27	Copying a simple class with new	92
Example 4-28	Copying a complex class with new	92
Example 4-29	Simple class with copy function	93
Example 4-30	Using copy function	94
Example 4-31	Complex class with deep copy function	94
Example 4-32	Basic Transactor	97
Example 5-1	Arbiter model using ports	101
Example 5-2	Testbench using ports	101
Example 5-3	Top-level netlist without an interface	102
Example 5-4	Simple interface for arbiter	103
Example 5-5	Top module using a simple arbiter interface	103

Example 5-6	Testbench using a simple arbiter interface	104
Example 5-7	Arbiter using a simple interface	104
Example 5-8	Connecting an interface to a module that uses ports	105
Example 5-9	Interface with modports	105
Example 5-10	Arbiter model with interface using modports	106
Example 5-11	Testbench with interface using modports	106
Example 5-12	Arbiter model with interface using modports	107
Example 5-13	Interface with a clocking block	109
Example 5-14	Race condition between testbench and design	111
Example 5-15	Testbench using interface with clocking block	113
Example 5-16	Signal synchronization	115
Example 5-17	Synchronous interface sample and module drive	115
Example 5-18	Testbench using interface with clocking block	116
Example 5-19	Interface signal drive	117
Example 5-20	Driving a synchronous interface	117
Example 5-21	Interface signal drive	118
Example 5-22	Bidirectional signals in a program and interface	119
Example 5-23	Bad clock generator in program block	120
Example 5-24	Good clock generator in module	121
Example 5-25	Top module using a simple arbiter interface	121
Example 5-26	Top-level scope for arbiter design	122
Example 5-27	Cross-module references with \$root	123
Example 5-28	Checking a signal with an if-statement	124
Example 5-29	Simple procedural assertion	124
Example 5-30	Error from failed procedural assertion	125
Example 5-31	Creating a custom error message in a procedural assertion	125
Example 5-32	Error from failed procedural assertion	125
Example 5-33	Creating a custom error message	126
Example 5-34	Concurrent assertion to check for X/Z	126
Example 5-35	ATM router model header without an interface	128
Example 5-36	Top-level netlist without an interface	129
Example 5-37	Testbench using ports	130
Example 5-38	Rx interface	132
Example 5-39	Tx interface	132
Example 5-40	ATM router model with interface using modports	133
Example 5-41	Top-level netlist with interface	133
Example 5-42	Testbench using interface with clocking block	134
Example 6-1	Simple random class	139
Example 6-2	Constraint without random variables	141

Example 6-3	Constrained-random class	142
Example 6-4	Constrain variables to be in a fixed order	142
Example 6-5	Random sets of values	143
Example 6-6	Inverted random set constraint	143
Example 6-7	Inverted random set constraint	143
Example 6-8	Choosing from an array of possible values	144
Example 6-9	Using randc to chose array values in random order	145
Example 6-10	Weighted random distribution with dist	146
Example 6-11	Dynamically changing distribution weights	146
Example 6-12	Bidirectional constraint	147
Example 6-13	Constraint block with implication operator	148
Example 6-14	Constraint block with if-else operator	148
Example 6-15	Expensive constraint with mod and unsized variable	149
Example 6-16	Efficient constraint with bit extract	149
Example 6-17	Class Unconstrained	149
Example 6-18	Class with implication	150
Example 6-19	Class with implication and constraint	151
Example 6-20	Class with implication and solve...before	152
Example 6-21	Using constraint_mode	154
Example 6-22	Checking write length with a valid constraint	155
Example 6-23	The randomize() with statement	156
Example 6-24	Building a bathtub distribution	157
Example 6-25	Constraint with a variable bound	159
Example 6-26	dist constraint with variable weights	159
Example 6-27	rand_mode disables randomization of variables	160
Example 6-28	Using the implication constraint as a case statement	161
Example 6-29	Turning constraints on and off with constraint_mode	162
Example 6-30	Class with an external constraint	163
Example 6-31	Program defining external constraint	163
Example 6-32	Signed variables cause randomization problems	164
Example 6-33	Randomizing unsigned 32-bit variables	164
Example 6-34	Randomizing unsigned 8-bit variables	165
Example 6-35	Constraining dynamic array size	165
Example 6-36	Random strobe pattern class	166
Example 6-37	Using random strobe pattern class	167
Example 6-38	First attempt at sum constraint: bad_sum1	167
Example 6-39	Program to try constraint with array sum	168
Example 6-40	Output from bad_sum1	168
Example 6-41	Second attempt at sum constraint: bad_sum2	168

Example 6-42	Output from bad_sum2	168
Example 6-43	Third attempt at sum constraint: bad_sum3	169
Example 6-44	Output from bad_sum3	169
Example 6-45	Fourth attempt at sum_constraint: bad_sum4	169
Example 6-46	Output from bad_sum4	169
Example 6-47	Simple foreach constraint: good_sum5	170
Example 6-48	Output from good_sum5	170
Example 6-49	Creating ascending array values with foreach	170
Example 6-50	UniqueArray class	171
Example 6-51	Unique value generator	172
Example 6-52	Using the UniqueArray class	172
Example 6-53	Command generator using randsequence	173
Example 6-54	Random control with randcase and \$urandom_range	175
Example 6-55	Equivalent constrained class	176
Example 6-56	Creating a decision tree with randcase	177
Example 6-57	Simple pseudorandom number generator	178
Example 6-58	Ethernet switch configuration class	180
Example 6-59	Building environment with random configuration	181
Example 6-60	Simple test using random configuration	182
Example 6-61	Simple test that overrides random configuration	182
Example 7-1	Interaction of begin...end and fork...join	185
Example 7-2	Output from begin...end and fork...join	185
Example 7-3	Fork...join_none code	186
Example 7-4	Fork...join_none output	186
Example 7-5	Fork...join_any code	187
Example 7-6	Output from fork...join_any	187
Example 7-7	Generator class with a run task	188
Example 7-8	Dynamic thread creation	189
Example 7-9	Bad fork...join_none inside a loop	190
Example 7-10	Execution of bad fork...join_none inside a loop	190
Example 7-11	Automatic variables in a fork...join_none	191
Example 7-12	Steps in executing automatic variable code	191
Example 7-13	Disabling a thread	192
Example 7-14	Limiting the scope of a disable fork	193
Example 7-15	Using disable label to stop threads	194
Example 7-16	Using wait fork to wait for child threads	194
Example 7-17	Blocking on an event in Verilog	195
Example 7-18	Output from blocking on an event	196
Example 7-19	Waiting for an event	196

Example 7-20	Output from waiting for an event	196
Example 7-21	Passing an event into a constructor	197
Example 7-22	Waiting for multiple threads with wait fork	198
Example 7-23	Waiting for multiple threads by counting triggers	198
Example 7-24	Waiting for multiple threads using a thread count	199
Example 7-25	Semaphores controlling access to hardware resource	200
Example 7-26	Exchanging objects using a mailbox: the Generator class	203
Example 7-27	Bounded mailbox	204
Example 7-28	Output from bounded mailbox	205
Example 7-29	Producer–consumer without synchronization, part 1	205
Example 7-30	Producer–consumer without synchronization, continued	206
Example 7-31	Producer–consumer without synchronization output	206
Example 7-32	Producer–consumer synchronized with an event	207
Example 7-33	Producer–consumer synchronized with an event, continued	208
Example 7-34	Output from producer–consumer with event	208
Example 7-35	Producer–consumer synchronized with a mailbox	209
Example 7-36	Output from producer–consumer with mailbox	210
Example 7-37	Basic Transactor	211
Example 7-38	Environment class	212
Example 7-39	Basic test program	213
Example 8-1	Base Transaction class	216
Example 8-2	Extended Transaction class	217
Example 8-3	Constructor with argument in an extended class	219
Example 8-4	Driver class	219
Example 8-5	Generator class	220
Example 8-6	Generator class using factory pattern	222
Example 8-7	Environment class	223
Example 8-8	Simple test program using environment defaults	224
Example 8-9	Injecting extended transaction from test	224
Example 8-10	Base and extended class	225
Example 8-11	Copying extended handle to base handle	226
Example 8-12	Copying a base handle to an extended handle	226
Example 8-13	Using \$cast to copy handles	226
Example 8-14	Transaction and BadTr classes	227
Example 8-15	Calling class methods	227
Example 8-16	Building an Ethernet frame with composition	230
Example 8-17	Building an Ethernet frame with inheritance	231
Example 8-18	Building a flat Ethernet frame	232
Example 8-19	Base transaction class with a virtual copy function	233

Example 8-20	Extended transaction class with virtual copy method	234
Example 8-21	Base transaction class with copy_data function	234
Example 8-22	Extended transaction class with copy_data function	235
Example 8-23	Base transaction class with copy_data function	235
Example 8-24	Base callback class	237
Example 8-25	Driver class with callbacks	237
Example 8-26	Test using a callback for error injection	238
Example 8-27	Test using callback for scoreboard	239
Example 9-1	Incomplete D-flip flop model missing a path	244
Example 9-2	Functional coverage of a simple object	249
Example 9-3	Coverage report for a simple object	250
Example 9-4	Coverage report for a simple object, 100% coverage	251
Example 9-5	Functional coverage inside a class	253
Example 9-6	Test using functional coverage callback	254
Example 9-7	Callback for functional coverage	255
Example 9-8	Cover group with a trigger	255
Example 9-9	Module with SystemVerilog Assertion	255
Example 9-10	Triggering a cover group with an SVA	256
Example 9-11	Using auto_bin_max set to 2	257
Example 9-12	Report with auto_bin_max set to 2	258
Example 9-13	Using auto_bin_max for all cover points	258
Example 9-14	Using an expression in a cover point	259
Example 9-15	Defining bins for transaction length	259
Example 9-16	Coverage report for transaction length	260
Example 9-17	Specifying bin names	261
Example 9-18	Report showing bin names	261
Example 9-19	Conditional coverage — disable during reset	262
Example 9-20	Using stop and start functions	262
Example 9-21	Functional coverage for an enumerated type	262
Example 9-22	Report with auto_bin_max set to 2	263
Example 9-23	Specifying transitions for a cover point	263
Example 9-24	Wildcard bins for a cover point	264
Example 9-25	Cover point with ignore_bins	264
Example 9-26	Cover point with auto_bin_max and ignore_bins	264
Example 9-27	Cover point with illegal_bins	265
Example 9-28	Basic cross coverage	266
Example 9-29	Coverage summary report for basic cross coverage	267
Example 9-30	Specifying cross coverage bin names	268
Example 9-31	Cross coverage report with labeled bins	268

Example 9-32	Excluding bins from cross coverage	269
Example 9-33	Specifying cross coverage weight	270
Example 9-34	Cross coverage with bin names	271
Example 9-35	Cross coverage with binsof	271
Example 9-36	Mimicking cross coverage with concatenation	272
Example 9-37	Specifying comments	272
Example 9-38	Specifying per-instance coverage	273
Example 9-39	Report all bins including empty ones	273
Example 9-40	Specifying the coverage goal	274
Example 9-41	Simple parameter	274
Example 9-42	Pass-by-reference	275
Example 9-43	Original class for transaction length	275
Example 9-44	solve...before constraint for transaction length	276
Example 10-1	Interface with clocking block	280
Example 10-2	Testbench using physical interfaces	281
Example 10-3	Testbench using virtual interfaces	282
Example 10-4	Testbench using virtual interfaces	283
Example 10-5	Interface for 8-bit counter	285
Example 10-6	Counter model using X_if interface	285
Example 10-7	Testbench using an array of virtual interfaces	286
Example 10-8	Counter testbench using virtual interfaces	287
Example 10-9	Driver class using virtual interfaces	288
Example 10-10	Testbench using a typedef for virtual interfaces	289
Example 10-11	Driver using a typedef for virtual interfaces	289
Example 10-12	Testbench using an array of virtual interfaces	290
Example 10-13	Testbench passing virtual interfaces with a port	290
Example 10-14	Interface with tasks for parallel protocol	292
Example 10-15	Interface with tasks for serial protocol	293

