

Index

Symbols

! 171
cycle delay 100, 122, 345
\$ 37, 171, 314–315
\$cast 58, 270, 272, 277
\$dist_exponential 187
\$dist_normal 187
\$dist_poisson 187
\$dist_uniform 187
\$error 108
\$exit 95, 101, 118
\$fatal 108, 165–166
\$fclose 64
\$feof 40, 64
\$finish 95, 101, 118, 392
\$fopen 40, 64
\$fscanf 40, 64–65
\$get_coverage 331
\$info 108
\$isunknown 27, 109
\$sprintf 60, 327
\$random 187
\$realtime 76–77
\$root 105, 141–142
\$sformat 60
\$size 29–30, 35, 42
\$strobe 342
\$system 418
\$test\$plusargs 356–357, 418
\$time 27, 76–77
\$timeformat 27, 76–77
\$unit 104–105, 141, 383, 415
\$urandom 187

\$urandom_range 42, 187
\$value\$plusargs 356–357, 418
\$warning 108
%0t 27
%m 327
%t 76
+= 64, 70, 145
+ntb_random_seed 356
:: 136, 238
<< 53
>> 53–55, 156
?: operator 31, 43
[\$] 36
[*] 38
[] 34
^= 68
`define 49

Numerics

2-state types 27, 130, 382, 386, 390, 392
4-state types 27, 130, 390–392

A

abstract class 282–284, 286
Accellera xxx
accumulate operator 64
Active region 93–94, 96, 98, 102
Agent class 158
always block 218
always blocks in programs 101
anonymous enumerated type 55
arguments

- default value 69–70
- sticky 67
- task and function 66
- type 67

array

- assignment 35
- associative 38–40, 42, 47
- compare 31
- constraint 195
- copy 31
- dynamic 34–35, 38, 42, 46–47
- fixed-size 28, 35, 38, 42, 46–47
- handle 150
- linked list 40
- literal 29, 36
- locator methods 42
- methods 41
- multidimensional 28, 30–31
- packed 33–34
- queue 36, 38, 46
- reduction methods 41
- unpacked 29, 34

assertion

- concurrent 108
- coverage 300
- immediate 107–108
- procedural 166

associative array 38–40, 42, 47

at_least option 304, 328

ATM router 109–112, 114–116

atomic stimulus generation 204

auto_bin_max 311–312, 318

auto_bin_max option 327

automatic 68, 74–75, 95, 131–132, 139, 225–226

- in threads 224

aval 390, 392

B

- backdoor load 106
- Backus-Naur Form 205
- base class 263
- BaseTr 282, 292
- begin...end 66, 218
 - optional in tasks and functions 66
- Bergeron, Janick xxxv, 4, 351, 421
- bidirectional constraints 176–177
- bidirectional signal 100

- binsof 322–325
- bit data type 27, 385, 396
- bit streaming
 - see streaming operator 53
- blueprint pattern 265
- BNF 205
- bounded mailbox 245–246
- break 64
- bval 390, 392
- byte data type 27, 384–385, 396

C

- calc_crc method 261–262, 273
- callback
 - coverage 307
 - creation 286
 - inject disturbance 284
 - scoreboard 288
 - usage 287
- cast 52, 58, 61, 65, 270, 277
 - \$cast 53, 272, 280
 - function return value 65
- Cerny, Eduard 421
- chandle data type 385, 387–389, 396
- char 384
- Checker class 158
- class 126, 128
- class constructor 129
- class scope resolution operator 136, 238
- clock generator 102
- clocking block 89–91, 95, 100–101, 114–115
- Cohen, Ben 421
- comment
 - covergroup 327
- comment option 305, 327–328
- compilation unit 104
- composition 259, 274–276, 278
- concatenation
 - string 59
- concurrent assertion 108
- conditional operator 31, 43
- Config class 255
- const type 59, 68, 105, 168
- constrained 8
- constrained-random test 8–9, 161, 187, 295
- constraint

- array 195
- block 165
- dist 169–170, 173, 188, 331
- in extended class 270
- inside 168, 171–174, 176–178, 195
- constraint_mode 182–184, 191
- constructor 129, 131, 263, 282
- containment 144
- context 410–411
- continue 64
- copy
 - deep 153
 - method 153–154, 266–267, 279–280, 282
 - object 151, 279
 - shallow 153
- copy_data method 280–282
- covergroup
 - comment 327
 - generic 325–326
 - option 327
 - sample 303, 307–308
 - trigger 307
- coverpoint 303–304, 306–307, 309–329
- CRC 127
- cross coverage 319–325
- cross module reference 105, 340–341, 346
- cross_num_print_missing option 329
- CRT 161, 187, 295
- Cummings, Cliff 92, 94, 421
- cyclic random 165
- cyclic redundancy check 127

D

- data type
 - bit 27
 - byte 27
 - int 27
 - integer 25
 - logic 26
 - longint 27
 - real 25
 - reg 25–26
 - shortint 27
 - time 25
 - wire 26
- DDR 89

- deallocation 132
- decrement 63
- default statement 89
- default value 69–70
 - 2-state and 4-state 130
- delete method 35, 37, 39, 48
- derived class 263
- Direct Programming Interface 381
- disable 219, 228–232
- disable fork 229–230
- disable label 230–232
- display method 139–140, 261–262, 271
- dist constraint 169–170, 173, 188, 331
- do...while loop 39, 57, 64
- domain 310
- double data rate clock 89
- double data type 384–385
- downcasting 270
- DPI 381
- DPI-C 382, 389
- Driver class 158
- dynamic array 34–35, 38, 42, 46–47
- dynamic threads 223–224

E

- enumerated types 55
- enumerated values 56
- enumeration 55
- Environment class 158
- event 233–237, 253
- event triggered 233–237, 250
- exists 39–40, 48
- expression width 60
- extended class 263
- extern
 - see external routine declaration 139
- external constraint 192–193
- external routine declaration 139–141, 255, 283

F

- file I/O 64
- final block 118
- find_first method 43
- find_index method 43–44
- find_last method 43
- find_last_index method 43
- first 39

first method 39, 47, 57–58
 fixed-size array 28, 35, 38, 42, 46–47
 float data type 384
 for loop 29–30, 39, 57, 63
 force design signals 106
 foreach constraint 195, 200–201, 362
 foreach loop 29–30, 35, 37, 40
 fork...join 218–219
 fork...join_any 218, 221–222
 fork...join_none 218, 220–223, 226–227
 four-state types 27
 see 4-state types 27
 function 65
 arguments 66
 functional coverage 422
 using callbacks 289

G

garbage collection 132
 Generator class 158, 222–223, 236, 238,
 243, 265, 267, 292
 get_coverage 331
 get_inst_coverage 331
 getc method 59–60
 goal option 329

H

handle 128–129
 array 150
 Haque, Faisal 421
 Hardware Description Language xxx
 Hardware Verification Language xxx, 2
 HDL xxx
 histogram 172–173
 hook 266, 285
 HVL xxx–xxx1, 2, 14

I

iff 109, 315
 ignore_bins 317–318, 322–323
 illegal_bins 318
 immediate assertion 107–108
 implicit port connection 103
 import 382
 increment 63
 inheritance 260, 274
 initialization in declaration 74

in-line constraint 192
 inout
 argument type 67
 port type 117
 input
 argument type 67
 port type 117
 insert queue 37
 inside 168, 171
 inside constraint 168, 171–174, 176–178,
 195
 instantiate 129
 int data type 27, 384–385, 396
 integer data type 25, 27, 51
 interface 83, 90, 104, 339
 connecting to port 85
 procedural code 347
 virtual 333–334, 336–338, 341–347
 interprocess communication xxx1, 2, 217,
 232, 253
 intersect 322–323, 325
 io_printf 388, 403
 IPC
 see interprocess communication 217
 iterator argument 43

L

last method 57
 LC3 microcontoller 118
 linked list 40
 local 157, 290–291
 logic data type 26–27, 90, 385, 396
 long long int 404
 longint data type 27, 384–385, 396
 LRM 98
 LRM SystemVerilog 47, 70, 94, 101, 128,
 193, 217, 333, 421

M

macro 48
 macromodule 104
 Magellan 422
 mailbox 240, 246, 251–253
 bounded 245–246
 unbounded 245
 makes Jack a dull boy xxxiii
 malloc 129
 max method 42

method 128, 263
 virtual 272–274, 279–281, 283
 min method 42
 modport 85, 114
 module 127
 Monitor class 158
 multidimensional array 28, 30–31

N

name function 55, 174
 new
 constructor 129–131
 copying objects 151
 new function 130
 new[] operator 35, 131
 next method 39, 47, 57
 nonblocking assignment 84, 88, 92–93, 98, 421
 null 129, 133, 189
 num method 39

O

object 128–129
 copy 151, 279
 deallocation 132
 instantiation 129
 Object-Oriented Programming
 see OOP 125
 Observed region 94
 OOP 125–126
 terminology 128, 263
 OOP analogy
 badge 132
 car 126
 house 128
 open array 400
 Open arrays 400
 OpenVera xxx
 option
 at_least 304, 328
 auto_bin_max 327
 comment 305, 327–328
 cross_num_print_missing 329
 goal 329
 per_instance 327–328, 331
 weight 323–325, 327
 options.auto_bin_max 311–312, 318
 or method 41

output
 argument type 67
 port type 117

P

pack 155–156, 404–405
 package 49, 59, 104, 127
 packed array 33–34
 parameter 49, 59, 104
 parameterized class 290
 parent class 263
 Patt, Yale 422
 per_instance option 327–328, 331
 Perl hash array 39
 physical interfaces 333
 PLI 381, 388
 polymorphism 273–274
 pop_back 37
 port
 connecting to interface 85
 post_randomize 185–186, 203, 205
 post-decrement 63
 post-increment 63
 Postponed 94, 96–97
 pre_randomize 185–186
 pre-decrement 63
 pre-increment 63
 prev method 47, 57
 primitive 104
 PRNG 162, 166, 209–211
 procedural assertion 166
 process 217
 product method 41
 program 74, 95, 104, 127, 227, 257
 program blocks
 single vs. multiple 95
 property 128, 263
 prototype 128, 263, 282
 pseudo-random number generator 162, 166, 209–210
 public 157
 pure
 imported method 407
 virtual method 282–284, 286
 push_back method 43
 push_front 37
 putc method 60

Q

queue 36, 38, 42, 46
 literal 36–37

R

rand 165
 rand_mode 188–189
 randc 165, 174, 195, 201
 randcase 207–208
 random real 167
 random seed 12–13, 162, 166, 209, 211
 random stability 210–212
 randomize function 166–167
 randomize() with 184, 192
 randomize(null) 189
 randsequence 205
 Reactive region 94, 96, 98
 real data type 25, 27, 51, 76, 167, 384–385, 396
 ref 383
 argument type 70
 port type 117
 reference counting 133
 reg data type 25–26
 return 72
 routine arguments 66
 run method 265, 267–268

S

s_vpi_vecval 391
 sample method 303, 307–308
 scenario generation 204
 scoreboard 18, 45–46, 158, 285, 287–289
 using callbacks 285
 semaphore 238–240, 253
 shortint data type 27, 384–385, 396
 shortreal data type 384–385, 396
 signature 274
 sine function 385, 407
 size function 35, 187, 195
 solve...before 180–182, 330–331, 362
 sparse matrix 38
 start method 315
 state variables 189–190
 static
 method 136–138
 storage 74
 variable 135–137

stop method 315
 streaming operator 53–54, 155–156
 string concatenation 59
 string data type 59–60, 384–385, 396
 struct 50–52, 55
 sub class 263
 substr method 60
 sum method 41, 44, 196
 super class 263
 Sutherland, Stuart 351, 422
 SV_PACKED_DATA_NELEMS 390
 SVA 109, 300, 421–422
 svBit 384–386, 388, 393–394
 svBitVecVal 384–386, 390, 392, 401
 svDimension 401
 svdpi.h 384, 386, 388, 401
 svGetArrayElemPtr1 404
 svGetArrayPtr 400–402
 svGetArrElemPtr 402
 svGetArrElemPtr1 402
 svGetArrElemPtr2 402–403
 svGetArrElemPtr3 402
 svGetNameFromScope 416
 svGetScope 416
 svGetScopeFromName 416
 svHigh 401–403
 svIncrement 401
 svLeft 401–402
 svLogic 384, 392
 svLogicVecVal 384, 390–392, 401
 svLow 401–403
 svOpenArrayHandle 400–401, 403–404
 svRight 401–402
 svScope 416
 svSetScope 416
 svSize 401
 svSizeOfArray 401
 synchronous drive 98
 system() 418
 SystemC 381
 SystemVerilog Assertion 109, 300, 309, 421–422

T

task 65
 arguments 66
 template - see parameterized class
 The Matrix 1, 422

this 143–144
 thread 217
 time data type 25, 27, 76
 time literals 76
 timeprecision 75–76
 timescale 75–77, 105–106
 timeunit 75–76
 tolower method 59
 toupper method 59–60
 transactor 158
 transition coverage 317
 triggered 235
 triggered method 233–237, 250
 two-state types
 see 2-state types 27
 typedef 48
 array 50, 73
 class 146
 enum 56
 struct 50–52, 55
 union 51
 virtual interface 345

U

uint user-defined type 49, 199
 unbounded mailbox 245
 union 51
 unique method 43
 unpack 155–156, 404–405
 unpacked array 28–29, 33–34
 unsigned 27, 49, 187, 194, 200

V

van der Schoot, Hans 422
 vc_hdrs.h 384
 Verification Methodology Manual for SystemVerilog
 see VMM 1
 Verilog Programming Interface 391
 Verilog-1995 xxix–xxx, 25, 28, 32, 66–67, 74, 110, 187, 333
 Verilog-2001 xxix–xxx, 28, 32, 59–60, 64, 66, 74, 81, 85, 422
 veriusers.h 388
 virtual
 class - see abstract class
 interface 333–334, 336–338, 341–347

 memory 273
 method 272–274, 279–281, 283
 virtual method 264, 273
 VMM 1, 4–5, 18, 90, 206, 265, 300, 341, 421
 void data type 65, 385, 396
 void function 65, 186
 VPI 381, 391
 vpi_control 392
 vpiFinish 392

W

wait 219, 224, 229, 238, 257
 wait fork 227
 weight option 323–325, 327
 which 232
 wildcard 317, 325
 wire data type 26, 90
 wrap_up method 268

X

XMR 340–341, 346
 xor method 41

