### Index

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>0, 82, =#, 261, ##</td>
</tr>
<tr>
<td>$</td>
<td>assertcontrol, 242, assertfailoff, 251, 284, assertfailon, 251, 284, assertkill, 151, assertnonvacuouson, 251, 284, assertoff, 151, asserton, 151, assertpassoff, 251, 284, assertpasson, 251, 284, assertvacuousoff, 284, assertvacuousoff, 243, changed_gclk, 258, 260, changing_gclk, 258, 260, countones, 150, countones (as Boolean), 151, error, 60, inferred_clock, 271, inferred_disable, 271, info, 60, isunknown, 149, onehot, 147, onehot0, 147, past, 73, past_gclk, 258, 260, rising_gclk, 258, 260, rose, 67, 71, rose — edge detection, 68, rose_gclk, 258, 260, sampled, 257, stable, 72, stable_gclk, 258, 260, stable in procedural block, 73, steady_gclk, 258, 260, warning, 60, .matched, 195, .matched — overlapped operator, 197, .matched with non-overlapping operator, 197, .triggered — end point of a sequence, 188, .triggered (replaced for .ended), 187, .triggered with non-overlapping operator, 190, .triggered with overlapping operator, 189</td>
</tr>
<tr>
<td>[</td>
<td>[*], 101, [*m:n], 82, 98, [*m], 82, 94, 98, [+], 101, [-&gt;m], 82, 114, [-&gt; m:n], 82, 111, [=m], 82, 107,</td>
</tr>
<tr>
<td></td>
<td>Abort properties, 280, ABV adoption in existing design, 64, Accept_on, 280</td>
</tr>
</tbody>
</table>

© Springer International Publishing Switzerland 2016
A.B. Mehta, SystemVerilog Assertions and Functional Coverage, DOI 10.1007/978-3-319-30539-4
Always, 262
And, 124
Antecedent, 37, 39, 40
Application
‘and’ operator, 127
asynchronous FIFO assertions, 207
building a counter using local variables, 229
calling subroutines and local variables, 224
clock delay operator, 84
consecutive repetition range operator, 101
first_match, 138
GoTo repetition—non-consecutive operator, 116
have you transmitted all different lengths of a frame?, 393
if then else, 145
‘intersect’ operator, 132
intersect operator (interesting application), 133
local variables, 179
.matched, 198
‘not’ operator, 141
‘or’ operator, 128
recursive property, 182, 183
repetition non-consecutive operator, 112
seq1 within seq2, 122
sig1 throughout seq1, 118
Applications and important topics, 207
Application assertion control, 152
Application $countones, 150
Application $isunknown, 149
Assert #0, 252
Assert, 35
$Assertcontrol, 285, 288, 289
Assert final, 252
Assertion, 9
advantages, 9
evolution, 7
for specification and review, 26
improve observability, 12
in an emulator, 16
in static formal, 17
major benefits, 15
shorten time to develop, 10
types, 24
Assertion and coverage driven methodology, 364
Assertion based verification (ABV) and functional coverage (FC) based methodology, 362
Assume #0, 254
Assume, 119, 201, 205
Assume final, 254
Asynchronous abort, 280
Asynchronous assertions, 247
Asynchronous FIFO assertions, 207
Asynchronous FIFO testbench and assertions, 210
B
Bind, 61
Binding, 62
Binding properties, 61
Binding properties to design module internal signals, 63
Bins, 372, 382
Blocking action block, 232
Blocking statement, 201
Blocking vs. non-blocking action block, 233
Building a counter, 228
C
Calling subroutines, 222
Case statement, 270
Checkers, 290
nested, 295
illegal conditions, 296
important points, 298
instantiation rules, 301
Chip functionality assertions, 24
Chip interface assertions, 25
Clock delay, 81, 180, 229
Clock delay range, 84
Clock delay range operator, 94
Clock domain crossing (CDC), 155
Clock edge, 44
Clocking basics, 42
Clocking block, 48
Code coverage, 361
Concurrent assertion, 35
Concurrent assertion — with an implication, 240
Concurrent assertion — with ‘cover’, 241
Consecutive repetition operator, 94, 228
Consecutive repetition range, 98
Consecutive repetition range operator, 101
Consequent, 37, 39, 40
Conventions, 28
Cover #0, 254
Cover final, 254
Coverage
follow the bugs, 366
Covergroup, 367
Covergroup/coverpoint example, 371
Covergroup/coverpoint with bins, 374
Covergroup – formal and actual arguments, 375
Covergroup in a 'class', 376
Coverpoint, 367
Cyclic dependency, 236

D
Default clocking block, 48
Deferred 'assume', 254
Deferred 'cover', 254
Deferred immediate assertions, 252
Detect bugs, 15
Detecting and using endpoint of a sequence, 187
Difference between [=m:n] and [->m:n], 115
Disable iff, 58
Disable (property) operator, 58

E
Edge detection, 68, 71
Embedding concurrent assertions in procedural code, 217
Empty sequence, 243
End event, 114
Enough assertions?, 26
Eventually, 264
Expect, 201

F
FIFO assertions, 207
First_match, 137
First_match complex_seq1, 82
Followed by, 261
Formal, 205
Formal arguments, 55
Functional coverage, 5, 11, 16, 366, 385, 401
  bins for transition coverage, 382
  binsof and intersect, 388
  control-oriented, 361
  cross coverage, 378
  data-oriented, 362
  ignore_bins, 386
  illegal_bins, 387
  performance, 1, 53, 69, 70, 86, 371, 391
  performance implication, 392
  PCI cycles transition coverage, 385
  predefined coverage system tasks and functions, 401
  wildcard bins, 386
Functional coverage – language features, 367
Functional coverage options, 395, 397
Functional coverage options for 'covergroup' type - example, 400
Functional coverage options—instance specific —example, 397
Future sampled value functions, 260

G
Gated Clk, 52
Gating expression, 74
Glitch, 33, 54, 253, 260
Global clocking, 260, 287
GoTo repetition, 114

I
IEEE 1800 SystemVerilog, 5
IEEE-1800–2009/2012 features, 251
if (expression) property_expr1 else property_expr2, 82, 143
if then else, 145
Immediate assertions, 31
Implication operator, 37, 40
Intersect, 131
Intersect and and :: What’s the difference?, 137

L
LAB Answers, 343
LAB1
  assertions with/without implication and 'bind', 305
LAB2
  overlap and non-overlap operators, 310
LAB3
  synchronous FIFO assertions, 313
LAB4
  counter, 321
LAB5
  data transfer protocol, 327
LAB6
  PCI read protocol, 336
Let declarations, 273
Let
  in immediate and concurrent assertions, 277
  local scope, 274
  with parameters, 275
Local variable composite sequence with an ‘OR’, 171
Local variables, 167
Local variables – ‘and’ of composite sequences, 173

M
Module interface assertions, 24
Multiple clocks, 155
Multiple implications, 234
Multiple threads, 85
Multiply clocked properties
  ‘and’ operator, 158
  ‘and’ operator between same clocks, 160
  ‘and’ operator between two different clocks, 159