



Questa® Essentials

Student Workbook

© 1991-2009 Mentor Graphics Corporation
All rights reserved.

This document contains information that is proprietary to Mentor Graphics Corporation. The original recipient of this document may duplicate this document in whole or in part for internal business purposes only, provided that this entire notice appears in all copies. In duplicating any part of this document, the recipient agrees to make every reasonable effort to prevent the unauthorized use and distribution of the proprietary information.

This document is for information and instruction purposes. Mentor Graphics reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in written agreements between Mentor Graphics and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Mentor Graphics whatsoever.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

RESTRICTED RIGHTS LEGEND 03/97

U.S. Government Restricted Rights. The SOFTWARE and documentation have been developed entirely at private expense and are commercial computer software provided with restricted rights. Use, duplication or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions set forth in the license agreement provided with the software pursuant to DFARS 227.7202-3(a) or as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, as applicable.

Contractor/manufacturer is:

Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: www.mentor.com
SupportNet: supportnet.mentor.com/

Send Feedback on Documentation: supportnet.mentor.com/user/feedback_form.cfm

TRADEMARKS: The trademarks, logos and service marks ("Marks") used herein are the property of Mentor Graphics Corporation or other third parties. No one is permitted to use these Marks without the prior written consent of Mentor Graphics or the respective third-party owner. The use herein of a third-party Mark is not an attempt to indicate Mentor Graphics as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A current list of Mentor Graphics' trademarks may be viewed at: www.mentor.com/terms_conditions/trademarks.cfm.

End-User License Agreement: You can print a copy of the End-User License Agreement from: www.mentor.com/terms_conditions/enduser.cfm.

Table of Contents

Module 1	
Verification Management: The Verification Plan	11
Objectives	11
True or False?	12
Verification Trends	13
Managing Verification Complexity Through a Process	14
Measurement and Management	15
Verification Disciplines	16
Open Verification Methodology (OVM)	17
OVM Resources	18
Common Verification Flow	19
Ideal Verification Flow	20
Verification Management: The Ideal Process	21
Verification Management: Components of a Good Verification Plan	22
Verification Management: Creating the Verification Plan	23
Planning — Common Obstacles	24
Planning — What Gets Verified?	25
Questa Verification Process	26
Questa Verification Management	27
Questa Verification Management Tasks	28
Verification and Coverage Metrics	29
Unified Coverage Database (UCDB)	30
Questa Verification Management Flow	33
Create the Verification Plan	34
Guidelines for Excel Spreadsheets	35
Recognized Coverage Item Types	36
Linking Verification Plan Sections to the Design	37
Excel Spreadsheet Example	38
Customize the Verification Plan: Adding Columns	39
Customize the Verification Plan: Startstoring	40
Example: FPU Verification Plan in Excel	41
Example 1: FPU Verification Plan: Divide by Zero	42
Example 2: FPU Verification Plan: Operation Delay	44
Opening the Test Browser and Test Tracker	46
Importing the Verification Plan Into the Test Browser	47
Importing a Hierarchical Verification Plan	48
Merging Test Data With the Verification Plan	49
Merging Test Data With the Verification Plan (Command Line)	51
Totals Merge versus Test Associated Merge	52
Viewing Test Data in the Test Tracker	53
Summary	55
Lab 1: The Verification Plan	56

Module 2	
Code Coverage	57
Objectives	57
Purpose of Code Coverage	58
Code Coverage	59
Types of Code Coverage	60
Coverage Modes	63
Statement Execution Coverage	64
Branch or Decision Coverage	65
Expression Coverage	67
Condition Coverage	68
Toggle Coverage	70
Finite State Machine (FSM) Coverage	71
Invoking Code Coverage: Two-Step Flow	72
Invoking Code Coverage: Alternative Two-Step Flows	73
Invoking Code Coverage: Three-Step Flow	74
Union of Coverage Types	76
Enabling Code Coverage in the GUI, 2-Step Flow	77
Enabling Code Coverage in the GUI, 3-Step Flow	78
Code Coverage Windows in Questa	79
Missed Coverage	80
Missed Coverage and Details	82
Missed FSM Coverage	83
Instance Coverage	85
Toggle Coverage in the Objects Pane	86
Toggle Coverage Data	87
FSM Coverage Data in the Objects Window	89
Code Coverage in the Source Window	90
Current Exclusions	93
Excluding Lines and Files From Coverage Statistics	95
Exclusion Filter Files	96
Coverage Text Report	97
Saving and Reloading Coverage Data	98
Managing Coverage Data	99
Managing Coverage Data Off-Line	101
Code Coverage and Optimizations	102
Summary	104
Lab 2: Code Coverage	105
Module 3	
Functional Coverage	107
Objectives	107
What Is Functional Coverage?	108
Part of Total Coverage	109
Functional Coverage, Test Automation, Constraint-Driven Verification	110
SystemVerilog Coverage Constructs	111
Covergroup	112
Cover Directive	113

Table of Contents

Functional Coverage Plan	114
What to Cover?	115
When to Sample?	116
How Much Data to Cover?	117
Cross Coverage to Relate and Filter Data	118
Conditional Sampling	119
Transition Coverage	120
Functional Coverage in the GUI	121
Covergroups Window	122
Cover Directives Window	123
Summary	124
Module 4	
Debugging SystemVerilog Assertions	125
Objectives	125
Assertion Based Verification in Questa	126
Open Verification Library	127
Questa Verification Library	128
SystemVerilog Assertions	130
Assertions Window	131
Assertion Commands	132
Assertion Error Reporting	133
Enabling the Assertion Thread Viewer (ATV)	136
Viewing the Assertion Thread Viewer (ATV)	137
Assertion Thread Viewer (ATV) Window	140
Assertion Thread Viewer (ATV) Graphical Symbols	142
Create Assertion Reports	143
Active Threads	144
SystemVerilog Bind Construct	145
Example: SystemVerilog Bind	146
Example: VHDL Bind	147
Assertions and Performance	148
Summary	149
Lab 3: Debugging SystemVerilog Assertions	150
Module 5	
Constrained Random Verification	151
Objectives	151
Stimulus Generation	152
Random Stimulus	153
Directed versus Random Testing	154
Object Based Randomization	155
Non-Object Based Randomization	156
Constraint Solve Order	157
Randomization in Questa	158
Binary Decision Diagram (BDD)	159
Questa Coding Guidelines	160
What Happens When randomize() Is Called?	162

Failures During Randomize	163
Debugging randomize() Failures	164
Debugging Failures in Constraint Solver	165
Randomize Commands in modelsim.ini (General)	166
Randomize Commands in modelsim.ini (Performance-Related)	167
vsim Randomization Options	170
Summary	171
Lab 4: Debugging a Failure in the Constraint Solver	172
Module 6	
Verification Management: Test Tracking and Debugging	173
Objectives	173
Creating a Test Environment	174
Storing Attributes in the UCDB	176
Obtaining a Coverage Report Summary	180
Saving the UCDB	181
Coverage Modes	182
Coverage View Mode	183
Post-Simulation Analysis	184
Rerunning Tests and Executing Commands	185
Post-Simulation Analysis and Debug: The Wave Window	189
Post-Simulation Analysis and Debug: The Message Viewer	190
Post-Simulation Analysis and Debug: The Source Window	191
Viewing Coverage Statistics in the Browser Window	192
Rank Coverage to Determine Test Contribution	194
Merge Test Plan With Contributing Tests	197
Typical Verification Tracking	199
The Questa Verification Process: Test Tracking	200
Test Tracking Process	201
Link Verification Plan to Coverage Items	203
Questa Verification Analysis	204
Example: FPU Verification Tracking	206
Example: FPU Divide by Zero	207
Finding Unlinked Items in the Test Plan	210
Finding Unlinked Items in Functional Coverage	211
Fixing Unlinked Problems	212
Generating HTML Reports	215
Summary	216
Lab 5: Verification Management: Test Tracking and Debugging	217
Module 7	
Debugging Techniques	219
Objectives	219
“Hardware” versus “Software” Debugging	220
How Do I Debug a SystemVerilog Class-Based Environment?	221
Viewing and Debugging SystemVerilog Class Objects	222
Tracing Function/Method Calls	223
Displaying Class Details	224

Table of Contents

Looking at Class Property Values	225
Questa Debug Windows	226
Processes Window	228
SystemVerilog Classes in the Structural Window	231
Moving From Objects to Source and Wave Windows	232
Analyze Dynamic Data Over Time	233
Wave Window: Expanded Time	234
Debugging Dynamic Objects	236
Watch Window	237
Additional Questa Debug Features	238
Text-Based Dataflow: Tracing Readers	239
Text-Based Dataflow: Tracing Drivers	240
User-Defined Radices	241
Managing User-Defined Radices	242
SystemVerilog Class Debugging	243
Class Tree Window	244
Class Graph Window	248
Parameterized Classes	250
Querying Classes With typespec	252
Example: Querying Classes With typespec	253
Debug Parameterized Classes with TypeSpecInfo	255
TypeSpecInfo Debug Example	256
Summary	257
Lab 6: SystemVerilog Debugging	258

Module 8

Direct Programming Interface (DPI)	259
Objectives	259
Programming Language Interface (PLI)	260
Direct Programming Interface (DPI)	261
Why DPI?	262
DPI versus PLI	263
Performance	265
Terms and Definitions: Import and Export	266
Terms and Definitions: Functions and Tasks	268
Example: Imported Task	269
DPI Use Flow	270
Import Declarations Example: Unix	271
Import Declarations Example: Windows	272
Export Declarations Example: Unix	273
Export Declarations Example: Windows	274
C as Golden Reference Model	275
C as Golden Reference Model – Testbench	277
C as Golden Reference Model – SystemVerilog	278
C as a Testbench	279
Example: C as a Testbench	281
C as a Utility Library	282
Argument and Data Passing in DPI	283

Compiling and Loading DPI in Questa	287
From an OVM Perspective	288
From an OVM Perspective: C/C++ Function	289
Typical Issues/FAQs	290
Summary	292
Lab 7: DPI	293
Module 9	
Power Aware Verification	295
Objectives	295
Low-Power Designs Have Unique Challenges	296
Verification Challenges	297
Power Verification Before Power Aware	298
What Is Power Aware?	299
What Is Power Intent?	300
Power Down and Power Up Behavior	301
Power Aware Models	302
Overlay Power Control on RTL Design Flow	303
What Needs to Be Specified?	305
Low Power Verification Solution	306
UPF Capabilities	307
UPF Commands and Package Functions Supported in Questa	308
Questa Power Aware Verification Flow	309
HDL Compilation	310
Power Annotation	311
Simulation	312
Summary: Questa Power Aware Verification Flow	313
Power Aware Example	314
Power Aware Example – UPF File	315
What Is a Power Domain/Island?	316
Power Aware Example – Simulation Flow	317
Power Aware Example – Simulation Results	318
Interleaver Example	319
Creating the UPF File: Power Domain Specification (Top-Down Approach)	320
Power Domain View	321
Creating the UPF File: Power Domain Specification in Interleaver’s Sub-Blocks	322
Creating the UPF File: Creating a “Switchable” Domain	323
Creating the UPF File: Adding Retention and Isolation	324
Wave Window – Data Corruption	325
Wave Window – Data Retention	326
Wave Window – Data Isolation	327
Power Aware Verification Summary	328
Summary	329
Lab 8: Power Aware	330
Module 10	
Performance	331
Objectives	331

Table of Contents

What Is vopt?	332
What vopt Does	333
Definition: Throughput	334
Break Down: The Optimization Step	335
Incremental Optimization	336
3-Step Method	337
2-Step Method	338
1-Step Method	339
Preserving Visibility	340
Example: Floating Generics	343
Preserving VPI/PLI Visibility	344
Learn Flow Control Files	346
Basic Gate Optimization Flow	347
Sample Cell Report	348
Sample Optimization Log	349
Analyzing Results	350
Other Examples	351
Check Results	352
Useful Gate-Level Switches	353
Compiled SDF	355
Optimize Subsets of the Design	357
Black Box Flow	358
Example: Black Box Flow	359
Black Box – More Detail	360
End Result	362
Performance Profiling	363
Viewing Profiler Results	365
Graphical Views	366
Summary	368
Lab 9: Simulation and Performance	369

Module 11

Transactions	371
Objectives	371
What Is a Transaction	372
Transaction Streams	374
Verilog Transactions in the Wave Window	375
Transaction Recording Flow	376
Recording Transactions in Verilog	377
Verilog Recording Procedure	378
Verilog API Example	384
Verilog Phase versus Parallel Transaction Example	385
Verilog API Prototypes	386
Transaction Life-Cycle	387
Recording Attributes and Relations	388
Viewing Transactions in the Wave Window	389
Viewing Related Transactions	390
Customizing Transaction Appearance	394

Debugging With Tcl	395
Transaction Debugging Commands.....	396
Summary	397
Lab 10: Transactions	398