

The Verilog[®] PLI Handbook

Second Edition

A User's Guide
and
Comprehensive Reference
on the
Verilog Programming Language Interface

Stuart Sutherland
Sutherland HDL, Inc.



KLUWER ACADEMIC PUBLISHERS

Table of Contents

Dedication	v
Table of Contents	vii
About the Author	xv
List of Examples	xvii
Foreword	xxi
Acknowledgments	xxiii
 Introduction	 1
Intended Audience:	1
The IEEE 1364 Verilog PLI standard	2
The history of the Verilog PLI	2
1985: The TF routines	2
1988: The ACC routines	3
1990: The OVI PLI 1.0 standard	3
1993: The OVI PLI 2.0 standard	3
1995: The IEEE 1364-1995 PLI standard and VPI routines	4
2001: The IEEE 1364-2001 PLI standard	4
Ambiguities in the Verilog PLI standard	5
Organization of this book	5
About the PLI examples in this book	6
Other sources of information	7

Part One: The VPI Portion of the Verilog PLI Standard

CHAPTER 1: Creating PLI Applications Using VPI Routines	11
1.1 The capabilities of the Verilog PLI	11
1.2 General steps to create a PLI application	13
1.3 User-defined system tasks and system functions	14
1.4 The \$helloworld PLI application example.....	15
1.5 The \$show_value PLI application example.....	19
CHAPTER 2: Interfacing VPI Applications to Verilog Simulators	27
2.1 General PLI terms as used in this book	27
2.2 System tasks and system functions.....	29
2.3 Instantiated Verilog designs	31
2.4 How PLI applications work.....	34
2.5 calltf routines	36
2.6 compiletf routines	37
2.7 sizetf routines.....	39
2.8 VPI Simulation callback routines	41
2.9 PLI routine inputs and outputs.....	42
2.10 A complete system function example — \$pow	42
2.11 Interfacing PLI applications to Verilog simulators	46
2.12 Using the VPI user_data field.....	51
2.13 Compiling and linking PLI applications.....	53
CHAPTER 3: How to Use the VPI Routines.....	55
3.1 Specification of \$show_all_nets and \$show_all_signals.....	55
3.2 The VPI routine library.....	56
3.3 Advantages of the VPI library	57
3.4 Verilog HDL objects	59
3.5 Obtaining object handles	62
3.6 Accessing the arguments of a system task/function	65
3.7 Printing messages from VPI applications.....	66
3.8 Accessing object properties	67
3.9 Reading the logic values of Verilog objects.....	69
3.10 Reading the current simulation time.....	71
3.11 Controlling simulation from PLI applications.....	73
3.12 A complete PLI application using VPI routines	74
3.13 Obtaining handles for reg and variable data types	78
3.14 Obtaining a handle to the current hierarchy scope	85
3.15 Obtaining handles to multiple task/function arguments.....	91

CHAPTER 4: Details about the VPI Routine Library.....	97
4.1 PLI application performance considerations	98
4.2 The VPI string buffer.....	99
4.3 VPI error handling	100
4.4 VPI object diagrams	102
4.5 Obtaining handles for objects	106
4.6 System task and system function objects	116
4.7 Storing data for each instance of a system task/function.....	122
4.8 Traversing Verilog hierarchy using object relationships.....	128
4.9 Writing messages to files.....	137
4.10 Reading and using simulation times	140
4.11 User-defined invocation options.....	144
4.12 Controlling Simulations.....	148
CHAPTER 5: Reading and Modifying Values Using VPI Routines.....	151
5.1 Accessing objects which have logic values.....	151
5.2 Reading object logic values.....	152
5.3 Writing values to Verilog objects.....	173
5.4 Returning logic values of system functions.....	177
5.5 Reading and writing automatic variables	180
5.6 Reading IEEE 1364-2001 Verilog attributes.....	182
5.7 Reading IEEE 1364-1995 specparam constant attributes.....	182
5.8 Accessing Verilog net, reg, memory and arrays.....	185
5.9 Reading and modifying delay values.....	186
CHAPTER 6: Synchronizing to Simulations Using VPI Callbacks.....	197
6.1 PLI application callbacks.....	197
6.2 Sharing information between callback routines	198
6.3 Registering simulation callback routines.....	201
6.4 Removing scheduled callbacks.....	204
6.5 Avoiding memory leaks with simulation callbacks.....	205
6.6 Simulation action-related callbacks.....	206
6.7 Simulation time-related callbacks	216
6.8 Simulation event-related callbacks.....	235
CHAPTER 7: Interfacing to C Models Using VPI Routines.....	241
7.1 How to interface C models with Verilog simulations	242
7.2 Creating the C language model	244
7.3 A C model example	245
7.4 Creating a Verilog shell module.....	248
7.5 Creating a combinational logic interface to a C model	249
7.6 Creating a sequential logic interface to a C model	256

7.7	Synchronizing with the end of a simulation time step.....	259
7.8	Synchronizing with a future simulation time step.....	263
7.9	Allocating storage within a C model.....	263
7.10	Representing propagation delays in a C model.....	267

Part Two: The TF/ACC Portion of the Verilog PLI Standard

CHAPTER 8: Creating PLI Applications Using TF and ACC Routines.....273

8.1	The capabilities of the Verilog PLI.....	273
8.2	General steps to create a PLI application.....	275
8.3	User-defined system tasks and system functions.....	276
8.4	The \$hello PLI application example.....	277
8.5	The \$show_value PLI application example.....	280

CHAPTER 9: Interfacing TF/ACC Applications to Verilog Simulators.....287

9.1	General PLI terms as used in this book.....	287
9.2	System tasks and system functions.....	289
9.3	Instantiated Verilog designs.....	291
9.4	How PLI applications work.....	294
9.5	calltf routines.....	296
9.6	checktf routines.....	297
9.7	size tf routines.....	299
9.8	misctf routines.....	300
9.9	PLI routine inputs and outputs.....	302
9.10	A complete system function example — \$pow.....	303
9.11	Interfacing PLI applications to Verilog simulators.....	305
9.12	Using the user_data field.....	310
9.13	Compiling and linking PLI applications.....	311

CHAPTER 10: How to Use the TF Routines.....313

10.1	The TF Library.....	313
10.2	System tasks and system functions.....	315
10.3	The PLI string buffer.....	318
10.4	Controlling simulation.....	319
10.5	Printing messages.....	319
10.6	Checking system task/function arguments.....	323
10.7	The TF work area.....	325
10.8	Reading and using simulation times.....	329
10.9	Reading simulation invocation options.....	334
10.10	Utility TF routines.....	335
10.11	A complete PLI application using TF Routines.....	337

CHAPTER 11: Reading and Writing Values Using TF Routines.....	341
11.1 Working with a 4-logic value, multiple strength level system	341
11.2 How the PLI writes values into Verilog	342
11.3 Reading and writing 2-state values.....	344
11.4 Reading and writing 4-state logic values using C strings.....	349
11.5 Reading Verilog strings	357
11.6 Reading and writing 4-state values using aval/bval encoding.....	358
11.7 Reading 4-state logic values with strengths	371
11.8 Reading from and writing into Verilog memory arrays	378
CHAPTER 12: Synchronizing to Simulations Using Miscf Routines	401
12.1 The purpose of the miscf routine.....	401
12.2 Automatic callbacks for simulation events.....	405
12.3 Application scheduled callbacks at the end of a time step	408
12.4 Application-scheduled callbacks at a future simulation time	414
12.5 System task/function argument value change callbacks.....	418
12.6 Simulation save and restart checkpoint files	422
12.7 A complete example of using miscf routine callbacks	424
CHAPTER 13: Interfacing to C Models Using TF Routines.....	431
13.1 How to interface C models with Verilog simulations	432
13.2 Creating the C language model	434
13.3 A C model example	435
13.4 Creating a Verilog shell module.....	437
13.5 Creating a combinational logic interface to a C model	439
13.6 Creating a sequential logic interface to a C model.....	445
13.7 Synchronizing with the end of a simulation time step.....	448
13.8 Synchronizing with a future simulation time step.....	451
13.9 Allocating storage within a C model	451
13.10 Representing propagation delays in a C model	455
CHAPTER 14: How to Use the ACC Routines.....	459
14.1 Specification of \$show_all_nets and \$show_all_signals.....	459
14.2 The ACC routine library	460
14.3 Advantages of the ACC library	462
14.4 Verilog HDL objects	463
14.5 ACC handle routines	465
14.6 ACC next routines	466
14.7 Accessing object types and fulltypes.....	468
14.8 Accessing the names of objects	470
14.9 The ACC string buffer	471
14.10 Reading the logic values of Verilog objects	472

14.11	A complete PLI application using ACC routines	473
14.12	Accessing handles for reg and variable data types	475
14.13	Obtaining handles to the current hierarchy scope	479
14.14	Obtaining handles to multiple task/function arguments	483
CHAPTER 15: Details on the ACC Routine Library		487
15.1	PLI application performance considerations	487
15.2	Initializing and configuring ACC routines	489
15.3	ACC routine error handling	492
15.4	Using ACC object diagrams	494
15.5	Using ACC handle routines	498
15.6	Using ACC next routines	499
15.7	Traversing Verilog hierarchy using object relationships	501
15.8	Traversing hierarchy across module ports	506
15.9	Identifying modules and library cells	512
15.10	Accessing loads and drivers	514
15.11	Accessing model timing	516
15.12	Counting the number of objects	523
15.13	Collecting and maintaining lists of object handles	525
15.14	Obtaining object handles using an object's name	528
15.15	Comparing ACC handles	532
CHAPTER 16: Reading and Modifying Values Using ACC Routines		535
16.1	Using ACC fetch routines	536
16.2	Reading object type properties	538
16.3	Accessing an object's source code location	542
16.4	Reading the simulation invocation commands	544
16.5	Accessing objects in simulation which have logic values	546
16.6	Reading the values of system task/function arguments	547
16.7	Reading object logic values	552
16.8	Writing values into Verilog objects	564
16.9	Returning logic values of system functions	570
16.10	Reading module time scale information	572
16.11	Reading delay values	575
16.12	Writing delay values into an object	584
16.13	Reading parameter constant values	590
16.14	Using constants as model attributes	592
16.15	Reading and modifying path pulse controls	594
CHAPTER 17: Using the Value Change Link (VCL)		601
17.1	An overview of the VCL routines	601
17.2	Adding and removing VCL flags on Verilog objects	602

17.3	Using the VCL consumer routine	605
17.4	An example of using Value Change Link routines.....	608
17.5	Obtaining object handles from the consumer routine.....	609
CHAPTER 18: Interfacing to C Models Using ACC Routines		611
18.1	How to interface C models with Verilog simulations	612
18.2	Creating the C language model	614
18.3	A C model example	615
18.4	Creating a Verilog shell module	618
18.5	Creating a combinational logic interface to a C model	619
18.6	Creating a sequential logic interface to a C model.....	624
18.7	Synchronizing with the end of a simulation time step.....	627
18.8	Synchronizing with a future simulation time step	631
18.9	Allocating storage within a C model	631
18.10	Representing propagation delays in a C model	637

Appendices

APPENDIX A: Linking PLI Applications to Verilog Simulators		641
A.1	The PLI interface mechanism	642
A.2	Linking to the Cadence Verilog-XL and NC-Verilog simulators.....	649
A.3	Linking to the Synopsys VCS simulator.....	657
A.4	Linking to the Model Technology ModelSim simulator	664
APPENDIX B: The IEEE 1364-2001 TF Routine Library		669
B.1	TF routine definitions.....	670
APPENDIX C: The IEEE 1364-2001 ACC Routine Library		685
C.1	ACC object relationships	686
C.2	ACC routine definitions	699
APPENDIX D: The IEEE 1364-2001 VPI Routine Library		719
D.1	VPI object relationships	720
D.2	VPI routine definitions.....	754
Index		767
About the CD.....		783