

# Contents

<b>1</b>	<b>Why SYSTEMC: ESL and TLM.....</b>	<b>1</b>
1.1	Introduction.....	1
1.2	ESL Overview.....	2
1.2.1	Design Complexity .....	2
1.2.2	Shortened Design Cycle = Need For Concurrent Design .....	3
1.3	Transaction-Level Modeling.....	7
1.3.1	Abstraction Models.....	7
1.3.2	An Informal Look at TLM.....	8
1.3.3	TLM Methodology.....	10
1.4	A Language for ESL and TLM: SystemC .....	14
1.4.1	Language Comparisons and Levels of Abstraction .....	15
1.4.2	SystemC: IEEE 1666 .....	16
1.4.3	Common Skill Set.....	16
1.4.4	Proper Simulation Performance and Features.....	16
1.4.5	Productivity Tool Support.....	17
1.4.6	TLM Concept Support .....	17
1.5	Conclusion .....	18
<b>2</b>	<b>Overview of SystemC.....</b>	<b>19</b>
2.1	C++ Mechanics for SystemC.....	20
2.2	SystemC Class Concepts for Hardware .....	22
2.2.1	Time Model.....	22
2.2.2	Hardware Data Types.....	23
2.2.3	Hierarchy and Structure .....	23
2.2.4	Communications Management .....	23
2.2.5	Concurrency .....	24
2.2.6	Summary of SystemC Features for Hardware Modeling .....	24
2.3	Overview of SystemC Components .....	25
2.3.1	Modules and Hierarchy.....	25
2.3.2	SystemC Threads and Methods .....	25
2.3.3	Events, Sensitivity, and Notification.....	26

2.3.4	SystemC Data Types .....	27
2.3.5	Ports, Interfaces, and Channels .....	27
2.3.6	Summary of SystemC Components .....	28
2.4	SystemC Simulation Kernel .....	29
<b>3</b>	<b>Data Types .....</b>	<b>31</b>
3.1	Native C++ Data Types .....	31
3.2	SystemC Data Types Overview .....	32
3.3	SystemC Logic Vector Data Types .....	33
3.3.1	sc_bv<W> .....	33
3.3.2	sc_logic and sc_lv<W> .....	34
3.4	SystemC Integer Types .....	35
3.4.1	sc_int<W> and sc_uint<W> .....	35
3.4.2	sc_bigint<W> and sc_bignint<W> .....	35
3.5	SystemC Fixed-Point Types .....	36
3.6	SystemC Literal and String .....	39
3.6.1	SystemC String Literals Representations .....	39
3.6.2	String Input and Output .....	40
3.7	Operators for SystemC Data Types .....	41
3.8	Higher Levels of Abstraction and the STL .....	43
3.9	Choosing the Right Data Type .....	44
3.10	Exercises .....	44
<b>4</b>	<b>Modules .....</b>	<b>47</b>
4.1	A Starting Point: sc_main .....	47
4.2	The Basic Unit of Design: SC_MODULE .....	49
4.3	The SC_MODULE Class Constructor: SC_CTOR .....	50
4.4	The Basic Unit of Execution: Simulation Process .....	51
4.5	Registering the Basic Process: SC_THREAD .....	52
4.6	Completing the Simple Design: main.cpp .....	53
4.7	Alternative Constructors: SC_HAS_PROCESS .....	53
4.8	Two Styles Using SystemC Macros .....	55
4.8.1	The Traditional Coding Style .....	55
4.8.2	Recommended Alternate Style .....	56
4.9	Exercises .....	57
<b>5</b>	<b>A Notion of Time .....</b>	<b>59</b>
5.1	sc_time .....	59
5.1.1	SystemC Time Resolution .....	60
5.1.2	Working with sc_time .....	61
5.2	sc_time_stamp() .....	61
5.3	sc_start() .....	62
5.4	wait(sc_time) .....	63
5.5	Exercises .....	64

<b>6</b>	<b>Concurrency</b>	65
6.1	Understanding Concurrency	65
6.2	Simplified Simulation Engine	68
6.3	Another Look at Concurrency and Time	70
6.4	The SystemC Thread Process	71
6.5	SystemC Events	72
6.5.1	Causing Events	73
6.6	Catching Events for Thread Processes	74
6.7	Zero-Time and Immediate Notifications	75
6.8	Understanding Events by Way of Example	78
6.9	The SystemC Method Process	81
6.10	Catching Events for Method Processes	83
6.11	Static Sensitivity for Processes	83
6.12	Altering Initialization	86
6.13	The SystemC Event Queue	87
6.14	Exercises	88
<b>7</b>	<b>Dynamic Processes</b>	89
7.1	Introduction	89
7.2	sc_spawn	89
7.3	Spawn Options	91
7.4	A Spawned Process Example	92
7.5	SC_FORK/SC_JOIN	93
7.6	Process Control Methods	96
7.7	Exercises	97
<b>8</b>	<b>Basic Channels</b>	99
8.1	Primitive Channels	100
8.2	sc_mutex	100
8.3	sc_semaphore	102
8.4	sc_fifo	104
8.5	Exercises	106
<b>9</b>	<b>Evaluate-Update Channels</b>	107
9.1	Completed Simulation Engine	108
9.2	SystemC Signal Channels	110
9.3	Resolved Signal Channels	113
9.4	Template Specializations of sc_signal Channels	115
9.5	Exercises	116
<b>10</b>	<b>Structure</b>	117
10.1	Module Hierarchy	117
10.2	Direct Top-Level Implementation	119

10.3	Indirect Top-Level Implementation.....	119
10.4	Direct Submodule Header-Only Implementation .....	120
10.5	Direct Submodule Implementation .....	120
10.6	Indirect Submodule Header-Only Implementation.....	121
10.7	Indirect Submodule Implementation.....	122
10.8	Contrasting Implementation Approaches.....	123
10.9	Exercises .....	123
<b>11</b>	<b>Communication .....</b>	<b>125</b>
11.1	Communication: The Need for Ports .....	125
11.2	Interfaces: C++ and SystemC .....	126
11.3	Simple SystemC Port Declarations .....	129
11.4	Many Ways to Connect .....	130
11.5	Port Connection Mechanics .....	132
11.6	Accessing Ports From Within a Process .....	134
11.7	Exercises .....	135
<b>12</b>	<b>More on Ports &amp; Interfaces.....</b>	<b>137</b>
12.1	Standard Interfaces.....	137
12.1.1	SystemC FIFO Interfaces.....	137
12.1.2	SystemC Signal Interfaces .....	139
12.1.3	sc_mutex and sc_semaphore Interfaces .....	140
12.2	Sensitivity Revisited: Event Finders and Default Events.....	140
12.3	Specialized Ports .....	142
12.4	The SystemC Port Array and Port Policy .....	145
12.5	SystemC Exports.....	148
12.6	Connectivity Revisited.....	153
12.7	Exercises .....	155
<b>13</b>	<b>Custom Channels and Data.....</b>	<b>157</b>
13.1	A Review of SystemC Channels and Interfaces.....	157
13.2	The Interrupt, a Custom Primitive Channel .....	158
13.3	The Packet, a Custom Data Type for SystemC .....	159
13.4	The Heartbeat, a Custom Hierarchical Channel.....	162
13.5	The Adaptor, a Custom Primitive Channel .....	164
13.6	The Transactor, a Custom Hierarchical Channel .....	166
13.7	Exercises .....	170
<b>14</b>	<b>Additional Topics .....</b>	<b>171</b>
14.1	Error and Message Reporting .....	171
14.2	Elaboration and Simulation Callbacks.....	174
14.3	Configuration .....	175
14.4	Programmable Structure .....	177
14.5	sc_clock, Predefined Processes .....	181

14.6	Clocked Threads, the SC_CTHREAD.....	182
14.7	Debugging and Signal Tracing.....	185
14.8	Other Libraries: SCV, ArchC, and Boost.....	187
14.9	Exercises .....	187
<b>15</b>	<b>SCV.....</b>	<b>189</b>
15.1	Introduction.....	189
15.2	Data Introspection.....	189
15.2.1	Components for scv_extension Interface.....	190
15.2.2	Built-In scv_extensions.....	192
15.2.3	User-Defined Extensions .....	193
15.3	scv_smart_ptr Template .....	193
15.4	Randomization .....	194
15.4.1	Global Configuration .....	194
15.4.2	Basic Randomization .....	196
15.4.3	Constrained Randomization.....	197
15.4.4	Weighted Randomization.....	198
15.5	Callbacks.....	200
15.6	Sparse Arrays.....	201
15.7	Transaction Sequences.....	202
15.8	Transaction Recording .....	203
15.9	SCV Tips.....	204
15.10	Exercises .....	204
<b>16</b>	<b>OSCI TLM.....</b>	<b>207</b>
16.1	Introduction.....	207
16.2	Architecture.....	208
16.3	TLM Interfaces .....	210
16.3.1	Unidirectional Blocking Interfaces .....	211
16.3.2	Unidirectional Non-Blocking Interfaces.....	211
16.3.3	Bidirectional Blocking Interface.....	213
16.4	TLM Channels .....	213
16.5	Auxiliary Components.....	214
16.5.1	TLM Master.....	215
16.5.2	TLM Slave .....	215
16.5.3	Router and Arbiter .....	216
16.6	A TLM Example.....	217
16.7	Summary.....	220
16.8	Exercises .....	220
<b>17</b>	<b>Odds &amp; Ends .....</b>	<b>223</b>
17.1	Determinants in Simulation Performance.....	223
17.1.1	Saving Time and Clocks .....	224
17.1.2	Moving Large Amounts of Data .....	225

17.1.3	Too Many Channels .....	226
17.1.4	Effects of Over Specification .....	227
17.1.5	Keep it Native .....	227
17.1.6	C++ Compiler Optimizations.....	227
17.1.7	C++ Compilers.....	227
17.1.8	Better Libraries .....	227
17.1.9	Better and More Simulation Computers .....	228
17.2	Features of the SystemC Landscape .....	228
17.2.1	Things You Wish Would Just Go Away .....	228
17.2.2	Development Environment .....	230
17.2.3	Conventions and Coding Style.....	230
17.3	Next Steps .....	231
17.3.1	Guidelines for Adopting SystemC .....	231
17.3.2	Resources for Learning More .....	231
<b>Appendix A</b>	.....	<b>235</b>
A.1	Background of C++ .....	236
A.2	Structure of a C Program .....	236
A.3	Comments .....	237
A.4	Streams (I/O).....	237
A.4.1	Streaming vs. printf.....	238
A.5	Basic C Statements .....	238
A.5.1	Expressions and Operators.....	238
A.5.2	Conditional.....	240
A.5.3	Looping.....	241
A.5.4	Altering Flow .....	242
A.6	Data Types.....	242
A.6.1	Built-In Data Types .....	243
A.6.2	User-Defined Data Types .....	243
A.6.3	Constants.....	246
A.6.4	Declaration vs. Definition .....	246
A.7	Functions.....	247
A.7.1	Pass By Value and Return .....	248
A.7.2	Pass by Reference .....	248
A.7.3	Overloading.....	249
A.7.4	Constant Arguments.....	249
A.7.5	Defaults for Arguments.....	250
A.7.6	Operators as Functions.....	250
A.8	Classes.....	251
A.8.1	Member Data and Member Functions .....	251
A.8.2	Constructors and Destructors.....	252
A.8.3	Destructors .....	255
A.8.4	Inheritance.....	256
A.8.5	Public, Private and Protected Access.....	258
A.8.6	Polymorphism .....	258

A.8.7	Constant Members .....	260
A.8.8	Static Members .....	260
A.9	Templates .....	261
A.9.1	Defining Template Functions.....	261
A.9.2	Using Template Functions .....	261
A.9.3	Defining Template Classes.....	262
A.9.4	Using Template Classes .....	262
A.9.5	Template Considerations.....	262
A.10	Names and Namespaces.....	263
A.10.1	Meaningful Names.....	263
A.10.2	Ordinary Scope .....	263
A.10.3	Defining Namespaces .....	264
A.10.4	Using Names and Namespaces .....	264
A.10.5	Anonymous Namespaces .....	264
A.11	Exceptions.....	265
A.11.1	Watching for and Catching Exceptions.....	265
A.11.2	Throwing Exceptions .....	266
A.11.3	Functions that Throw .....	267
A.12	Standard Library Tidbits .....	268
A.12.1	Strings .....	268
A.12.2	File I/O .....	268
A.12.3	Standard Template Library .....	270
A.13	Closing Thoughts .....	270
A.14	References.....	271
<b>Index</b>	.....	<b>273</b>

SystemC: From the Ground Up, Second Edition

Black, D.C.; Donovan, J.; Bunton, B.; Keist, A.

2010, XXIII, 281 p., Hardcover

ISBN: 978-0-387-69957-8