

Contents

1	The Third Revolution	1
	The Problem.....	1
	Divide and Conquer	3
	The General Model	4
	Rule of Seven	6
	Tightly Coupled vs. Loosely Coupled Systems	7
	The Challenge of Verification	10
	The Pursuit of Simplicity	11
	The Changing Landscape of Design	12
	Structure of This Book	12
2	Simplifying RTL Design	15
	Challenges.....	15
	Syntactic Fluff.....	16
	Concurrency and State Space	17
	Techniques	19
	Encapsulating Combinational Code.....	20
	Structuring Sequential Code	21
	Using High Level Data Types	23
	Thinking High-level	25
3	Reducing Complexity in Control-Dominated Designs.....	27
	Original Code.....	28
	State Space in the Original Design	29
	Partitioning	30
	Comments	30
	Syntactic fluff	32
	Preprocessing Sequential Code.....	32
	Preprocessing Combinational Code	34
	Refactoring Sequential Code	35
	Recoding the State Machine	35
	Relocating Other Sequential Code	37

Rewriting Combinational Code	38
Analyzing the New Code	40
Restructuring for a Hierarchical State Machine	41
System Verilog	41
Simplified Block Diagram	43
Summary	44
4 Hierarchical State Machines	47
Restructuring for a Hierarchical State Machine	47
General Model for HFSMs	47
Converting the BCU to a HFSM	51
5 Measuring and Minimizing State Space	55
Input, Output, and Internal State Space	55
Preliminary Calculations of State Space	56
Shallow vs. Deep State Space	58
The Cross Product of State Spaces	60
Sequential Processes and Internal State	63
Encapsulating Sequential Code	63
State Machines as Sequential Processes	64
Hierarchical State Machines	64
Examples	65
Summary- Counting State	68
Input State Space	68
Output State Space	68
Internal State Space	68
State Space for Hierarchical State Machines	69
6 Verification	71
Some Simple Examples of Verifiable Designs	72
Verification overview	73
Goals of Complete Verification	74
Verifying State Machines	75
Example: The BCU	77
A Canonical Design	80
Structure of the Canonical Design	80
Separating Data from Control	81
Verifying the Control Path: The State Machine	82
Line Coverage	83
Condition Coverage	83
Conditional Range Coverage	83
Cycle Coverage	84
Input State Coverage	84

Verifying the Data Path.....	85
Data Path Uniqueness	85
Data Range Coverage.....	85
Verifying the Data Path Algorithm	86
Summary.....	86
7 Reducing Complexity in Data Path Dominated Designs	89
Problems and Limitations in the Original Code	91
Minimizing Lines of Code	92
Other Versions of the Code	96
Task Version	97
More Code Size Reduction	98
Untimed Version.....	100
Experimental Versions	102
Simulation Results for the Different Versions of the DCT	103
Reference Versions.....	103
Synthesis Results	105
Gates per Line of Code	105
Formal Verification	106
Canonical Design.....	106
Summary	107
8 Simplifying Interfaces.....	109
Command-based Interface	109
Example: CPU Pipeline	111
Example: BCU	113
Example: USB	115
Impact on Verification.....	117
Separating Data and Control	117
General Connectivity	118
Concurrency and Analysis	120
Total State Space.....	120
Summary	120
9 Complexity at the Chip Level.....	123
From Command to Transaction Interfaces.....	126
JPEG Example	127
USB Example.....	128
Software Driver.....	128
Virtual Platforms and Software Development.....	129
Connectivity and Complexity at the SoC Level.....	129
Function and Structure	130
Connectivity and Bandwidth.....	130

Transactions and Complexity	131
Limits to Chip Level Verification	133
Sub-systems and SoC Design	134
Summary	137
10 Raising Abstraction Above RTL.....	139
The Challenge	139
Current High Level Synthesis Tools	140
Closing the Abstraction Gap	145
SystemC	146
The Right Usage Model for High Level Design	146
SystemVerilog as a High Level Design Language	147
Raising the Level of Abstraction of RTL	149
Domain Specific Languages	150
SystemVerilog as a Domain Specific Language	150
SystemVerilog Primitives	151
Proposal	152
11 SystemVerilog Extensions	155
Overview	155
Basic Extensions	155
smodule	158
bit_ff	159
bit_comb	159
\$clock	159
\$reset	159
state_machine	160
state_var	160
done (to signal the end of a sub-state machine activity)	160
combinational and sequential assignments in the same case statement	160
Other Capabilities	161
First	161
Functions and Tasks	161
Assignments Outside State Machines	161
Iterative State Loops	162
Fork and Join	168
Summary	168
There is No Substitute for Good Code	169

12 The Future of Design	171
Design	172
Function Does Not Scale	172
Small is Beautiful – and Tractable	175
How Does IP Help?.....	175
Automation and Scaling.....	176
The Future of Design	177
Verification.....	177
Visualization	178
Drivers of the Solution.....	179
Summary	180
Appendix A	181
Appendix B	191
Appendix C	207
Appendix D	223
References.....	231
Index.....	233



<http://www.springer.com/978-1-4419-8585-9>

The Simple Art of SoC Design
Closing the Gap between RTL and ESL
Keating, Synopsys Fellow, M.
2011, XVII, 234 p., Hardcover
ISBN: 978-1-4419-8585-9