

# *Co-Verification of Hardware and Software for ARM SoC Design*

*by Jason R. Andrews*



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON  
NEW YORK • OXFORD • PARIS • SAN DIEGO  
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Newnes is an imprint of Elsevier



Newnes

# Contents

|   |              |
|---|--------------|
| <b>Foreword</b> .....   | <b>xiii</b>  |
| <b>Preface</b> .....  | <b>xv</b>    |
| Why Is This Book Important?.....                                      | xv           |
| Audience .....  | xvi          |
| Prerequisite Knowledge .....  | xvi          |
| About Hardware/Software Co-Verification .....                         | xvi          |
| <b>Acknowledgments</b> .....  | <b>xvii</b>  |
| <b>About the Author</b> .....   | <b>xix</b>   |
| <b>About Verisity</b> .....   | <b>xxi</b>   |
| <b>What's on the CD-ROM?</b> .....                                    | <b>xxiii</b> |
| <b>Chapter 1: Embedded System Verification: An Introduction</b> ..... | <b>1</b>     |
| What's an Embedded System?.....                                       | 3            |
| Embedded Systems Are Everywhere .....                                 | 5            |
| Consumer Electronics .....  | 5            |
| Wireless.....   | 5            |
| Medical .....   | 5            |
| Networking.....   | 5            |
| Security .....  | 5            |
| Imaging.....  | 5            |
| Storage.....  | 5            |
| Automotive .....  | 5            |
| Design Constraints .....  | 6            |
| Cost.....   | 6            |
| Memory.....   | 6            |
| Power .....   | 7            |
| Real-Time Response .....  | 7            |

# Contents

---

|  |           |
|--|-----------|
| Performance.....   | 7         |
| System Size .....  | 8         |
| Reliability .....  | 8         |
| Time-to-Market .....   | 8         |
| Embedded Systems Decomposition.....                          | 9         |
| Microprocessors, Chips and Boards.....                       | 9         |
| Embedded System Classifications.....                         | 11        |
| Little or No Custom Hardware Design .....                    | 12        |
| A Lot of Custom Hardware – SoB Design .....                  | 12        |
| A Lot of Custom Hardware – SoC Design.....                   | 13        |
| Embedded System Design Process.....                          | 14        |
| Requirements.....  | 15        |
| System Architecture.....                                     | 15        |
| Microprocessor Selection .....                               | 15        |
| Hardware Design .....  | 16        |
| Software Design.....   | 16        |
| Hardware and Software Integration.....                       | 16        |
| Verification and Validation .....                            | 16        |
| Verification: Does it Work?.....                             | 17        |
| Validation: Did We Build the Right Thing?.....               | 17        |
| Human Interaction.....                                       | 18        |
| What is this Book About?.....                                | 20        |
| Scope and Outline.....                                       | 22        |
| <b>Chapter 2: Hardware and Software Design Process .....</b> | <b>25</b> |
| Three Components of SoC Verification .....                   | 25        |
| Verification Platform .....                                  | 26        |
| Software Engineer’s View of the World.....                   | 35        |
| Hardware Engineer’s View of the World.....                   | 37        |
| Example .....  | 37        |
| Software Development Tools .....                             | 39        |
| Editor .....   | 39        |
| Source Code Revision Control .....                           | 39        |
| Compiler.....  | 40        |
| Debugger.....  | 41        |
| Simulator .....  | 41        |
| Development Board.....                                       | 42        |
| Integrated Development Environment (IDE) .....               | 42        |

---

|  |           |
|--|-----------|
| Software Debugging Connections.....                                      | 42        |
| JTAG .....   | 43        |
| Stub.....  | 43        |
| Direct Connection .....  | 44        |
| Types of Software.....   | 44        |
| System Initialization and HAL .....                                      | 44        |
| Diagnostic Suite .....   | 45        |
| Real-Time Operating System (RTOS).....                                   | 45        |
| Device Drivers and Application Software .....                            | 45        |
| Software Development Process .....                                       | 46        |
| Hardware Development Tools.....  | 52        |
| Editor .....   | 52        |
| Source Code Revision Control .....                                       | 53        |
| Lint Tools.....  | 54        |
| Code Coverage .....  | 54        |
| Debugging Tools .....  | 55        |
| Verification Languages .....   | 55        |
| Assertions .....   | 56        |
| Debugging Defined .....  | 58        |
| Memory Models.....   | 59        |
| Microprocessor Models.....   | 61        |
| Hardware Design Process.....   | 62        |
| Microprocessor Review.....   | 63        |
| Hardware and Software Interaction .....                                  | 64        |
| Software Debugging Characteristics .....                                 | 64        |
| Hardware Debugging Characteristics.....                                  | 64        |
| <b>Chapter 3: SoC Verification Topics for the ARM Architecture .....</b> | <b>69</b> |
| ARM Background.....  | 69        |
| ARM Architecture .....   | 70        |
| ARM Architectures, Families, and CPU Cores.....                          | 71        |
| Thumb Instruction Set.....   | 75        |
| Programming Model.....   | 76        |
| Instruction Set.....   | 78        |
| Data Transfer Instructions .....   | 78        |
| Coprocessor Instructions .....   | 79        |
| Exceptions and Interrupts .....  | 80        |
| Memory Layout and Byte Order.....  | 83        |

|   |            |
|---|------------|
| ARM Bus Interface Protocols.....                              | 84         |
| ARM7TDMI Bus Protocol.....                                    | 85         |
| AMBA Specification.....                                       | 89         |
| Introduction to AMBA Protocols.....                           | 91         |
| AMBA ASB.....   | 91         |
| AMBA AHB.....   | 92         |
| AMBA APB.....   | 92         |
| AMBA 3.0 and AXI.....   | 92         |
| Summary of ARM CPU Bus Interfaces.....                        | 93         |
| AHB Tutorial.....   | 94         |
| Configuration at Reset.....                                   | 98         |
| Phases of AHB Transfer.....                                   | 99         |
| AHB Arbitration.....  | 99         |
| AHB Address Phase.....  | 101        |
| AHB Data Phase.....   | 104        |
| AHB-Lite.....   | 106        |
| Single-Layer and Multilayer AHB.....                          | 107        |
| ARM926EJ-S Example.....                                       | 107        |
| Interrupt Signals.....  | 111        |
| Instruction and Data Caches.....                              | 111        |
| Tightly Coupled Memory (TCM).....                             | 115        |
| ARM Summary.....  | 118        |
| <b>Chapter 4: Hardware/Software Co-Verification.....</b>      | <b>119</b> |
| History of Hardware/Software Co-Verification.....             | 119        |
| Commercial Co-Verification Tools Appear.....                  | 121        |
| Co-Verification Defined.....                                  | 124        |
| Definition.....   | 124        |
| Benefits of Co-Verification.....                              | 125        |
| Project Schedule Savings.....                                 | 125        |
| Co-Verification Enables Learning by Providing Visibility..... | 127        |
| Co-Verification Improves Communication.....                   | 127        |
| Co-Verification versus Co-Simulation.....                     | 128        |
| Co-Verification versus Co-Design.....                         | 128        |
| Is Co-Verification Really Necessary?.....                     | 129        |
| Co-Verification Methods.....                                  | 129        |
| Native Compiling Software.....                                | 130        |
| Instruction Set Simulation.....                               | 130        |

|  |            |
|--|------------|
| Hardware Stubs.....  | 131        |
| Real-Time Operating System (RTOS) Simulator.....                   | 132        |
| Microprocessor Evaluation Board .....                              | 132        |
| Waveforms, Log Files, and Disassembly .....                        | 133        |
| A Sample of Co-Verification Methods .....                          | 134        |
| Host-Code Mode with Logic Simulation.....                          | 134        |
| Instruction Set Simulation with Logic Simulation.....              | 137        |
| C Simulation .....   | 140        |
| RTL Model of CPU with Software Debugging .....                     | 144        |
| Hardware Model with Logic Simulation.....                          | 147        |
| Evaluation Board with Logic Simulation.....                        | 149        |
| In-Circuit Emulation .....   | 150        |
| FPGA Prototype.....  | 153        |
| Co-Verification Metrics.....                                       | 154        |
| Performance.....   | 155        |
| Verification Accuracy.....   | 155        |
| AHB Arbitration and Cycle Accuracy Issues .....                    | 158        |
| Modeling Summary .....   | 160        |
| Synchronization .....  | 161        |
| Types of Software.....   | 162        |
| Other Metrics .....  | 162        |
| <b>Chapter 5: Advanced Hardware/Software Co-Verification .....</b> | <b>165</b> |
| Direct Access to Simulation Memories.....                          | 165        |
| Memory Optimizations and Performance .....                         | 171        |
| Modes of Synchronization.....                                      | 175        |
| Interprocess Communication .....                                   | 177        |
| Mixing HDL and C Models .....                                      | 180        |
| Implicit Access .....  | 183        |
| Save and Restart.....  | 186        |
| Post-Processing Software Debugging Techniques .....                | 188        |
| Embedded Software Tool Issues.....                                 | 193        |
| Debugging Co-Verification Issues .....                             | 194        |
| <b>Chapter 6: Hardware Verification Environment and</b>            |            |
| <b>Co-Verification .....</b>                                       | <b>197</b> |
| Bus Monitor.....   | 197        |
| Protocol Checking.....   | 207        |
| Aligned Addresses .....  | 207        |

|   |            |
|---|------------|
| Issuing Idle Transfers .....                              | 207        |
| Assertions .....  | 208        |
| Assertion Definitions.....                                | 208        |
| Assertion Approaches .....                                | 210        |
| Declarative Assertions.....                               | 210        |
| Procedural Assertions.....                                | 212        |
| Formal Property Language.....                             | 212        |
| Pseudo-Comment Directives.....                            | 213        |
| Post-Processing Simulation History .....                  | 213        |
| Assertions for Simulation Acceleration and Emulation..... | 214        |
| Testbenches Using Bus Functional Models.....              | 215        |
| Directed Tests .....                                      | 216        |
| Constrained Random Tests .....                            | 217        |
| Testbench Architecture.....                               | 218        |
| Functional Coverage .....                                 | 220        |
| Compliance Suite .....                                    | 221        |
| Software Verification.....                                | 221        |
| Software Print Statements.....                            | 222        |
| Summary.....  | 227        |
| <b>Chapter 7: Methodology for an Example ARM SoC.....</b> | <b>229</b> |
| SoC Methodology Difficulty .....                          | 230        |
| Verification Efficiency .....                             | 231        |
| The Debugging Loop.....                                   | 232        |
| Co-Verification Methodology .....                         | 234        |
| System Initialization and HAL Development .....           | 235        |
| Diagnostics.....  | 235        |
| RTOS and Device Drivers.....                              | 236        |
| Application Software .....                                | 236        |
| Testbench Development.....                                | 236        |
| Three Verification Phases.....                            | 237        |
| Example of ARM Verification Flow .....                    | 239        |
| Block and Subsystem Verification .....                    | 239        |
| Initial System Integration .....                          | 240        |
| Focused Hardware Verification .....                       | 242        |
| Hardware/Software Co-Verification.....                    | 243        |
| System Software Testing.....                              | 244        |
| The Co-Verification Engineer.....                         | 246        |

|                            |            |
|----------------------------|------------|
| Conclusion .....           | 248        |
| Methodology Gridlock ..... | 249        |
| <b>Afterward .....</b>     | <b>253</b> |
| <b>Index .....</b>         | <b>255</b> |



# *Foreword*

This is a remarkable book.

Jason Andrews knows about the hardware and the software. He knows about the people, the tools, and the methodologies in the middle ground between hardware and software.

He can also write, explaining complex things so that you can really understand them.

One of the main reasons this middle area is so complex is there are just too many interacting issues to understand and too many decisions to make.

Jason takes care to enumerate the issues, explain how they interact, and describe the options for dealing with them.

Best of all, he explains which tools and methodologies are applicable for each situation. This is crucial because there are many distinct solutions for the problem, and you cannot possibly use them all. You need to make an informed judgment on what to do when.

Jason has either used or implemented most of these solutions, some of them twice, and he gives a very informed tour of the land and guides you through the possible compromises.

Please note that while Jason and I work for a verification company (Verisity) that would love to sell you verification solutions, this book is decidedly generic. It tells you what works, what does not, and why.

While the title of the book is *Co-Verification of Hardware and Software for ARM SoC Design*, I think this book has wider applicability. In fact, if any of the following apply, then you should begin by reading this book:

## ***Foreword***

---

- You are involved in the verification of products that contain both hardware and software, regardless of whether they are SoC-based or ARM-based.
- You are working on one side of the HW/SW divide, and want to see what the other side looks like.
- You are interested in creating tools for this area.

Yoav Hollander  
Founder and CTO, Verisity Inc.  
July 2004