

1. Record Nr.	UNINA9910458648603321
Autore	Bailey Brian <1959->
Titolo	ESL design and verification [[electronic resource]] : a prescription for electronic system-level methodology // Brian Bailey, Grant Martin, Andrew Piziali
Pubbl/distr/stampa	Amsterdam ; ; Boston, : Morgan Kaufmann, c2007
ISBN	1-281-05353-8 9786611053536 0-08-048883-8
Descrizione fisica	1 online resource (489 p.)
Collana	The Morgan Kaufmann series in systems on silicon
Altri autori (Persone)	MartinGrant (Grant Edmund) PizialiAndrew
Disciplina	621.3815
Soggetti	Systems on a chip - Design and construction Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Front cover; ESL DESIGN AND VERIFICATION; Copyright page; Table of contents; FOREWORD: ESL FROM THE TRENCHES; AUTHORS' ACKNOWLEDGMENTS; ABOUT THE AUTHORS; ABOUT THE CONTRIBUTORS; Chapter 1. WHAT IS ESL?; 1.1 SO, WHAT IS ESL?; 1.2 WHO SHOULD READ THIS BOOK; 1.3 STRUCTURE OF THE BOOK AND HOW TO READ IT; 1.4 CHAPTER LISTING; 1.5 THE PRESCRIPTION; References; Chapter 2. TAXONOMY AND DEFINITIONS FOR THE ELECTRONIC SYSTEM LEVEL; 2.1 TAXONOMY; 2.1.1 Introduction; 2.1.2 Model Taxonomy; 2.1.3 ESL Taxonomy; 2.2 DEFINITIONS; References; Chapter 3. EVOLUTION OF ESL DEVELOPMENT; 3.1 INTRODUCTION 3.2 MOTIVATION FOR ESL DESIGN3.3 TRADITIONAL SYSTEM DESIGN EFFECTIVENESS; 3.4 SYSTEM DESIGN WITH ESL METHODOLOGY; 3.5 BEHAVIORAL MODELING METHODOLOGY; 3.6 BEHAVIORAL MODELING ENVIRONMENTS; 3.7 HISTORICAL BARRIERS TO ADOPTION OF BEHAVIORAL MODELING; 3.8 AUTOMATED IMPLEMENTATION OF FIXED-FUNCTION HARDWARE; 3.9 AUTOMATED IMPLEMENTATION OF PROGRAMMABLE HARDWARE; 3.10 MAINSTREAMING ESL METHODOLOGY; 3.11 PROVOCATIVE THOUGHTS; 3.12 THE

PRESCRIPTION; References; Chapter 4. WHAT ARE THE ENABLERS OF ESL?; 4.1 TOOL AND MODEL LANDSCAPE; 4.2 SYSTEM DESIGNER REQUIREMENTS; 4.3 SOFTWARE TEAM REQUIREMENTS 4.4 HARDWARE TEAM REQUIREMENTS 4.5 WHO WILL SERVICE THESE DIVERSE REQUIREMENTS?; 4.6 FREE OR OPEN SOURCE SOFTWARE; 4.7 SUMMARY; 4.8 THE PRESCRIPTION; References; Chapter 5. ESL FLOW; 5.1 SPECIFICATIONS AND MODELING; 5.2 PRE-PARTITIONING ANALYSIS; 5.3 PARTITIONING; 5.4 POST-PARTITIONING ANALYSIS AND DEBUG; 5.5 POST-PARTITIONING VERIFICATION; 5.6 HARDWARE IMPLEMENTATION; 5.7 SOFTWARE IMPLEMENTATION; 5.8 USE OF ESL FOR IMPLEMENTATION VERIFICATION; 5.9 PROVOCATIVE THOUGHTS; 5.10 SUMMARY; 5.11 THE PRESCRIPTION; References; Chapter 6. SPECIFICATIONS AND MODELING; 6.1 THE PROBLEM OF SPECIFICATION 6.2 REQUIREMENTS MANAGEMENT AND PAPER SPECIFICATIONS 6.3 ESL DOMAINS; 6.4 EXECUTABLE SPECIFICATIONS; 6.5 SOME ESL LANGUAGES FOR SPECIFICATION; 6.6 PROVOCATIVE THOUGHTS: MODEL-BASED DEVELOPMENT; 6.7 SUMMARY; 6.8 THE PRESCRIPTION; References; Chapter 7. PRE-PARTITIONING ANALYSIS; 7.1 STATIC ANALYSIS OF SYSTEM SPECIFICATIONS; 7.2 THE ROLE OF PLATFORM-BASED ESL DESIGN IN PRE-PARTITIONING ANALYSIS; 7.3 DYNAMIC ANALYSIS; 7.4 ALGORITHMIC ANALYSIS; 7.5 ANALYSIS SCENARIOS AND MODELING; 7.6 DOWNSTREAM USE OF ANALYSIS RESULTS; 7.7 CASE STUDY: JPEG ENCODING; 7.8 SUMMARY AND PROVOCATIVE THOUGHTS 7.9 THE PRESCRIPTION; References; Chapter 8. PARTITIONING; 8.1 INTRODUCTION; 8.2 FUNCTIONAL DECOMPOSITION; 8.3 ARCHITECTURE DESCRIPTION; 8.4 PARTITIONING; 8.5 THE HARDWARE PARTITION; 8.6 THE SOFTWARE PARTITION; 8.7 RECONFIGURABLE COMPUTING; 8.8 COMMUNICATION IMPLEMENTATION; 8.9 PROVOCATIVE THOUGHTS; 8.10 SUMMARY; 8.11 THE PRESCRIPTION; References; Chapter 9. POST-PARTITIONING ANALYSIS AND DEBUG; 9.1 ROLES AND RESPONSIBILITIES; 9.2 HARDWARE AND SOFTWARE MODELING AND CO-MODELING; 9.3 PARTITIONED SYSTEMS AND RE-PARTITIONING; 9.4 PRE-PARTITIONED MODEL COMPONENTS; 9.5 ABSTRACTION LEVELS 9.6 COMMUNICATION SPECIFICATION

Sommario/riassunto

Visit the authors' companion site! <http://www.electronicssystemlevel.com/> - Includes interactive forum with the authors! Electronic System Level (ESL) design has mainstreamed - it is now an established approach at most of the world's leading system-on-chip (SoC) design companies and is being used increasingly in system design. From its genesis as an algorithm modeling methodology with 'no links to implementation', ESL is evolving into a set of complementary methodologies that enable embedded system design, verification and debug through to the hardware and software implementation
