1. Record Nr. UNINA9910960804503321

Autore	Alexander Perry
Titolo	System-level design with Rosetta / / Perry Alexander
Pubbl/distr/stampa	San Francisco, : Elsevier/Morgan Kaufman, c2007
ISBN	1-281-01500-8 9786611015008 0-08-049837-X
Edizione	[1st ed.]
Descrizione fisica	1 online resource (375 p.)
Collana	Morgan Kaufmann series in systems on silicon
Disciplina	621.3815
Soggetti	Integrated circuits - Computer-aided design Microelectronics
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 (p. 339-341) and index.
Nota di contenuto	Front Cover; System-Level Design with Rosetta; Copyright Page; Contents; Acknowledgments; Foreword; Preface; Part I: Introduction; Chapter 1. Introduction; 1.1 What is System-Level Specification?; 1.2 Rosetta's Design Goals; 1.3 Anatomy of a Specification; 1.4 Learning Rosetta; Part II: The Expression Language; Chapter 2. Items, Values, Types, and Declarations; 2.1 Labels, Values, and Types; 2.2 Item Declarations and Type Assertions; 2.3 Universal Operations; Chapter 3. Expressions; 3.1 Atomic Expressions; 3.2 Function Application; 3.3 Operator Application; 3.4 If Expressions 3.5 Case Expressions3.6 Let Expressions; 3.7 Compound Expressions; Chapter 4. Elemental Types; 4.1 The Boolean Type; 4.2 The Number Types; 4.3 The Character Type; 4.4 The Element Type; 4.5 The Top and Bottom Types; 4.6 Element Literals; 4.7 Operator Result Types; Chapter 5. Composite Types; 5.1 Type Formers; 5.2 Set Types; 5.3 Multiset Types; 5.4 Sequence Types; Chapter 6. Functions; 6.1 Direct Function Definition; 6.2 Function Values and Function Types; 6.3 Evaluating Functions; 6.4 Universally Quantified Parameters; Chapter 7. Higher-Order Functions; 7.1 Domain, Range, and Return Functions 7.2 Alternate Higher-Order Function Notation7.3 Minimum and Maximum; 7.4 Quantifiers and Comprehension; 7.5 Sequences and Higher-Order Functions; 7.6 Function Inclusion and Composition;

Chapter 8. User-Defined Types; 8.1 Defining New Types; 8.2 Defining Types By Extension; 8.3 Defining Types By Comprehension; 8.4 Defining Constructed Types; 8.5 Functions as Type Definition Tools; Part III: The Facet Language; Chapter 9. Facet Basics; 9.1 A First Model - An AM Modulator; 9.2 Composing Models - Adding Constraints; 9.3 Combinational Circuits - A Simple Adder; 9.4 Defining State - A 2-bit Counter
9.5 Defining Structure - A 2-bit Adder9.6 Specification Reuse - Using Packages; 9.7 Abstract Specification - Architecture Definition; Chapter 10. Defining Facets; 10.1 Direct Facet Definition; 10.2 Separable Definitions; 10.3 Facets and Hardware Description Languages; 10.4 Facet Styles; 10.5 Scoping Rules; 10.6 Basics of Facet Semantics; Chapter 11. Packages, Libraries, and Components; 11.1 Packages; 11.2 Libraries; 11.3 Components; Part IV: Domains and Interactions; Chapter 12. Domains; 12.1 Elements of a Domain; 12.2 The Standard Domains; 12.3 Domains and Facet Types; Chapter 13. Reflection
13.1 Template Expressions and AST Structures13.2 Interpreting AST Structures; 13.3 Domain Declarations; 13.4 Defining Engineering Domains; 13.5 Defining New Model-of-Computation Domains; 13.6 Defining New Unit-of-Semantics Domains; 13.7 Defining Ticked and Dereferencing Expressions; 13.8 Consistent Domain Extension; Chapter 14. The Facet Algebra; 14.1 Facet Products and Sums; 14.2 Facet Homomorphism and Isomorphism; 14.3 Conditional Expressions; 14.4 Let Expressions; 14.5 Higher-Order Facets; Chapter 15. Domain Interactions; 15.1 Projection Functions, Functors, and Combinators
15.2 Defining Interactions

Sommario/riassunto

The steady and unabated increase in the capacity of silicon has brought the semiconductor industry to a watershed challenge. Now a single chip can integrate a radio transceiver, a network interface, multimedia functions, all the ""glue"" needed to hold it together as well as a design that allows the hardware and software to be reconfigured for future applications. Such complex heterogeneous systems demand a different design methodology. A consortium of industrial and government labs have created a new language and a new design methodology to support this effort. Rosetta permits designers t