Preface

Purpose of the Book

The primary objective of this book is to equip readers for entry-level roles as integrated circuit (IC) designers in the industry and as hardware design researchers in academia. It seeks to foster a deep understanding of the field by introducing the industry IC design flow and offering tape-out or pseudo tape-out projects for hands-on practice, facilitating project-based learning (PBL) experiences.

Industry IC Design Flow

The IC design industry places significant emphasis on the intricate IC design flow, encompassing a comprehensive suite of electronic design automation (EDA) tools for tasks such as simulation, synthesis, layout, timing analysis, and more, along with established design and verification methodologies. This book commences with an overview of the industry IC design flow, with a primary focus on register-transfer level (RTL) design, the automation of simulation and verification, and system-on-chip (SoC) integration. To build connections between RTL design and physical hardware, field-programmable gate array (FPGA) synthesis and implementation is utilized to illustrate the hardware description and performance evaluation.

Part III of the book includes three appendices/tutorials on commonly used toolsets and industry design and simulation approaches. These encompass the Vim editor, Siemens ModelSim simulator, AMD Vivado, as well as guidance on the structured project directory, fundamental Unix/Linux commands, and the effective automation of simulations using TCL scripts. All three appendices/tutorials are readily accessible through the accompanying Git repository, ensuring that readers can easily utilize supplementary materials and resources to enhance their learning experience.

Project-Based Learning

RTL design is pivotal as it not only describes hardware functionality but also directly impacts chip performance, encompassing aspects such as area, speed, power efficiency, and successful chip fabrication within specified timing constraints. Hence, the second objective of this book is to provide readers with practical, hands-on experience through tape-out or pseudo tape-out experiments, labs, and projects. These activities are centered on coding format, industry design rules (synthesizable Verilog designs, clock domain crossing, etc.), and commonly used bus protocols (arbitration, handshaking, etc.), as well as established design methodologies for widely adopted hardware components, including counters, timers, finite state machines (FSMs), I2C, single/dual-port and ping-pong buffers/register files, FIFOs, floating-point units (FPUs), numerical hardware (Fourier transform, matrix-matrix multiplication, etc.), direct memory access (DMA), image processing designs, neural networks, and more.

The project-based learning unfolds systematically throughout the book. For instance, it commences with the construction of a timer featuring two-level counters in Chapter 6, followed by the design of FSMs in Chapter 7. Subsequently, it progresses to the creation of an I2C design, a widely used serial bus protocol in modern chipsets, achieved through the integration of an FSM and a timer-based datapath in Chapter 8. It is imperative to underscore that most of the design projects showcased in this book are open-source in the accompanying Git repository. This not only enables readers to utilize these projects but also encourages them to expand and experiment with the designs.

The Contents at a Glance

The book is structured into three distinct parts as below.

Part I: Fundamentals of IC Design and Simulation with Verilog HDL (Chapters 1–8)

Part I is dedicated to introducing the industry IC design flow and encompasses fundamental knowledge and skills for RTL design and simulation using Verilog hardware description language (HDL). It provides practical examples aimed at enhancing the learning experience and skill development.

- **Chapter 1** functions as an introduction to the IC industry, offering background information and the essential knowledge required for RTL design with Verilog HDL.

- **Chapter 2** provides a concise overview of the industry IC design flow, encompassing both the front-end (specification, RTL design, verification, SoC integration, etc.) and back-end procedures (synthesis, layout, timing check, etc.), fabrication stages, and packaging and testing.

- **Chapters 3–5** explore the foundational aspects of Verilog, with an emphasis on creating synthesizable designs and establishing a basic simulation environment. The EDA tools employed in these chapters encompass Vim editor and Siemens ModelSim.

PBL 1–9 showcase the design of fundamental combinational and sequential circuits, the hierarchical design methodology, and the creation of automated simulation environments.

- **Chapter 6** centers on the synthesis results using AMD Vivado, establishing a connection between Verilog descriptions and specific hardware.
PBL 10–16 are provided to facilitate creating a linkage between Verilog code and their described hardware circuits.

- **Chapter 7** covers basic FSM design and simulation, serving as a fundamental timing controller for digital ICs.
PBL 17 offers an example of utilizing an FSM template for detecting a specific digit sequence.

- **Chapter 8** introduces design integration involving FSMs and datapaths, presenting two design cases: I2C and master-slave bus interfaces. This chapter serves as a versatile template, demonstrating how digital circuits can be integrated with timing controllers and well-structured datapaths.
PBL 18–21 are provided to demonstrate FSMD designs across various hardware architectures, alongside advanced design methodologies like pipeline and parallel computing.

Part II: Advanced IC Design and Integration (Chapters 9–12)

Part II provides a comprehensive exploration of various design architectures (streaming design, iterative design, pipeline and parallel computing, etc.), timing considerations/constraints employing RTL design, and SoC integration. This part illustrates these concepts through a variety of design examples including numerical hardware accelerators and neural engines. Additionally, Part II explores the prevalent SoC architecture utilized in the industry, focusing particularly on AMBA AXI (Advanced Microcontroller Bus Architecture - Advanced eXtensible Interface), alongside other commonly-used design components located on SoCs such as DMA and image/video processing units.

- **Chapter 9** centers on numerical hardware design and integration, leveraging the FPUs provided by this book. Examples featured in this chapter encompass a floating-point (FP) matrix-matrix adder, *axpy* computation, and a fundamental *ddot* design. These examples represent essential hardware accelerators frequently employed in scientific computing.
PBL 22–27 demonstrate RTL design and simulation for register files, as well as numerical hardware components like FP multiplication-addition circuits and matrix-matrix multipliers.

- **Chapter 10** explores high-performance design structures with a particular emphasis on streaming and iterative designs. This chapter illustrates

how specialized hardware designs can be employed to improve data processing efficiency and/or providing cost-effective solutions to meet a range of hardware design specifications.
PBL 28–30 showcase streaming and iterative design options tailored for various numerical applications, including iterative *ddot* design and Fourier transform.

- **Chapter 11** presents an introduction to timing constraints, emphasizing their critical importance in high-speed design and their substantial influence on the success or failure of timing analysis and chip fabrication.
  PBL 31–41 focus on exploring designs with attainable MOF (maximum operational frequency) and introduce design rules and options, including signal management across asynchronous clock domains and the implementation of high-speed designs using pipeline structures.

- **Chapter 12** is dedicated to SoC integration, particularly emphasizing the AMBA AXI bus, a leading bus protocol in the industry. It introduces various typical design components frequently employed in SoCs, including DMA, image processing units, and neural network engines.
  PBL 42–50 exemplify hardware design architectures and methodologies, encompassing topics such as floating-to-fixed point conversion, approximate design, hardware reuse, and more. These are tailored to meet various design specifications, addressing concerns such as accuracy, hardware resource and power constraints, and computational latency bounds. Industry-adopted projects, including DMA arbitration, image processing units, and sigmoid neural networks, are utilized in these projects to provide practical insights and applications.

Part III: Tutorials on EDA Tools and Essential Skills Related to IC Design and Simulation

This part provides tutorials that introduce EDA tools and essential knowledge and skills for IC design and simulation.

- **Appendix/Tutorial A** introduces the structured project directory and foundational Unix/Linux commands, accompanied by an exhaustive tutorial for proficiently utilizing the Vim editor.

- **Appendix/Tutorial B** furnishes a tutorial for using the Siemens ModelSim simulator, covering navigation of the GUI interface and effective automation of simulations through TCL scripts.

- **Appendix/Tutorial C** offers a comprehensive tutorial on using AMD Vivado for FPGA implementations, with a particular focus on RTL analysis for synthesized circuits and performance evaluation. This includes assessing hardware resource utilization, power consumption, and computational speed.

The Intended Audience

The book provides an overview of the IC design flow, with a specific focus on front-end aspects, covering design specification, RTL design, simulation, SoC integration, FPGA synthesis and implementation, and performance evaluation. It caters to a diverse readership, including junior and senior undergraduate students, as well as graduate students pursuing degrees in electrical engineering, computer engineering, computer science, and related fields. The target audience is expected to have a basic understanding of Boolean Algebra and Karnaugh Maps, as well as prior familiarity with digital logic components such as AND/OR gates, latches, and flip-flops.

This book also serves as an invaluable resource for entry-level RTL designers and verification engineers who are embarking on their journey in application-specific IC (ASIC) and FPGA design industry, offering essential knowledge and practical insights for their roles in the field.

How to Use This Book

This book is thoughtfully designed to encompass the material for one or two courses at the junior and senior undergraduate levels, and within graduate studies. To fully derive value from the content, it is recommended that readers have completed a prerequisite course covering topics such as Boolean Algebra, Karnaugh Maps, and fundamental digital logic. Below are two examples of teaching courses using this textbook.

Introduction to IC Design Flow/Digital System Design

This course is designed specifically for junior and senior undergraduate students, as well as those pursuing graduate studies in digital system/IC design and simulation. The course materials covered in Part I and Part III of the book, spanning Chapters 2–8 and Appendixes A–C, provide a solid foundation for instruction. Moreover, a series of PBL examples, numbered 1–21, have been included to complement the training lectures and facilitate practical hands-on labs and projects. Further teaching suggestions are available on the accompanying GitHub repository.

Advanced Digital System Design

This course is specifically tailored for senior undergraduate and graduate students who are pursuing studies in high-performance hardware design and SoC integration. The course materials, primarily covered in Part II and Part III of the book, spanning Chapters 1 and 9–12, along with Appendixes A–C, establish a robust foundation for instruction. For students without prior Verilog coding experience, Chapters 3–5 can be utilized to introduce the fundamental Verilog syntax for RTL design and simulation. Moreover, a series of practical PBL examples, numbered 6–16 and 22–50, can be integrated to complement the training lectures and provide hands-on labs and projects. Further teaching suggestions are available on the accompanying GitHub repository.

Supplements and Resources (Git Repository)

Numerous supplementary resources are accessible on the Git repository: `https://github.com/LBL-ICS/IC-Design/`, encompassing:

- The Verilog designs for experiments, labs, and projects presented in the book.
- Appendices A–C containing tutorials on the recommended project directory, Vim Editor, Siemens ModelSim, AMD Vivado, and other valuable commands and scripts for IC design projects.

Instructors teaching IC design-related courses can also access the following additional resources:

- Lecture slides covering all chapters.
- Extensive teaching suggestions for two courses: “Introduction to IC Design Flow” and “Advanced Digital System Design”.
- Verilog design and simulation materials for experiments, labs, and projects outlined in the book.
- Thorough solution manual for exercises/assignments.
- Two examination tests to facilitate course assessment.

Acknowledgments

The author expresses deep gratitude to Mario Vega, his former student and research assistant at the University of Houston Clear Lake, for his invaluable contribution in providing the foundational FPUs generated using Chisel hardware construction language (HCL). Mario, currently a researcher at Lawrence Berkeley National Laboratory, played a pivotal role in enhancing the content of this book.

Furthermore, the author acknowledges the research endeavors at Berkeley Laboratory since 2022 which have greatly expanded the book's scope to include code generator designs using Chisel HCL. Chisel, a domain-specific language developed by the University of California, Berkeley, provides an efficient solution for generating Verilog code for hardware implementations. This advancement has facilitated the integration of FPUs into the book's content in a highly productive manner.

Bio

Dr. Xiaokun Yang currently serves as an associate professor at the College of Science and Engineering at the University of Houston Clear Lake, located in Houston, Texas. Additionally, since 2022, he has held the role of affiliate faculty at Lawrence Berkeley National Laboratory, situated in Berkeley, California. Dr. Yang earned his Ph.D. in the Department of Electrical and Computer Engineering at Florida International University (FIU) in the spring of 2016. He also brings extensive industry experience, having worked as a senior ASIC design and verification engineer at Advanced Micro Devices (AMD) and China Electronic Corporation (CEC) during 2007–2012. Dr. Yang's research interests center on specialized hardware design and acceleration for future high-performance computing, design automation for numerical hardware and machine learning, and advanced high-performance SoC architecture.

Errata

This book has been self-prepared, encompassing all elements including text, tables, figures, code, indexing, and formatting. Acknowledging the possibility of errors, an accompanying Git repository provides an updated errata sheet and serves as a platform for reporting any identified mistakes.

X. Yang
January 2024
Houston, TX