Newnes

INCLUDES

FREE
NEWNES ONLINE
MEMBERSHIP

DIGITAL SYSTEMS DESIGN WITH FPGAs AND CPLDs

- VHDL coding used throughout the book
- Extensive worked examples
- Case studies highlight interfacing analog and digital

Ian Grout

Digital Systems Design with FPGAs and CPLDs

Ian Grout

ELSEVIER

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

Newnes is an imprint of Elsevier

Newnes

Newnes is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

Library of Congress Cataloging-in-Publication Data
Grout, Ian.
Digital systems design with FPGAs and CPLDs / Ian Grout.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-7506-8397-5 (alk. paper) 1. Digital electronics. 2. Digital circuits — Design and construction. 3. Field programmable gate arrays. 4. Programmable logic devices. I. Title.
TK7868.D5.G76 2008
621.381—dc22

2007044907

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

For information on all Newnes publications
visit our Web site at www.books.elsevier.com

Printed in the United States of America
08 09 10 11 12 13 10 9 8 7 6 5 4 3 2 1

Digital Systems Design with FPGAs and CPLDs

To my family, but especially to my parents and to Jane.

system

• **noun 1** A set of things working together as parts of a mechanism or an interconnecting network.

Oxford Dictionary of English

Preface

In days gone by, life for the electronic circuit designer seems to have been easier. Designs were smaller, ran at a slower speed, and could easily fit onto a single small printed circuit board. An individual designer could work on a problem and designs could be specified and developed using paper and pen only. The circuit schematic diagrams that were required could be rapidly drawn on the back of an envelope.

Struck by the success of the early circuit designs, customers started to ask for smaller, faster, and more complex circuits—and at a lower cost. The designers started to work on solving such problems, which has led to the rapidly expanding electronics industry that we have today. Driven by the demand from the customer, new materials and fabrication processes have been developed, new circuit design methodologies and design architectures have taken over many of the early traditional design approaches, and new markets for the circuits have evolved.

So how is the design problem tackled today? This is not an easy question to answer, and there is more than one way to develop an electronic circuit solution to any given problem. However, the design process is no longer the activity of a single individual. Rather, a team of engineers is involved in the key engineering activities of design, fabrication (manufacture), and test. All activities now involve the extensive use of computing resources, requiring the efficient use of software tools to aid design (electronic design automation, EDA and computer aided design, CAD), fabrication (Computer Aided Manufacture, CAM), and test (Computer Aided Test, CAT). The circuit is no longer a unique and isolated entity. Rather, it is part of a larger system. Increasingly, much of the design work is undertaken at the system level . . . at a suitably high level of design abstraction required to reduce design time and increase the designer efficiency. However, when it comes to the design detail, the correctly specified system must also work at the basic electric voltage and current level. How to go from an effective system-level specification to an efficient and working circuit implementation requires the skills of good designers who are aided by good design tools.

For the electronic circuit designer at an early stage in the design process, whether to implement the required circuit functionality using analogue circuit techniques or digital circuit techniques must be decided. However, sometimes the choice will have already been made, and increasingly a digital solution is the preferred choice. The wide use of digital signal processing (DSP) techniques facilitates complex operations that can provide superior performance to an analogue circuit equivalent; indeed some cannot be performed in analogue. Traditionally, DSP functions have been implemented using software programs written to operate on a target processor. The microprocessor (μP), microcontroller (μC), and digital signal processor provide the necessary digital circuits, in integrated circuit (IC) form, to implement the required functions. In fact, these processors are to be found in many everyday embedded electronics that we take for granted. This book could not have been written without the aid of an electronic system incorporating a microprocessor running a software operating system that in turn runs the word processor software.

Increasingly, the functions that have been traditionally implemented in software running on a processor-based digital system in the DSP world and many control applications are being evaluated in terms of performance that can be achieved in software. In many cases, the software solution will be slower than is desired, and the basic nature of the software programmed system means that this speed limitation cannot be overcome. The way to overcome the speed limitation is to perform the required operations in hardware designed for a particular application. However, custom hardware solutions will be expensive to acquire.

If there were a way to obtain the power of programmability with the power of hardware speed, then this would be provide a significant way forward.

Fortunately, programmable logic provides the power of programmability with the power of hardware speed by providing an IC with built-in digital electronic circuitry that is configured by the user for a particular application. Many devices can be reconfigured for different applications. Today, two main types of programmable logic ICs are commonly used: the field programmable gate array (FPGA) and complex programmable logic device (CPLD).

Therefore, it is possible to implement a complex digital system that can be developed and the functionality changed or enhanced using either a processor running a software program or programmable logic with a specific hardware configuration.

For an end-user, the functionality of both types of system will be the same—the design details are irrelevant to the end-user as long as the functionality of the unit is correct. In this book, to provide consistency and to differentiate between the processor and programmable logic, the following terminology will be used:

- A *processor* (microprocessor, μP; microcontroller, μC; or digital signal processor, DSP) will be *programmed* for a particular application using a *software programming language* (SPL).
- *Programmable logic* (field programmable gate array, FPGA; simple programmable logic device, SPLD; or complex programmable logic device, CPLD) will be *configured* using a *hardware description language* (HDL).

The aim of this book is to provide a reference source with worked examples in the area of electronic circuit design using programmable logic. In particular, field programmable gate arrays and complex programmable logic devices will be presented and examples of such devices provided.

The choice whether to use a software-programmed processor or hardware-configured programmable logic device is not a simple one, and many decisions figure into evaluating the pros and cons of a particular implementation prior to making a final decision. This book will provide an insight into the design capabilities and aspects relating to the design decisions for programmable logic so that an informed decision can be made.

The book is structured as follows:

Chapter 1 will introduce the types of programmable logic device that are available today, their differing architectures, and their use within electronic system design. Additionally, the terminology used in this area will be presented with the aim of demystifying the jargon that has evolved.

Chapter 2 will provide a background into the area of electronic systems design, the types of solutions that may be developed, and the decisions that will need to be made in order to identify the right technology choice for the design implementation. Typical design flows will be introduced and discussed for the different technologies.

Chapter 3 will introduce the design of printed circuit boards (PCBs). These provide the mechanical and electrical base onto which the electronic components will be mounted. The correct design of the PCB is essential to ensure that the electronic circuit can be realized (implemented) to operate to the correct specification (power supply voltage, thermal [heat] dissipation, digital clock frequency, analogue and digital circuit elements, etc.) and to ensure that the different electronic circuit components interact with each other correctly and do not provide unwanted effects. A correctly designed PCB will allow the circuit to perform as intended. A badly designed PCB will prevent the circuit from working altogether.

Chapter 4 will discuss the different programming languages that are used to develop digital designs for implementation in either a processor (software-programmed microprocessor, microcontroller, or digital signal processor) or in programmable logic (hardware-configured FPGA or CPLD). The main languages used will be introduced and examples provided. For programmable logic, the main hardware description languages used are Verilog®-HDL and VHDL (VHSIC Hardware Description Language). These are IEEE (Institute of Electrical and Electronics Engineers) standards, universally used in both education and industry.

Chapter 5 will introduce digital logic design principles. A basic understanding of the principles of digital circuit design, such as Boolean Logic, Karnaugh maps, and counter/state machine design will be expected. However, a review of these principles will be provided for designs in schematic diagram form and presented such that the design functionality may be mapped over a VHDL description in Chapter 6.

Chapter 6 will introduce VHDL as one of the IEEE standard hardware description languages available to describe digital circuit and system designs in an ASCII text-based format. This description can be simulated and synthesized. (Simulation will validate the design operation, and synthesis will translate the text-based description into a circuit design in terms of logic gates and the interconnections between the basic logic gates. The gates and gate connections are commonly referred to as the netlist.) The design examples provided in schematic diagram form in Chapter 5 will be revisited and modeled in VHDL.

Chapter 7 will introduce the development of digital signal processing algorithms in VHDL and the synthesis of the VHDL descriptions to target programmable logic (both FPGA and CPLD). Such algorithms include digital filters (low-pass, high-pass, and band-pass), digital PID (proportional plus integral plus derivative) control algorithms, and the FFT (fast Fourier transform, an efficient implementation of the discrete Fourier transform, DFT).

Chapter 8 will discuss the interfacing of programmable logic to what is commonly referred to as the real world. This is the analogue world that we live in, and such interfacing requires both the acquisition (capture) and the generation of analogue signals. To enable this, the digital programmable logic device will require an interface to the analogue world. For analogue signals to be captured and analyzed in digital, an analogue-to-digital converter (ADC) will be required. For analogue signals to be generated from the digital, a digital-to-analogue converter (DAC) will be required.

In this book, the convention used for the word *analogue* will use the -ue at the end of the word, unless a particular name already in use is referred to spelled as *analog*.

Chapter 9 will introduce the testing of the electronic system. In this, failure mechanisms in hardware and software will be introduced, and the need for efficient and cost-effective test programs from the prototyping phase of the design through high-volume manufacture and in-system testing.

Chapter 10 will introduce the increasing need to develop programmable logic–based designs at a high level of abstraction using behavioral descriptions of the system functionality, and the increasing requirements to enable the synthesis of these high-level designs into logic. With reference to a design flow taking a digital design developed in MATLAB® or Simulink® through a VHDL code equivalent for implementation in FPGA or CPLD technology, the synthesis of digital control system algorithms modeled and simulated in Simulink® will be translated into VHDL for implementation in programmable logic.

Throughout the book, the HDL examples provided and evaluated can be implemented within programmable logic–based circuits that may be designed by the user in addition to the PCB design examples that are provided. These examples have been developed to form the basis of laboratory experiments that can be used to accompany the text.

With the broad range of subject material and examples, a feature of the book is its potential for use in a range of learning and teaching scenarios. For example:

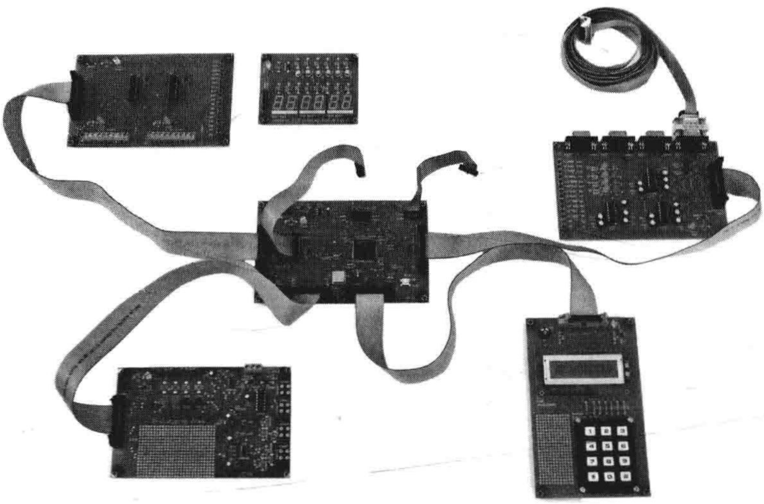
1. As an introduction to design of electronic circuits and systems using programmable logic. This would allow for the design approaches, programmable logic architectures, simulation, synthesis, and the final configuration of an FPGA or CPLD to be undertaken. It would also allow for investigation into the most appropriate HDL coding styles and device implementation constraints to be undertaken.

2. As an introduction to hardware description languages, in particular VHDL, allowing for case study designs to be developed and implemented within programmable logic. This would allow for VHDL code developers to see the

code working on real devices and to enable additional testing of the electronic circuit with such equipment as oscilloscopes and spectrum analyzers.

3. As an introduction to the design of printed circuit boards, in particular mixed-signal designs (mixed analogue and digital). This would allow issues relating to the design of the printed circuit board to be investigated and designs developed, fabricated, and tested.
4. As an introduction to digital signal processing algorithm development. This would allow the basics of DSP algorithms and their implementation in hardware on FPGAs and CPLDs to be investigated through the medium of VHDL code development, simulation, and synthesis.

The VHDL examples can be downloaded and run on the hardware prototyping arrangement that can be built by the reader using the designs provided in the book. This hardware arrangement is centered on a Xilinx® Coolrunner™-II CPLD on which to prototype the digital logic ideas, along with a set of input/output (I/O) boards. The full set of boards is shown in the figure below.



This arrangement consists of five main system boards and an optional seven-segment display board. The appendices and design schematics are available at the author's Web site for this book (refer to http://books.elsevier.com/companions/9780750683975 and follow the hyperlink to the author's site).

Abbreviations

A

AC	alternating current
ADC	analogue-to-digital converter
ALU	arithmetic and logic unit
AM	amplitude modulation
AMD	advanced micro devices
AMS	analogue and mixed-signal
AND	logical AND operation on two or more digital signals
ANSI	American National Standards Institute
AOI	automatic optical inspection
ASCII	American Standard Code for Information Interchange
ASIC	application-specific integrated circuit
ASP	analogue signal processor
ASSP	application-specific standard product
ATA	AT attachment
ATE	AT equipment
ATPG	AT program generation
AWG	arbitrary waveform generator
	American wire gauge
AXI	automatic X-ray inspection

B

BASIC	Beginner's All-purpose Symbolic Instruction Code
BCD	binary coded decimal
BGA	ball grid array
BiCMOS	bipolar and CMOS
BIST	built-in self-test

bit	*b*inary dig*it*
BJT	bipolar junction transistor
BNC	bayonet Neill-Concelman connector
BPF	band-pass filter
BSDL	boundary scan description language
BS(I)	British Standards (Institution)
BST	boundary scan test

C

CAD	computer-aided design
CAE	computer-aided engineering
CAM	computer-aided manufacture
CAT	computer-aided test
CBGA	ceramic BGA
CD	compact disk
CE	chip enable
CERDIP	ceramic DIP
CERQUAD	ceramic quadruple side
CIC	cascaded integrator comb
CISC	complex instruction set computer
CLB	configurable logic block
CLCC	ceramic leadless chip carrier
	ceramic leaded chip carrier
CMOS	complementary metal oxide semiconductor
COTS	commercial off-the-shelf
CPGA	ceramic PGA
CPLD	complex PLD
CPU	central processing unit
CQFP	ceramic quad flat pack
CS	chip select
CSOIC	ceramic SOIC
CSP	chip scale packaging
CSSP	customer specific standard product
CTFT	continuous-time Fourier transform
CTS	clear to send
CUT	circuit under test

D

DAC	digital-to-analogue converter
DAE	differential and algebraic equation
DAQ	data acquisition
dB	decibel
DBM	digital boundary module
DC	direct current
DCD	data carrier detected
DCE	data communication equipment
DCI	digitally controlled impedance
DCPSS	DC power supply sensitivity
DDC	direct digital control
DDR	double data rate
DDS	direct digital synthesis
DfA	design for assembly
DfD	design for debug
DFF	D-type flip-flop
DfM	design for manufacturability
DfR	design for reliability
DfT	design for testability
DFT	discrete Fourier transform
DfX	design for X
DfY	design for yield
DIB	device interface board
DIL	dual in-line
DIMM	dual in-line memory module
DIP	dual in-line package
DL	defect level
DMM	digital multimeter
DNL	differential nonlinearity
DoD	U.S. Department of Defense
DPLL	digital PLL
dpm	defects per million
DR	data register
DRAM	dynamic RAM
DRC	design rules checking

DRDRAM	direct Rambus DRAM
DSM	deep submicron
DSP	digital signal processing
	digital signal processor
DSR	data set ready
DTE	data terminal equipment
DTFT	discrete-time Fourier transform
DTR	data terminal ready
DUT	device under test
DVD	digital versatile disk

E

EC	European Commission
ECL	emitter coupled logic
ECU	electronic control unit
EDA	electronic design automation
EDIF	electronic design interchange format
EHF	extremely high frequency
EIAJ	Electronic Industries Association of Japan
ELF	extremely low frequency
EMC	electromagnetic compatibility
EMI	electromagnetic interference
ENB	effective number of bits
EOC	end of conversion
EOS	electrical overstress
EEPROM	electrically erasable PROM
E^2EPROM	electrically erasable PROM
EPROM	erasable PROM
ERC	electrical rules checking
ESD	electrostatic discharge
ESIA	European Semiconductor Industry Association
ESL	electronic system level
ESS	environmental stress screening
EU	European Union
EX-NOR	NOT-EXCLUSIVE-OR
EX-OR	logical EXCLUSIVE-OR operation on two or more digital signals

F

F	Farad
FA	failure analysis
FBGA (FPBGA)	fine pitch ball grid array
FCC	Federal Communications Commission (USA)
FET	field effect transistor
FFT	fast Fourier transform
FIFO	first-in, first-out
FIR	finite impulse response
FM	frequency modulation
FPAA	field programmable analogue array
FPGA	field programmable gate array
FPT	flying probe tester
FR-4	flame retardant with approximate dielectric constant of 4
FRAM	ferromagnetic RAM
FSM	finite state machine
FT	functional tester

G

GaAs	gallium arsenide
GAL	generic array of logic
GDSII	Graphic Data System II stream file format
GND	ground
GPIB	general purpose interface bus
GTL	Gunning transceiver logic
GTO	gate turn-off thyristor
GUI	graphical user interface

H

HBM	human body model
HBT	heterojunction bipolar transistor
HDIP	hermetic DIP
HDL	hardware description language
HF	high frequency
HPF	high-pass filter
HSTL	high-speed transceiver logic
HTML	hyphertext markup language