FPGA-based Implementation of Signal Processing Systems

FPGA-based Implementation of Signal Processing Systems

Second Edition

Roger Woods
Queen's University, Belfast, UK

John McAllister
Queen's University, Belfast, UK

Gaye Lightbody
University of Ulster, UK

Ying Yi
SN Systems – Sony Interactive Entertainment, UK

WILEY

This edition first published 2017

Registered Offices
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA
John Wiley & Sons, Ltd., The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Office
The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Library of Congress Cataloging-in-Publication Data

Names: Woods, Roger, 1963- author. | McAllister, John, 1979- author. |
Lightbody, Gaye, author. | Yi, Ying (Electrical engineer), author.
Title: FPGA-based implementation of signal processing systems / Roger Woods,
John McAllister, Gaye Lightbody, Ying Yi.
Description: Second editon. | Hoboken, NJ : John Wiley & Sons Inc., 2017. |
Revised edition of: FPGA-based implementation of signal processing systems /
Roger Woods … [et al.]. 2008. | Includes bibliographical references and index.
Identifiers: LCCN 2016051193 | ISBN 9781119077954 (cloth) | ISBN 9781119077978 (epdf) |
ISBN 9781119077961 (epub)
Subjects: LCSH: Signal processing–Digital techniques. | Digital integrated
circuits. | Field programmable gate arrays.
Classification: LCC TK5102.5 .F647 2017 | DDC 621.382/2–dc23 LC record available at
https://lccn.loc.gov/2016051193

Cover Design: Wiley
Cover Image: © filo/Gettyimages;
(Graph) Courtesy of the authors

Set in 10/12pt WarnockPro by Aptara Inc., New Delhi, India

10 9 8 7 6 5 4 3 2 1

The book is dedicated by the main author to his wife, Pauline, for all for her support and care, particularly over the past two years.

The support from staff from the Royal Victoria Hospital and Musgrave Park Hospital is greatly appreciated.

Contents

Preface

DSP and FPGAs

Digital signal processing (DSP) is the cornerstone of many products and services in the digital age. It is used in applications such as high-definition TV, mobile telephony, digital audio, multimedia, digital cameras, radar, sonar detectors, biomedical imaging, global positioning, digital radio, speech recognition, to name but a few! The evolution of DSP solutions has been driven by application requirements which, in turn, have only been possible to realize because of developments in silicon chip technology. Currently, a mix of programmable and dedicated system-on-chip (SoC) solutions are required for these applications and thus this has been a highly active area of research and development over the past four decades.

The result has been the emergence of numerous technologies for DSP implementation, ranging from simple microcontrollers right through to dedicated SoC solutions which form the basis of high-volume products such as smartphones. With the architectural developments that have occurred in field programmable gate arrays (FPGAs) over the years, it is clear that they should be considered as a viable DSP technology. Indeed, developments made by FPGA vendors would support this view of their technology. There are strong commercial pressures driving adoption of FPGA technology across a range of applications and by a number of commercial drivers.

The increasing costs of developing silicon technology implementations have put considerable pressure on the ability to create dedicated SoC systems. In the mobile phone market, volumes are such that dedicated SoC systems are required to meet stringent energy requirements, so application-specific solutions have emerged which vary in their degree of programmability, energy requirements and cost. The need to balance these requirements suggests that many of these technologies will coexist in the immediate future, and indeed many hybrid technologies are starting to emerge. This, of course, creates a considerable interest in using technology that is programmable as this acts to considerably reduce risks in developing new technologies.

Commonly used DSP technologies encompass software programmable solutions such as microcontrollers and DSP microprocessors. With the inclusion of dedicated DSP processing engines, FPGA technology has now emerged as a strong DSP technology. Their key advantage is that they enable users to create system architectures which allow the resources to be best matched to the system processing needs. Whilst memory resources are limited, they have a very high-bandwidth, on-chip capability. Whilst the prefabricated aspect of FPGAs avoids many of the deep problems met when developing

SoC implementations, the creation of an efficient implementation from a DSP system description remains a highly convoluted problem which is a core theme of this book.

Book Coverage

The book looks to address FPGA-based DSP systems, considering implementation at numerous levels.

- **Circuit-level** optimization techniques that allow the underlying FPGA fabric to be used more intelligently are reviewed first. By considering the detailed underlying FPGA platform, it is shown how system requirements can be mapped to provide an area-efficient, faster implementation. This is demonstrated for a number of DSP transforms and fixed coefficient filtering.
- **Architectural** solutions can be created from a signal flow graph (SFG) representation. In effect, this requires the user to exploit the highly regular, highly computative, data-independent nature of DSP systems to produce highly parallel, pipelined FPGA-based circuit architectures. This is demonstrated for filtering and beamforming applications.
- **System** solutions are now a challenge as FPGAs have now become a heterogeneous platform involving multiple hardware and software components and interconnection fabrics. There is a need for a higher-level system modeling language, e.g. dataflow which will facilitate architectural optimizations but also to address system-level considerations such as interconnection and memory.

The book covers these areas of FPGA implementation, but its key differentiating factor is that it concentrates on the second and third areas listed above, namely the creation of circuit architectures and system-level modeling; this is because circuit-level optimization techniques have been covered in greater detail elsewhere. The work is backed up with the authors' experiences in implementing practical real DSP systems and covers numerous examples including an adaptive beamformer based on a QR-based recursive least squares (RLS) filter, finite impulse response (FIR) and infinite impulse response (IIR) filters, a full search motion estimation and a fast Fourier transform (FFT) system for electronic support measures. The book also considers the development of intellectual property (IP) cores as this has become a critical aspect in the creation of DSP systems. One chapter is given over to describing the creation of such IP cores and another to the creation of an adaptive filtering core.

Audience

The book is aimed at working engineers who are interested in using FPGA technology efficiently in signal and data processing applications. The earlier chapters will be of interest to graduates and students completing their studies, taking the readers through a number of simple examples that show the trade-off when mapping DSP systems into FPGA hardware. The middle part of the book contains a number of illustrative, complex DSP system examples that have been implemented using FPGAs and whose performance clearly illustrates the benefit of their use. They provide insights into how to best use the complex FPGA technology to produce solutions optimized for speed, area and power which the authors believe is missing from current literature. The book

summarizes over 30 years of learned experience of implementing complex DSP systems undertaken in many cases with commercial partners.

Second Edition Updates

The second edition has been updated and improved in a number of ways. It has been updated to reflect technology evolutions in FPGA technology, to acknowledge developments in programming and synthesis tools, to reflect on algorithms for Big Data applications, and to include improvements to some background chapters. The text has also been updated using relevant examples where appropriate.

Technology update: As FPGAs are linked to silicon technology advances, their architecture continually changes, and this is reflected in Chapter 5. A major change is the inclusion of the ARM® processor core resulting in a shift for FPGAs to a heterogeneous computing platform. Moreover, the increased use of graphical processing units (GPUs) in DSP systems is reflected in Chapter 4.

Programming tools update: Since the first edition was published, there have been a number of innovations in tool developments, particularly in the creation of commercial C-based high-level synthesis (HLS) and open computing language (OpenCL) tools. The material in Chapter 7 has been updated to reflect these changes, and Chapter 10 has been changed to reflect the changes in model-based synthesis tools.

"Big Data" processing: DSP involves processing of data content such as audio, speech, music and video information, but there is now great interest in collating huge data sets from on-line facilities and processing them quickly. As FPGAs have started to gain some traction in this area, a new chapter, Chapter 12, has been added to reflect this development.

Organization

The FPGA is a heterogeneous platform comprising complex resources such as hard and soft processors, dedicated blocks optimized for processing DSP functions and processing elements connected by both programmable and fast, dedicated interconnections. The book focuses on the challenges of implementing DSP systems on such platforms with a concentration on the high-level mapping of DSP algorithms into suitable circuit architectures.

The material is organized into three main sections.

First Section: Basics of DSP, Arithmetic and Technologies

Chapter 2 starts with a DSP primer, covering both FIR and IIR filtering, transforms including the FFT and discrete cosine transform (DCT) and concluding with adaptive filtering algorithms, covering both the least mean squares (LMS) and RLS algorithms. Chapter 3 is dedicated to computer arithmetic and covers number systems, arithmetic functions and alternative number representations such as logarithmic number representations (LNS) and coordinate rotation digital computer (CORDIC). Chapter 4 covers the technologies available to implement DSP algorithms and includes microprocessors, DSP microprocessors, GPUs and SoC architectures, including systolic arrays. In Chapter 5, a detailed description of commercial FPGAs is given with a concentration on the two main vendors, namely Xilinx and Altera, specifically their UltraScale™/Zynq® and

Stratix® 10 FPGA families respectively, but also covering technology offerings from Lattice and MicroSemi.

Second Section: Architectural/System-Level Implementation

This section covers efficient implementation from circuit architecture onto specific FPGA families; creation of circuit architecture from SFG representations; and system-level specification and implementation methodologies from high-level representations. Chapter 6 covers only briefly the efficient implementation of FPGA designs from circuit architecture descriptions as many of these approaches have been published; the text covers distributed arithmetic and reduced coefficient multiplier approaches and shows how these have been applied to fixed coefficient filters and DSP transforms. Chapter 7 covers HLS for FPGA design including new sections to reflect Xilinx's Vivado HLS tool flow and also Altera's OpenCL approach. The process of mapping SFG representations of DSP algorithms onto circuit architectures (the starting point in Chapter 6) is then described in Chapter 8. It shows how dataflow graph (DFG) descriptions can be transformed for varying levels of parallelism and pipelining to create circuit architectures which best match the application requirements, backed up with simple FIR and IIR filtering examples.

One of the ways to perform system design is to create predefined designs termed IP cores which will typically have been optimized using the techniques outlined in Chapter 8. The creation of such IP cores is outlined in Chapter 9 and acts to address the key to design productivity by encouraging "design for reuse." Chapter 10 considers model-based design for heterogeneous FPGA and focuses on dataflow modeling as a suitable design approach for FPGA-based DSP systems. The chapter outlines how it is possible to include pipelined IP cores via the white box concept using two examples, namely a normalized lattice filter (NLF) and a fixed beamformer example.

Third Section: Applications to Big Data, Low Power

The final section of the book, consisting of Chapters 11–13, covers the application of the techniques. Chapter 11 looks at the creation of a soft, highly parameterizable core for RLS filtering, showing how a generic architecture can be created to allow a range of designs to be synthesized with varying performance. Chapter 12 illustrates how FPGAs can be applied to Big Data applications where the challenge is to accelerate some complex processing algorithms. Increasingly FPGAs are seen as a low-power solution, and FPGA power consumption is discussed in Chapter 13. The chapter starts with a discussion on power consumption, highlights the importance of dynamic and static power consumption, and then describes some techniques to reduce power consumption.

Acknowledgments

The authors have been fortunate to receive valuable help, support and suggestions from numerous colleagues, students and friends, including: Michaela Blott, Ivo Bolsens, Gordon Brebner, Bill Carter, Joe Cavallaro, Peter Cheung, John Gray, Wayne Luk, Bob Madahar, Alan Marshall, Paul McCambridge, Satnam Singh, Steve Trimberger and Richard Walke.

The authors' research has been funded from a number of sources, including the Engineering and Physical Sciences Research Council, Xilinx, Ministry of Defence, Qinetiq,

BAE Systems, Selex and Department of Employment and Learning for Northern Ireland.

Several chapters are based on joint work that was carried out with the following colleagues and students: Moslem Amiri, Burak Bardak, Kevin Colgan, Tim Courtney, Scott Fischaber, Jonathan Francey, Tim Harriss, Jean-Paul Heron, Colm Kelly, Bob Madahar, Eoin Malins, Stephen McKeown, Karen Rafferty, Darren Reilly, Lok-Kee Ting, David Trainor, Richard Turner, Fahad M Siddiqui and Richard Walke.

The authors thank Ella Mitchell and Nithya Sechin of John Wiley & Sons and Alex Jackson and Clive Lawson for their personal interest and help and motivation in preparing and assisting in the production of this work.

List of Abbreviations

1D	One-dimensional
2D	Two-dimensional
ABR	Auditory brainstem response
ACC	Accumulator
ADC	Analogue-to-digital converter
AES	Advanced encryption standard
ALM	Adaptive logic module
ALU	Arithmetic logic unit
ALUT	Adaptive lookup table
AMD	Advanced Micro Devices
ANN	Artificial neural network
AoC	Analytics-on-chip
API	Application program interface
APU	Application processing unit
ARM	Advanced RISC machine
ASIC	Application-specific integrated circuit
ASIP	Application-specific instruction processor
AVS	Adaptive voltage scaling
BC	Boundary cell
BCD	Binary coded decimal
BCLA	Block CLA with intra-group, carry ripple
BRAM	Block random access memory
CAPI	Coherent accelerator processor interface
CB	Current block
CCW	Control and communications wrapper
CE	Clock enable
CISC	Complex instruction set computer
CLA	Carry lookahead adder
CLB	Configurable logic block
CNN	Convolutional neural network
CMOS	Complementary metal oxide semiconductor
CORDIC	Coordinate rotation digital computer
CPA	Carry propagation adder
CPU	Central processing unit
CSA	Conditional sum adder

CSDF	Cyclo-static dataflow
CWT	Continuous wavelet transform
DA	Distributed arithmetic
DCT	Discrete cosine transform
DDR	Double data rate
DES	Data Encryption Standard
DFA	Dataflow accelerator
DFG	Dataflow graph
DFT	Discrete Fourier transform
DG	Dependence graph
disRAM	Distributed random access memory
DM	Data memory
DPN	Dataflow process network
DRx	Digital receiver
DSP	Digital signal processing
DST	Discrete sine transform
DTC	Decision tree classification
DVS	Dynamic voltage scaling
DWT	Discrete wavelet transform
E^2PROM	Electrically erasable programmable read-only memory
EBR	Embedded Block RAM
ECC	Error correction code
EEG	Electroencephalogram
EPROM	Electrically programmable read-only memory
E-SGR	Enhanced Squared Givens rotation algorithm
EW	Electronic warfare
FBF	Fixed beamformer
FCCM	FPGA-based custom computing machine
FE	Functional engine
FEC	Forward error correction
FFE	Free-form expression
FFT	Fast Fourier transform
FIFO	First-in, first-out
FIR	Finite impulse response
FPGA	Field programmable gate array
FPL	Field programmable logic
FPU	Floating-point unit
FSM	Finite state machine
FSME	Full search motion estimation
GFLOPS	Giga floating-point operations per second
GMAC	Giga multiply-accumulates
GMACS	Giga multiply-accumulate per second
GOPS	Giga operations per second
GPUPU	General-purpose graphical processing unit
GPU	Graphical processing unit
GRNN	General regression neural network
GSPS	Gigasamples per second

HAL	Hardware abstraction layer
HDL	Hardware description language
HKMG	High-K metal gate
HLS	High-level synthesis
I2C	Inter-Integrated circuit
I/O	Input/output
IC	Internal cell
ID	Instruction decode
IDE	Integrated design environment
IDFT	Inverse discrete Fourier transform
IEEE	Institute of Electrical and Electronic Engineers
IF	Instruction fetch
IFD	Instruction fetch and decode
IFFT	Inverse fast Fourier transform
IIR	Infinite impulse response
IM	Instruction memory
IoT	Internet of things
IP	Intellectual property
IR	Instruction register
ITRS	International Technology Roadmap for Semiconductors
JPEG	Joint Photographic Experts Group
KCM	Constant-coefficient multiplication
KM	Kernel memory
KPN	Kahn process network
LAB	Logic array blocks
LDCM	Logic delay measurement circuit
LDPC	Low-density parity-check
LLVM	Low-level virtual machine
LMS	Least mean squares
LNS	Logarithmic number representations
LPDDR	Low-power double data rate
LS	Least squares
lsb	Least significant bit
LTI	Linear time-invariant
LUT	Lookup table
MA	Memory access
MAC	Multiply-accumulate
MAD	Minimum absolute difference
MADF	Multidimensional arrayed dataflow
MD	Multiplicand
ME	Motion estimation
MIL-STD	Military standard
MIMD	Multiple instruction, multiple data
MISD	Multiple instruction, single data
MLAB	Memory LAB
MMU	Memory management unit
MoC	Model of computation

MPE	Media processing engine
MPEG	Motion Picture Experts Group
MPSoC	Multi-processing SoC
MR	Multiplier
MR-DFG	Multi-rate dataflow graph
msb	Most significant bit
msd	Most significant digit
MSDF	Multidimensional synchronous dataflow
MSI	Medium-scale integration
MSPS	Megasamples per second
NaN	Not a Number
NLF	Normalized lattice filter
NRE	Non-recurring engineering
OCM	On-chip memory
OFDM	Orthogonal frequency division multiplexing
OFDMA	Orthogonal frequency division multiple access
OLAP	On-line analytical processing
OpenCL	Open computing language
OpenMP	Open multi-processing
ORCC	Open RVC-CAL Compiler
PAL	Programmable Array Logic
PB	Parameter bank
PC	Program counter
PCB	Printed circuit board
PCI	Peripheral component interconnect
PD	Pattern detect
PE	Processing element
PL	Programmable logic
PLB	Programmable logic block
PLD	Programmable logic device
PLL	Phase locked loop
PPT	Programmable power technology
PS	Processing system
QAM	Quadrature amplitude modulation
QR-RLS	QR recursive least squares
RAM	Random access memory
RAN	Radio access network
RCLA	Block CLA with inter-block ripple
RCM	Reduced coefficient multiplier
RF	Register file
RISC	Reduced instruction set computer
RLS	Recursive least squares
RNS	Residue number representations
ROM	Read-only memory
RT	Radiation tolerant
RTL	Register transfer level
RVC	Reconfigurable video coding

SBNR	Signed binary number representation
SCU	Snoop control unit
SD	Signed digits
SDF	Synchronous dataflow
SDK	Software development kit
SDNR	Signed digit number representation
SDP	Simple dual-port
SERDES	Serializer/deserializer
SEU	Single event upset
SFG	Signal flow graph
SGR	Squared Givens rotation
SIMD	Single instruction, multiple data
SISD	Single instruction, single data
SMP	Shared-memory multi-processors
SNR	Signal-to-noise ratio
SoC	System-on-chip
SOCMINT	Social media intelligence
SoPC	System on programmable chip
SPI	Serial peripheral interface
SQL	Structured query language
SR-DFG	Single-rate dataflow graph
SRAM	Static random access memory
SRL	Shift register lookup table
SSD	Shifted signed digits
SVM	Support vector machine
SW	Search window
TCP	Transmission Control Protocol
TFLOPS	Tera floating-point operations per second
TOA	Time of arrival
TR	Throughout rate
TTL	Transistor-transistor logic
UART	Universal asynchronous receiver/transmitter
ULD	Ultra-low density
UML	Unified modeling language
VHDL	VHSIC hardware description language
VHSIC	Very high-speed integrated circuit
VLIW	Very long instruction word
VLSI	Very large scale integration
WBC	White box component
WDF	Wave digital filter