

THE ZYNQ[®] BOOK

Louise H. Crockett
Ross A. Elliot
Martin A. Enderwitz
Robert W. Stewart



*Embedded Processing with the ARM[®] Cortex[®]-A9
on the Xilinx[®] Zynq[®]-7000 All Programmable SoC*

In association with



The Zynq Book

Embedded Processing with the
ARM[®] Cortex[®]-A9 on the Xilinx[®]
Zynq[®]-7000 All Programmable SoC

Louise H. Crockett

Ross A. Elliot

Martin A. Enderwitz

Robert W. Stewart

Department of Electronic and Electrical Engineering

University of Strathclyde

Glasgow, Scotland, UK

1st Edition

This edition first published July 2014 by Strathclyde Academic Media.

© Louise H. Crockett, Ross A. Elliot, Martin A. Enderwitz and Robert W. Stewart.

Open Source Licence to Use and Reproduce

This book is available in print and as an electronic book (PDF format).

Text and diagrams from this book may be reproduced in their entirety and used for non-profit academic purposes, provided that a clear reference to the original source is made in all derivative documents. This reference should be of the following form:

L. H. Crockett, R. A. Elliot, M. A. Enderwitz and R. W. Stewart, *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*, First Edition, Strathclyde Academic Media, 2014.

Requests to use content from this book for other than non-profit academic purposes should be made to info@zynqbook.com.

This book may not be reproduced in its original form and sold by any unauthorised third party.

Tutorial Materials

Tutorial materials are distributed via the book's companion website: www.zynqbook.com.

The tutorials are provided with the same *Open Source License to Use and Reproduce*, and the same *Warning and Disclaimer*, as detailed elsewhere on this page in reference to the main book.

Warning and Disclaimer

The best efforts of the authors and publisher have been used to ensure that accurate and current information is presented in this book. This includes researching the topics covered and developing examples. The material included is provided on an "as is" basis in the best of faith, and neither the authors and publishers make any warranty of any kind, expressed or implied, with regard to the documentation contained in this book. The authors and publisher shall not be held liable for any loss or damage resulting directly or indirectly from any information contained herein.

Trademarks

ARM, Cortex, AMBA, Thumb and TrustZone are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

NEON is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

This publication is independent and it is not affiliated with, or endorsed, sponsored or authorised by ARM Limited.

Xilinx, the Xilinx logo, Artix, ISE, Kintex, LogiCORE, Petalogix, Spartan, Virtex, Vivado, Zynq, and WebPACK are registered trademarks of Xilinx. All rights reserved.

MATLAB and Simulink are registered trademarks of MathWorks, Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

All other trademarks used in this book are acknowledged as belonging to their respective companies. The use of trademarks in this book does not imply any affiliation with, or endorsement of, this book by trademark owners.

The Zynq Book

Embedded Processing with the
ARM® Cortex®-A9 on the Xilinx®
Zynq®-7000 All Programmable SoC

Foreword

For over two years now, academics, industry professionals and "makers" worldwide have had access to development boards that use the Zynq®-7000 All Programmable SoC from Xilinx®. These boards including the ZedBoard, the Zc702, Zc706 and others have provided users heretofore unprecedented abilities to build their own customizable System on Chip (SoC) solutions. The Zynq SoC integrates an ARM® dual Cortex®-A9 based processor system with Xilinx 7-series FPGA fabric and by doing so it provides the power, performance and capacity benefits of an ASIC combined with the hardware programmability benefits of an FPGA.

The strong demand for these devices and the first of its kind nature goes hand-in-hand with a demand for supporting collateral in the form of documentation, training, tutorials and guidebooks. As the first book written in the English language, "The Zynq Book" does a remarkable job of filling a critical need, and the team from the University of Strathclyde have put together a very comprehensive book, which covers the essential information every Zynq user needs to know.

The book begins appropriately with an overview of the Zynq device, following up with a description of the ZedBoard. The book then moves very quickly into information needed to build designs targeted for the Zynq family, describing in depth both the development flow for these devices as well as the implications of various design choices. As this is a hybrid device which is both software and hardware programmable, the content comprehensively spans hardware design tools as well as the higher level software tools and flow. A special spotlight on Vivado® High Level Synthesis (HLS) is included, which showcases the productivity benefits offered by HLS as well as the synergy with the high level programming model offered by the Cortex-A9 processors. Of critical importance are the

interfaces that connect the Processor System to the Programmable Logic or FPGA. The book does an excellent job of providing an overview of these interfaces and guidance on appropriately configuring them. Finally no book on an SoC is complete without a description of the embedded software run-time environment. The concluding chapter guides the reader through the nuances of booting Linux® on their very own custom SoC.

It is no coincidence that the first English language book on Zynq originates from the University of Strathclyde. Since 2005 Xilinx has worked closely with the Department of Electronic and Electrical Engineering at the University of Strathclyde in Scotland, UK. As the Xilinx Endowed Chair, Professor Bob Stewart played a leading role in developing and disseminating best practices for using Xilinx *All Programmable* technologies in academia and industry. He and his team have created outstanding pedagogical material that has enhanced the educational experience of tens of thousands of students and engineers around the world. The successful relationship undoubtedly reflects the University of Strathclyde's pragmatic, industry-driven ethos and its position as a leading technological institution with an excellent international reputation. Indeed the University of Strathclyde was founded in 1796 by Professor John Anderson, whose bequest was to create "a place of useful learning". It is very clear that the University continues to be a place of *useful learning* and a strong partner with Xilinx for technology education and research. It is also worth saying that we are particularly pleased to be associated with an institution recognized in the UK as University of the Year for 2013, and the UK Entrepreneurial University of the Year for 2014.

This book is indeed a must read item for the first time Zynq user!

Vidya Rajagopalan

Corporate Vice President of Processing, Systems, Software and Applications (PSSA),
Xilinx, Inc.

June 2014.

Acknowledgements

There are a number of people to whom we must extend our sincerest thanks for their support and practical help in producing The Zynq Book.

First of all, it simply would not have come to life without the vision of Patrick Lysaght, Senior Director at Xilinx, whose idea it was to create a book about Zynq. The project soon took on a life of its own, and we realised that there are many interesting things to write about Zynq! The resulting book is the product not just of our own efforts, but the many people who have helped us along the way.

Firstly, nobody could have been more helpful than Cathal McCabe, who has managed the project from Xilinx University Program. His knowledgeable support has been absolutely vital, and we are indebted to him for his patience, thoughtful input, and thoroughness. This book would not have turned out nearly so well without Cathal.

We also greatly appreciate the input of several colleagues from Xilinx: Sagheer Ahmad, Brian Gaide, Austin Lesea, Joshua Lu, Duncan Mackay, Daryl Nees, Stephen Neuendorffer, Parimal Patel, Fernando Martinez Vallina, Tim Vanevenhoven, and Y.C. Wang all gave of their time to read the book and provide constructive criticism. Some of these helpful people also took the trouble to work through the tutorials and share their feedback. It has been a great help to access their experience — thanks guys! We must also thank Barrie Mullins for his help and support.

Our colleagues at University of Strathclyde have been a great help as well, and we would like to thank them for their various inputs to the project. Several of them generously took the time to read and review chapters, or to work through and test tutorials, and provided much thoughtful feedback: many thanks to Douglas Allan, Dani Anderson, Dale Atkinson,

Kenneth Barlee, Iain Chalmers, Fraser Coutts, David Crawford, Sam Edwards, Poppy Harvey, Connor Hughes, Sarunas Kalade, Phil Karagiannakis, David Northcote and Kenneth Osborne. We also appreciate the contributions of all to the general “Zynq-chat” which has now become a feature of our working environment!

Last but not least, we must of course extend our thanks to our family and friends, who accepted with good grace our lack of participation in social activities, particularly as the various deadlines approached (!), and were most generous with their encouragement.

Louise Crockett, Ross Elliot, Martin Enderwitz, Bob Stewart.
June 2014.

Contents

Foreword	xxi
Acknowledgements	xxiii
CHAPTER 1 Introduction	1
1.1 System-on-Chip with Zynq	2
1.2 Simple Anatomy of an Embedded SoC	5
1.3 Design Reuse	7
1.4 Raising the Abstraction Level	7
1.5 SoC Design Flow	8
1.6 Practical Elements	10
1.7 About This Book	10
1.8 References	12
PART A Getting to Know Zynq	13
CHAPTER 2 The Zynq Device (“What is it?”)	15
2.1 Processing System	16
2.1.1 Application Processing Unit (APU)	17
2.1.2 A Note on the ARM Model	20
2.1.3 Processing System External Interfaces	21
2.2 Programmable Logic	22

2.2.1	The Logic Fabric	23
2.2.2	Special Resources: DSP48E1s and Block RAMs	25
2.2.3	General Purpose Input/Output	28
2.2.4	Communications Interfaces	29
2.2.5	Other Programmable Logic External Interfaces	29
2.3	Processing System — Programmable Logic Interfaces	30
2.3.1	The AXI Standard	30
2.3.2	AXI Interconnects and Interfaces	31
2.3.3	EMIO Interfaces	34
2.3.4	Other PL-PS Signals	34
2.4	Security	35
2.4.1	Secure Boot	35
2.4.2	Hardware Support	36
2.4.3	Runtime Security	36
2.5	Zynq-7000 Family Members	39
2.6	Chapter Review	40
2.7	Architecture Reference Guide	41
2.8	References	44

CHAPTER 3 Designing with Zynq (“How do I work with it?”) 47

3.1	Getting Started	48
3.1.1	Obtaining Design Tools	48
3.1.2	Design Tool Editions and Licensing	49
3.1.3	Design Tool Functionality	50
3.1.4	Third Party Tools	51
3.1.5	System Setup and Requirements	51
3.2	An Outline of the Design Flow	53
3.2.1	Requirements and Specification	54
3.2.2	System Design	54
3.2.3	Hardware Development and Testing	55
3.2.4	Software Development and Testing	58
3.2.5	System Integration and Testing	60
3.3	SoC Design Teams	60
3.4	System-Level IP-Focused Design with Vivado	62
3.5	The ISE and Vivado Design Suites	64
3.5.1	Features Comparison	64

3.5.2	Upgrading to Vivado	66
3.6	Development Boards	67
3.6.1	Zynq-7000 SoC ZC702 Evaluation Kit	67
3.6.2	Zynq-7000 SoC Video & Imaging Kit	69
3.6.3	Zynq-7000 ZC706 Evaluation Kit	69
3.6.4	ZedBoard	69
3.6.5	ZYBO	69
3.6.6	Third Party Boards	70
3.6.7	Accessories and Expansions	71
3.6.8	Working with Development Boards	72
3.7	Support and Documentation	72
3.8	Chapter Review	72
3.9	References	73
CHAPTER 4	Device Comparisons (“Why do I need Zynq?”)	77
4.1	Device Selection Criteria	78
4.2	Comparison A: Zynq versus FPGA	80
4.2.1	MicroBlaze Processor	80
4.2.2	MicroBlaze MicroController System	84
4.2.3	PicoBlaze	85
4.2.4	ARM Cortex-M1	85
4.2.5	Other Processor Types	85
4.2.6	Summary Comments	87
4.3	Comparison B: Zynq versus Standard Processor	89
4.3.1	Processor Operation	89
4.3.2	Execution Profiling	92
4.3.3	Summary Comments	94
4.4	Comparison C: Zynq versus a Discrete FPGA-Processor Combination	94
4.5	Exploiting the Zynq Architecture and Design Flow	96
4.6	Chapter Review	98
4.7	References	99
CHAPTER 5	Applications and Opportunities (“What can I do with it?”)	101
5.1	An Overview of Applications	102

5.1.1	Automotive	102
5.1.2	Communications	102
5.1.3	Defence and Aerospace	103
5.1.4	Robotics, Control and Instrumentation	103
5.1.5	Image and Video Processing	104
5.1.6	Medical	105
5.1.7	High Performance Computing (HPC)	105
5.1.8	Others and Future Applications	105
5.2	When Can Zynq Really Help...?	106
5.3	Communications: Software Defined Radio (SDR)	107
5.3.1	Trends in Wireless Communications	107
5.3.2	Introducing Software Defined Radio (SDR)	108
5.3.3	SDR Implementation and Enabling Technologies	108
5.3.4	Cognitive Radio	110
5.4	Smart Systems and Smart Networks	111
5.4.1	What is a Smart System?	111
5.4.2	Examples of Smart Systems	112
5.4.3	Smart Networks: Communications for Smart Systems	114
5.4.4	Related Concepts	115
5.5	Image and Video Processing, and Computer Vision	115
5.5.1	Image and Video Processing	115
5.5.2	Computer Vision	116
5.5.3	Levels of Abstraction	117
5.5.4	Implementation of Image Processing Systems	118
5.5.5	Computer Vision on Zynq Example: Road Sign Recognition	120
5.6	Dynamic System-on-Chip	121
5.6.1	Run Time System Flexibility	121
5.6.2	Dynamic Partial Reconfiguration (DPR)	121
5.6.3	DPR Application Examples	122
5.6.4	Benefits of DPR	124
5.7	Further Opportunities: the Zynq 'EcoSystem'	125
5.7.1	What is the Ecosystem?	125
5.7.2	What is the Opportunity?	126
5.8	Chapter Review	128
5.9	References	128

CHAPTER 6	The ZedBoard	133
6.1	Introducing Zed	133
6.2	ZedBoard System Architecture	134
6.3	The Design Flow for ZedBoard	136
6.4	Getting Started with the ZedBoard	137
6.4.1	What's in the Box?	137
6.4.2	Hardware Setup	137
6.4.3	Programming the ZedBoard	138
6.5	MicroZed	142
6.6	Documentation, Tutorials and Support	142
6.6.1	Documentation about the ZedBoard	142
6.6.2	Demonstrations and Tutorials	143
6.6.3	Online Courseware	143
6.6.4	Other ZedBoard Resources and Support	144
6.7	ZedBoard.org Community	144
6.7.1	Community Projects	144
6.7.2	Blogs	145
6.7.3	Support Forums	145
6.8	Chapter Review	145
6.9	References	146
 CHAPTER 7	 Education, Research and Training	 147
7.1	Technology Trends and SoC Education	148
7.2	University Teaching with Zynq	149
7.2.1	Teaching with Xilinx Tools and Boards	149
7.2.2	Digital Design and FPGA Teaching	150
7.2.3	Computer Science	150
7.2.4	Embedded Systems and SoC Design	150
7.2.5	Algorithm Implementation (e.g. Signal, Image, and Video Processing)	151
7.2.6	Design Reuse	152
7.2.7	New and Emerging Design Methods	153
7.2.8	Sensing, Robotics, and Prototyping	154
7.2.9	An Example Course	154
7.3	Projects and Competitions	155
7.4	Academic Research	156

7.5	The Xilinx University Program (XUP)	158
7.5.1	Introducing XUP	158
7.5.2	Software Support and Licenses	158
7.5.3	XUP Development and Teaching Boards	159
7.5.4	XUP Workshops and Training Materials	159
7.5.5	Technical Support for Universities	160
7.5.6	Eligibility	160
7.5.7	Getting in Touch with XUP	160
7.6	Training for Industry	160
7.6.1	Courses and Authorised Training Providers	160
7.6.2	Other Resources	161
7.6.3	Online Videos	161
7.7	Chapter Review	161
7.8	References	162

CHAPTER 8 First Designs on Zynq 165

8.1	Software Installation Guide	166
8.2	Aims and Outcomes	166
8.3	Overview of Exercise 1A	166
8.4	Overview of Exercise 1B	167
8.5	Overview of Exercise 1C	168
8.6	Possible Extensions	169
8.7	What Next?	169
8.8	References	169

PART B Zynq SoC & Hardware Design 171

CHAPTER 9 Embedded Systems and FPGAs 173

9.1	What is an Embedded System?	173
9.1.1	Applications	174
9.1.2	Generic Embedded System Architecture	175
9.2	Processors	176
9.2.1	Co-processors	177
9.2.2	Processor Cache	178
9.2.3	Execution Cycles	180
9.2.4	Interrupts	183

9.3	Buses	184
9.3.1	System and Peripheral Buses	185
9.3.2	Bus Masters and Slaves	186
9.3.3	Bus Arbitration	186
9.3.4	Memory Access	187
9.3.5	Bus Bandwidth	188
9.4	Chapter Review	189
9.5	References	189
CHAPTER 10	Zynq System-on-Chip Design Overview	191
10.1	Interfacing and Signals	192
10.1.1	PS-PL AXI Interfaces	192
10.1.2	PL Co-Processing Interfaces	193
10.1.3	Interrupt Interface	196
10.2	Interconnects	197
10.2.1	Interconnect Features	197
10.2.2	Interconnects, Masters and Slaves	198
10.2.3	Connectivity	199
10.2.4	AXI_HP Interfaces	200
10.2.5	AXI_ACP Interface	202
10.2.6	AXI_GP Interfaces	202
10.3	Memory	202
10.3.1	Memory Interfaces	203
10.3.2	On-Chip Memory (OCM)	208
10.3.3	Memory Map	210
10.4	Interrupts	211
10.4.1	Interrupt Signals	212
10.4.2	Generic Interrupt Controller (GIC)	212
10.4.3	Interrupt Sources	213
10.4.4	Interrupt Prioritisation and Handling	217
10.4.5	Further Reading	218
10.5	Chapter Review	219
10.6	References	219
CHAPTER 11	Zynq System-on-Chip Development	221
11.1	Hardware/Software Partitioning	221

11.2	Profiling	224
11.3	Software Development Tools	226
11.3.1	Software Tools	226
11.3.2	Hardware Configuration Tools	227
11.3.3	Software Development Kit (SDK)	228
11.3.4	Microprocessor Debugger	228
11.3.5	Sourcery CodeBench Lite Edition for Xilinx Cortex-A9 Compiler Toolchain	229
11.3.6	Logic Analysers	229
11.3.7	System Generator for DSP	229
11.4	Chapter Review	230
11.5	References	230
CHAPTER 12	Next Steps in Zynq SoC Design	231
12.1	Prerequisites	231
12.2	Aims and Outcomes	232
12.3	Overview of Exercise 2A	232
12.4	Overview of Exercise 2B	232
12.5	Overview of Exercise 2C	233
12.6	Overview of Exercise 2D	234
12.7	Possible Extensions	235
12.8	What Next?	236
CHAPTER 13	IP Block Design	237
13.1	Overview	237
13.2	Industry Trends and Philosophy	239
13.3	IP Core Design Methods	240
13.3.1	HDL	240
13.3.2	System Generator	241
13.3.3	HDL Coder	241
13.3.4	Vivado High-Level Synthesis	243
13.3.5	Choosing the Right IP Creation Method	244
13.4	Simulation and Documentation	244
13.4.1	Simulation	244
13.4.2	Documentation	249
13.5	Chapter Review	252

13.6	References	252
CHAPTER 14	Spotlight on High-Level Synthesis	255
14.1	High-Level Synthesis Concepts	256
14.1.1	What is High-Level Synthesis (HLS)?	256
14.1.2	Motivations for High-Level Synthesis	257
14.1.3	Design Metrics and Hardware Architectures	259
14.2	Development of HLS Tools	260
14.3	HLS Source Languages	262
14.3.1	C	262
14.3.2	C++	263
14.3.3	SystemC	263
14.3.4	Other Languages for High-Level Synthesis	264
14.4	Introducing Vivado HLS	264
14.4.1	What Does Vivado HLS Do?	264
14.4.2	Vivado HLS Design Flow	267
14.4.3	C Functional Verification and C/RTL Cosimulation ..	269
14.4.4	Implementation Metrics and Considerations	271
14.4.5	Overview of the High-Level Synthesis Process	272
14.4.6	Solutions: Exploring the Design Space	276
14.4.7	Vivado HLS Library Support	277
14.5	HLS in the Design Flow for Zynq	277
14.6	Chapter Review	278
14.7	References	278
CHAPTER 15	Vivado HLS: A Closer Look	281
15.1	Anatomy of a Vivado HLS Project	282
15.2	Vivado HLS User Interfaces	283
15.2.1	Graphical User Interface	284
15.2.2	Command Line Interface (CLI)	286
15.3	Data Types	287
15.3.1	C and C++ Native Data Types	287
15.3.2	Vivado HLS Arbitrary Precision Data Types for C and C++	289
15.3.3	Arbitrary Precision Types for SystemC	292
15.3.4	Floating Point Data Types and Operators	294