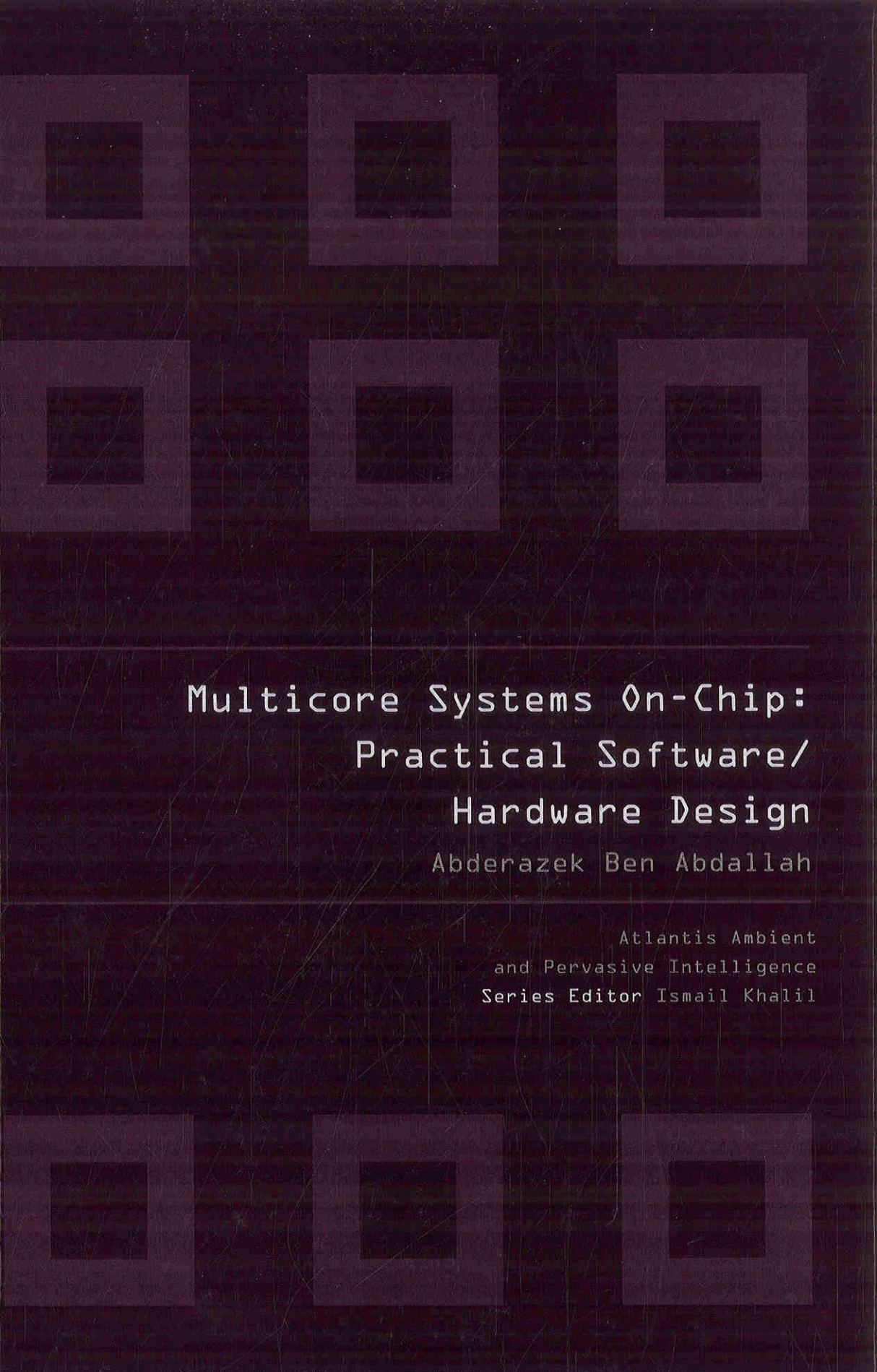


The background of the cover features a repeating pattern of squares. Each square consists of a dark outer frame and a lighter, textured inner square. These squares are arranged in a grid, with some squares appearing slightly offset or faded to create a sense of depth and movement. The overall color palette is dark, with shades of brown, black, and grey.

Multicore Systems On-Chip: Practical Software/ Hardware Design

Abderazek Ben Abdallah

Atlantis Ambient
and Pervasive Intelligence
Series Editor Ismail Khalil

Contents

Preface	vii
List of Figures	xiii
List of Tables	xvii
1. Multicore Systems Design Methodology	1
1.1 Introduction	1
1.2 MCSocCs Design Problems	2
1.3 Multicore architecture platform	4
1.4 Application specific MCSocC design method	5
1.5 QueueCore architecture	6
1.5.1 Hardware pipeline structure	7
1.5.2 Floating point organization	8
1.6 QueueCore synthesis and evaluation results	12
1.6.1 Methodology	12
1.6.2 Design results	13
1.7 Conclusion	14
2. Design for Low Power Systems	17
2.1 Introduction	17
2.2 Power Aware Technological-level Design optimizations	19
2.2.1 Factors affecting CMOS power consumption	19
2.2.2 Reducing voltage and frequency	20
2.2.3 Reducing capacitance	21
2.3 Power Aware Logic-level Design Optimizations	22
2.3.1 Clock gating	22
2.3.2 Logic encoding	23
2.3.3 Data guarding	23
2.4 Power-Aware System Level Design Optimizations	24
2.4.1 Hardware system architecture power consumption optimizations	24

2.4.2	Operating system power consumption optimization	28
2.4.3	Application, compilation techniques and algorithm	29
2.4.4	Energy reduction in network protocols	30
2.5	Conclusion	34
3.	Network-on-Chip for Multi- and Many-Core Systems	35
3.1	Introduction	35
3.2	Router Architecture	37
3.2.1	Switching Technique	38
3.2.2	Router Components	39
3.2.3	Pipeline Processing	40
3.3	Topology	48
3.3.1	Interconnection Topologies	48
3.3.2	Topological Properties	52
3.4	Routing	55
3.4.1	Deadlocks and Livelocks	55
3.4.2	Routing Algorithm	56
3.5	Summary	59
4.	Parallelizing Compiler for High Performance Computing	61
4.1	Instruction Level Parallelism	61
4.2	Parallel Queue Compiler	64
4.3	Parallel Queue Compiler Frame Work	65
4.3.1	1-offset P-Code Generation Phase	66
4.3.2	Offset Calculation Phase	68
4.3.3	Instruction Scheduling Phase	70
4.3.4	Natural Instruction Level Parallelism Extraction: Statement Merging Transformation	71
4.3.5	Assembly Generation Phase	73
4.3.6	Results	74
4.4	Conclusion	78
5.	Dual-Execution Processor Architecture for Embedded Computing	79
5.1	Introduction	79
5.2	System Architecture	81
5.2.1	Pipeline Structure	82
5.2.2	Fetch unit	83
5.2.3	Decode unit	84
5.2.4	Dynamic switching mechanism	84
5.2.5	Calculation of produced and consumed data	85
5.2.6	Queue-Stack Computation unit	85
5.2.7	Sources-Results computing mechanism	86

5.2.8	Issue unit	89
5.2.9	Execution unit	91
5.2.10	Shared storage mechanism	92
5.2.11	Covop instruction execution mechanism	93
5.2.12	Interrupt handling mechanism	94
5.3	Sub-routine call handling mechanism	97
5.4	Hardware design and Evaluation results	100
5.4.1	DEP System pipeline control	101
5.4.2	Hardware Design Result	102
5.4.3	Comparison results	105
5.5	Conclusions	106
6.	Low Power Embedded Core Architecture	107
6.1	Introduction	107
6.2	Produced Order Queue Computing Overview	108
6.3	QC-2 Core Architecture	111
6.3.1	Instruction Set Design Considerations	111
6.3.2	Instruction Pipeline Structure	112
6.3.3	Dynamic Operands Addresses Calculation	114
6.3.4	QC-2 FPA Organization	115
6.3.5	Circular Queue-Register Structure	118
6.4	Synthesis of the QC-2 Core	119
6.4.1	Design Approach	119
6.5	Results and Discussions	121
6.5.1	Execution Speedup and Code Analysis	121
6.5.2	Synthesis Results	122
6.5.3	Speed and Power Consumption Comparison with Synthesizable CPU Cores	124
6.6	Conclusions	125
7.	Reconfigurable Multicore Architectures	127
7.1	Introduction	127
7.1.1	Performance-Driven Approaches in Software	128
7.1.2	Performance-Driven Approaches in Hardware	129
7.1.3	Potential of FPGA SOC's	130
7.2	Runtime Hardware Adaptation	130
7.2.1	Processor Array Architectures	131
7.2.2	Datapath Array Architectures	135
7.2.3	Single Processor Architectures	138
7.3	Summary of Hardware Adaptation	140
7.4	Runtime Software Adaptation	142
7.4.1	Warp Processing	143

7.4.2	Dynamic Instruction Merging	148
7.4.3	Summary of Software Adaptation	152
7.5	Future Directions in Runtime Adaptation	153
7.5.1	Future Hardware Adaptation	153
7.5.2	Future Software Adaptation	155
7.5.3	Future Reconfiguration Infrastructures in FPGA SOCs	156
7.6	Conclusion	157
Bibliography		159
Index		177