

Contents

Acknowledgments	v
Abstract	vii
Zusammenfassung	ix
1 Introduction	1
1.1 Motivation	1
1.1.1 Energy-Efficiency	3
1.2 Open-Source Hardware	4
1.2.1 Risks of Open-Source Hardware	4
1.2.2 Benefits of Open-Source Hardware	5
1.3 RISC-V ISA	6
1.4 Thesis Overview	7
1.5 Contributions	8
1.6 List of Publications	9
2 Ariane: An Application-Class RISC-V Processor	13
2.1 Introduction	13
2.1.1 Contributions	15
2.2 Related Work	15
2.2.1 State of the Art since Publication	16
2.3 Architecture	17
2.3.1 Branch Prediction	21
2.3.2 Virtual Memory	22
2.3.3 Exception Handling	23

2.3.4	Privileged Extensions	23
2.3.5	Register Files	24
2.3.6	Scoreboard/Reorder Buffer	24
2.3.7	Functional Units	25
2.3.8	Physical Memory Attributes	26
2.3.9	Physical Memory Protection	26
2.3.10	Caches	27
2.3.11	Memory and Control Interfaces	27
2.3.12	Single-Core FPGA SoC	27
2.3.13	Debug Interface	29
2.3.14	Tracing and Performance Counters	29
2.3.15	Concluding Remarks	29
2.4	Implementation	30
2.4.1	Physical Design	33
2.5	Results	34
2.5.1	Methodology	34
2.5.2	Discussion	41
2.5.3	Comparison with Non-application-Class Cores	43
2.5.4	Comparison to Other Application-Class Cores	45
2.6	Conclusion	48
3	Homogenous Cache-Coherent Multi-core System	51
3.1	Introduction	51
3.2	Related Work	52
3.2.1	State of the Art since Publication	53
3.3	Architecture	53
3.3.1	Cache Subsystem	54
3.3.2	Support for RISC-V Atomic Operations	56
3.3.3	Platform Peripherals	57
3.3.4	Automatic Device Tree Generation	62
3.4	Simulation and Emulation Platforms	62
3.4.1	FPGA Flows	63
3.4.2	ASIC Flows	64
3.5	Results	64
3.6	Conclusion	66

4	Heterogenous Energy-Efficient Multi-Core System	67
4.1	Introduction	67
4.2	Contributions	68
4.3	General-Purpose Cores	70
4.3.1	Ariane High-Performance	70
4.3.2	Ariane Low-Power	70
4.4	Specialized Accelerator	71
4.5	Results	72
4.5.1	Core vs. Accelerator	74
4.5.2	Cell Library (Technology)	74
4.5.3	Body Bias Voltage	74
4.5.4	Supply Voltage	77
4.5.5	The FP Precision and Energy Trade-off	77
4.5.6	Summary	78
4.5.7	Comparison to State of the Art	79
4.6	Conclusion	80
5	Snitch: Energy-Efficient Multi-Core Cluster for Data-Oblivious Floating-Point Workloads	81
5.1	Introduction	81
5.2	Motivation	82
5.2.1	Design Goal: Area and Energy-Efficiency	83
5.2.2	Existing Mitigation Techniques and Architectures	84
5.2.3	Contributions	85
5.3	Architecture	87
5.3.1	Snitch Core Complex	89
5.3.2	Virtual Memory	94
5.3.3	Snitch Hive	96
5.3.4	Snitch Cluster	97
5.3.5	Stream Semantic Register (SSR)	100
5.3.6	FPU Sequence Buffer	100
5.4	Programming	103
5.4.1	Stream Semantic Registers	104
5.4.2	Floating-point unit (FPU) Sequencer	108
5.5	Results	109
5.5.1	Microkernels	111
5.5.2	Single-Core	113
5.5.3	Multi-Core	116

5.6	Related Work	124
5.6.1	Vector Architectures	124
5.6.2	Graphics Processing Units	129
5.6.3	Super-Scalar CPUs	130
5.7	Conclusion	130
6	Manticore: A Chiplet-Based Many-Core Architecture for Energy-Efficient Floating-Point Execution	135
6.1	Introduction	135
6.2	Related Work	136
6.3	Chiplet Architecture	136
6.3.1	Memory Hierarchy	139
6.3.2	Compute Cluster	141
6.4	Programming	142
6.4.1	Stream Semantic Registers (Xssr)	143
6.4.2	Floating-Point Repetition (Xfrep)	144
6.4.3	Typical SSR/FREP Execution	145
6.5	Runtime	145
6.5.1	Bootloader	147
6.5.2	Firmware	148
6.6	Prototype	150
6.6.1	Power Management	151
6.7	Silicon Performance	154
6.7.1	Efficiency	154
6.7.2	Roofline	154
6.7.3	Applications	157
6.8	Conclusion	159
7	Summary and Conclusion	161
7.1	Summary of Main Results	162
7.2	Outlook and Future Work	163
7.2.1	Future Work	164
7.2.2	Outlook	165
A	Chip Gallery	167
B	Acronyms	179

<i>CONTENTS</i>	xv
-----------------	----

Bibliography	191
---------------------	------------

Curriculum Vitae	209
-------------------------	------------