

Contents

1	Introduction	1
1.1	Research Contributions	5
1.2	Thesis Outline	6
1.3	Statement of Originality	6
	References	8
2	High-Level Synthesis of Dynamic Data Structures	11
2.1	Background	13
2.2	Analysis of the Filtering Algorithm	17
2.3	RTL Implementations	19
2.3.1	Lloyd's Algorithm	19
2.3.2	Filtering Algorithm	20
2.4	HLS Implementations	22
2.4.1	Lloyd's Algorithm	23
2.4.2	Filtering Algorithm	23
2.5	Performance Comparison	26
2.5.1	RTL Designs	26
2.5.2	HLS Designs	28
2.6	Summary	30
	References	32
3	Background	35
3.1	Profiling and User Annotation-Based Approaches	36
3.2	Automated Static Analyses for Static Control Parts	37
3.3	Limitations and Extensions of the Polyhedral Framework	40
3.4	HLS Support for Pointers and Dynamic Memory Allocation	41
3.5	Static Analysis Based on Separation Logic	43
3.5.1	Modelling Program State in Separation Logic	44
3.5.2	Programming Language	46
3.5.3	Symbolic Execution of Programs	47

3.5.4	Theorem Proving	49
3.5.5	Application to HLS	52
References	52
4	Heap Partitioning and Parallelisation	57
4.1	Motivating Example	59
4.2	Program Analysis	62
4.2.1	Inserting Cut-Points	63
4.2.2	Proving Communication-Free Parallelism	65
4.2.3	Assigning Heap Partition Information to Statements	67
4.3	Implementation	69
4.3.1	Heap Analyser	69
4.3.2	Memory Access	70
4.3.3	Dynamic Memory Allocation	71
4.3.4	Control and Data Flow	71
4.3.5	Source-to-Source Compiler	72
4.4	Experiments	74
4.5	Performance and Robustness of the Heap Analysis	80
4.6	Summary	83
References	84
5	Custom Multi-cache Architectures	85
5.1	Motivating Example	87
5.1.1	Memory Partitioning and Parallelisation	89
5.1.2	Parallel Access to Shared Resources	90
5.1.3	Custom Cache Sizing	91
5.2	Extended Static Program Analysis	92
5.2.1	Detecting Private and Shared Resources	94
5.2.2	Commutativity Analysis	95
5.3	Code Generation	97
5.4	Custom Cache Sizing	99
5.4.1	On-Chip Memory Utilisation Estimation	101
5.4.2	Cache Performance Estimation	102
5.4.3	Optimisation Strategy	104
5.5	Experiments	104
5.5.1	Hybrid Multi-cache Architectures	105
5.5.2	Validating the BRAM Estimation for Automated Cache Scaling	108
5.5.3	Validating Cache Performance Estimation	108
5.5.4	Latency and Resource Utilisation After Custom Cache Scaling	109
5.5.5	Energy Consumption	112
5.6	Summary	113
References	114

6 Conclusion 117

6.1 Outlook 120

6.2 Final Remarks 122

References 122

Appendix A: Context-Aware Heap Analysis. 125

<http://www.springer.com/978-3-319-53221-9>

Separation Logic for High-level Synthesis

Winterstein, F.

2017, XIX, 132 p. 19 illus., 7 illus. in color., Hardcover

ISBN: 978-3-319-53221-9