

Contents

Summary	9
Streszczenie	10
List of symbols and abbreviations	11
Introduction	15
1. Computing and FPGAs	19
1.1. Basics of FPGA devices	19
1.2. The challenges of efficient data processing	22
1.3. Massively parallel computing	24
1.4. Heterogeneous computing platforms	25
1.4.1. Graphics cards.....	27
1.4.2. FPGA accelerators	28
1.5. Data-intensive computing	30
1.5.1. The big O notation	31
1.5.2. Computational patterns	32
1.5.3. Distributed databases	33
1.6. FPGAs in data-intensive applications	34
1.7. Architectures for energy-efficient computing	38
1.8. Energy efficiency of FPGAs	41
1.9. Types of FPGA-enabled architectures	45
2. Custom processor design in FPGAs	51
2.1. The general architecture of a custom processor.....	51
2.1.1. Algorithm selection.....	51
2.1.2. An example of the SQL custom processor.....	52
2.1.3. The Finite-State Machine with Data	55
2.1.4. The controller and data path.....	57

2.2.	Algorithm scheduling	60
2.3.	Loop pipelining.....	62
2.4.	Control statements pipelining	67
2.4.1.	Conditional statement pipelining	67
2.4.2.	Loop statement pipelining.....	69
2.4.3.	Pipelining of the CFG	70
2.5.	Memory handling.....	73
2.5.1.	Local arrays of data.....	74
2.5.2.	Explicit data caching.....	75
2.5.3.	Sequential-Access Buffering.....	78
2.6.	The FPGA-oriented algorithms	79
3.	Data-intensive algorithms for FPGAs	82
3.1.	Sorting and searching.....	82
3.2.	The sorting nets.....	83
3.3.	The merge sort tree	87
3.3.1.	The FPGA-accelerated sorting system.....	89
3.4.	The Bloom filter.....	91
3.4.1.	The parallel Bloom filter	92
3.4.2.	Enhancement of the Bloom filter	94
3.4.3.	Modified Cuckoo hashing	96
3.5.	A shifting substring search	98
3.6.	The binary tree.....	100
3.6.1.	The binary-tree processor.....	101
3.6.2.	Mapping of patterns to memories	103
3.7.	The prefix tree.....	104
3.8.	The Aho-Corasick algorithm	106
4.	The Hash Binary Tree.....	109
4.1.	Hashing of the binary tree patterns	109
4.2.	Two-fold pipelined HBT architecture.....	110
4.3.	Memory requirements of the HBT.....	111
4.4.	The example application.....	112
4.5.	Conclusions.....	114
5.	Acceleration of genome matching.....	116
5.1.	Short-read alignment.....	116

5.2. Subsequences 117

5.3. The trie 118

5.4. The sequential co-processor 119

5.5. The pipelined co-processor 121

5.6. Inexact matching 123

5.7. The control block 127

5.8. Resource requirements 128

5.9. Reducing software memory requirements 130

5.10. Implementation results 130

5.11. Conclusions 132

6. Final remarks 133

Bibliography 142