

| | |
|-------------------------|---|
| 1. Record Nr. | UNINA9910760277003321 |
| Autore | Jain Vikram |
| Titolo | Towards Heterogeneous Multi-Core Systems-on-Chip for Edge Machine Learning : Journey from Single-Core Acceleration to Multi-core Heterogeneous Systems // Vikram Jain and Marian Verhelst |
| Pubbl/distr/stampa | Cham, Switzerland : , : Springer, , [2024] ©2024 |
| ISBN | 3-031-38230-7 |
| Edizione | [First edition.] |
| Descrizione fisica | 1 online resource (199 pages) |
| Disciplina | 005.758 |
| Soggetti | Edge computing Machine learning Systems on a chip - Design and construction |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Nota di bibliografia | Includes bibliographical references and index. |
| Nota di contenuto | Intro -- Preface -- Acknowledgments -- Contents -- List of Abbreviations -- List of Figures -- List of Tables -- 1 Introduction -- 1.1 Machine Learning at the (Extreme) Edge -- 1.1.1 Applications -- 1.1.2 Algorithms -- 1.1.3 Hardware -- 1.2 Open Challenges for ML Acceleration at the (Extreme) Edge -- 1.3 Book Contributions -- 2 Algorithmic Background for Machine Learning -- 2.1 Support Vector Machines -- 2.2 Deep Learning Models -- 2.2.1 Neural Networks -- 2.2.2 Training -- 2.2.3 Inference: Neural Network Topologies -- 2.2.4 Model Compression -- 2.3 Feature Extraction -- 2.4 Conclusion -- 3 Scoping the Landscape of (Extreme) Edge Machine Learning Processors -- 3.1 Hardware Acceleration of ML Workloads: A Primer -- 3.1.1 Core Mathematical Operation -- 3.1.2 General Accelerator Template -- 3.2 Evaluation Metrics -- 3.3 Survey of (Extreme) Edge ML Hardware Platforms -- 3.4 Evaluating the Surveyed Hardware Platforms -- 3.5 Insights and Trends -- 3.6 Conclusion -- 4 Hardware-Software Co-optimization Through Design Space Exploration -- 4.1 Motivation -- 4.2 Exploration Methodology -- 4.2.1 ZigZag -- 4.2.2 Post-Processing of ZigZag's Results -- 4.3 DNN Workload Comparison -- 4.3.1 Exploration Setup -- 4.3.2 Visualization of the Complete Trade-Off Space -- 4.3.3 Impact of HW Architecture on Optimal Workload -- |

4.3.4 Impact of Workload on Optimal HW Architecture -- 4.4
Conclusion -- 5 Energy-Efficient Single-Core Hardware Acceleration --
5.1 Motivation -- 5.2 Metrics for Hardware Optimization -- 5.3 State-
of-the-Art in Object Detection on FPGA -- 5.4 Cost-Aware Algorithmic
Optimization -- 5.4.1 Object Detection Algorithms -- 5.4.2
Quantization of Tiny-YOLOv2 -- Post-training Quantization --
Quantization-Aware Training -- 5.5 Cost-Aware Architecture
Optimization -- 5.5.1 Hardware Mapping of Convolutional Layers.
5.5.2 Hardware Architecture of the Accelerator -- 5.6 Cost-Aware
System Optimization -- 5.6.1 Data Communication Architecture --
5.6.2 Tiling Strategy -- 5.7 Implementation Results -- 5.8 Conclusion
-- 6 TinyVers: A Tiny Versatile All-Digital Heterogeneous Multi-core
System-on-Chip -- 6.1 Motivation -- 6.2 Algorithmic Background --
6.2.1 Convolution and Dense Operation -- 6.2.2 Deconvolution --
6.2.3 Support Vector Machines (SVMs) -- 6.3 TinyVers Hardware
Architecture -- 6.3.1 Smart Sensing Modes for TinyML -- 6.3.2 Power
Management -- 6.4 FlexML Accelerator -- 6.4.1 FlexML Architecture
Overview -- 6.4.2 Dataflow Reconfiguration -- 6.4.3 Efficient Zero-
Skipping for Deconvolution and Blockwise Structured Sparsity -- 6.4.4
Support Vector Machine -- 6.5 Deployment of Neural Networks on
TinyVers -- 6.6 Design for Test and Fault-Tolerance -- 6.7 Chip
Implementation and Measurement -- 6.7.1 Peak Performance Analysis
-- 6.7.2 Workload Benchmarks -- 6.7.3 Power Management -- 6.7.4
Instantaneous Power Trace -- Keyword Spotting Application -- Machine
Monitoring Application -- 6.8 Comparison with SotA -- 6.9 Conclusion
-- 7 DIANA: Digital and ANALog Heterogeneous Multi-core System-on-
Chip -- 7.1 Motivation -- 7.2 Design Choices -- 7.2.1 Dataflow
Concepts -- 7.2.2 Design Space Exploration -- 7.2.3 A Reconfigurable
Heterogeneous Architecture -- 7.2.4 Optimization Strategies for Multi-
core -- 7.3 System Architecture -- 7.3.1 The RISC-V CPU and Network
Control -- 7.3.2 Memory System -- 7.4 AIMC Computing Core -- 7.4.1
AIMC Core Microarchitecture -- 7.4.2 Memory Control Unit (MCU) --
7.4.3 AIMC Macro -- 7.4.4 Output Buffer and SIMD Unit -- 7.5 Digital
DNN Accelerator -- 7.6 Measurements -- 7.6.1 Efficiency vs. Accuracy
Trade-Off in the Analog Macro -- 7.6.2 Peak Performance and
Efficiency Characterization -- 7.6.3 Workload Performance
Characterization.
7.6.4 SotA Comparison -- 7.7 Conclusion -- 8 Networks-on-Chip to
Enable Large-Scale Multi-core ML Acceleration -- 8.1 Motivation -- 8.2
Background -- 8.2.1 Network-on-Chips -- 8.2.2 AXI Protocol -- Burst
-- Multiple Outstanding Transaction -- 8.3 Interconnect Architecture of
PATRONoC -- 8.4 Implementation Results -- 8.5 Performance
Evaluation -- 8.5.1 Uniform Random Traffic -- 8.5.2 Synthetic Traffic
-- 8.5.3 DNN Workload Traffic -- 8.6 Related Work -- 8.7 Conclusion
-- 9 Conclusion -- 9.1 Overview and Contributions -- 9.2 Suggestions
for Future Work -- 9.2.1 The Low Hanging Fruits -- 9.2.2 Medium
Term -- 9.2.3 Moonshot -- 9.3 Closing Remarks -- References --
References -- Index.
