

1. Record Nr.	UNINA9910586635503321
Titolo	Embedded computer systems : architectures, modeling, and simulation : 22nd International Conference, SAMOS 2022, Samos, Greece, July 3-7, 2022, proceedings / / Alex Orailoglu, Marc Reichenbach, and Matthias Jung, editors
Pubbl/distr/stampa	Cham, Switzerland : , : Springer International Publishing, , [2022] ©2022
ISBN	3-031-15074-0
Descrizione fisica	1 online resource (434 pages)
Collana	Lecture Notes in Computer Science ; ; v.13511
Disciplina	004.16
Soggetti	Embedded computer systems Embedded computer systems - Design and construction
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Contents -- High Level Synthesis -- High-Level Synthesis of Digital Circuits from Template Haskell and SDF-AP -- 1 Introduction -- 2 Related Work -- 2.1 High-Level Synthesis Tools -- 2.2 Temporal Models for Hardware Design -- 3 SDF-AP -- 4 From Functional Description to SDF-AP -- 4.1 Template Haskell: The Toolflow -- 4.2 Conformance Relation -- 4.3 The Basic Idea of Combining SDF-AP with a Functional Language -- 4.4 The Advantages -- 4.5 The Current Limitations -- 5 Node Decomposition -- 6 Case Studies -- 6.1 Dot Product Case Study -- 6.2 Center of Mass Case Study -- 6.3 DCT2D Case Study -- 7 Conclusion -- References -- Implementing Synthetic Aperture Radar Backprojection in Chisel - A Field Report -- 1 Introduction -- 2 System Description -- 2.1 Synthetic Aperture Radar (SAR) -- 2.2 Backprojection (BP) -- 2.3 Architecture Overview -- 3 Measurements -- 3.1 Code Size -- 3.2 Transparency of Generated Code -- 4 Experience Using Chisel -- 5 Conclusion -- A Code Examples -- A.1 Chisel Code -- A.2 VHDL Code -- References -- EasyHBM: Simple and Fast HBM Access for FPGAs Using High-Level-Synthesis -- 1 Introduction -- 2 Related Work -- 3 Methodology -- 4 Architecture -- 4.1 PE High-Level Synthesis -- 4.2 HBM Connection -- 5 Evaluation -- 5.1 Design Complexity -- 5.2 Setup Overhead -- 5.3

Performance -- 6 Conclusion -- References -- Memory Systems --
TREAM: A Tool for Evaluating Error Resilience of Tree-Based Models
Using Approximate Memory -- 1 Introduction -- 2 System Model --
2.1 Memory and Error Model -- 2.2 Definitions for Error Resilience --
2.3 Research Questions -- 3 TREAM: An Extension to Sklearn -- 3.1
High-Level Overview of TREAM -- 3.2 Implementation -- 4 The Effect
of Bit Flips in Tree-Based Models -- 4.1 Bit Flips in Split and Feature
Value -- 4.2 Bit Flips in Child Indices.
4.3 Bit Flips in Feature Index -- 5 Evaluation -- 6 Conclusion --
References -- Split'n'Cover: ISO26262 Hardware Safety Analysis with
SystemC -- 1 Introduction -- 2 Background -- 3 Related Work -- 4
Methodology -- 5 Implementation -- 6 Experimental Case Study and
Results -- 7 Conclusion and Future Work -- References -- Tagged
Geometric History Length Access Interval Prediction for Tightly Coupled
Memory Systems -- 1 Introduction -- 2 Background and Related Work
-- 3 Access Interval Prediction -- 3.1 System Overview -- 3.2 Theory
and Notation -- 4 TAGE Access Interval Predictor -- 4.1 Design -- 4.2
Interval Representation -- 4.3 Subprediction and Update Method -- 5
Performance Evaluation -- 5.1 Interval Representation -- 5.2
Subprediction and Update Method -- 5.3 Comparison -- 6 Conclusion
and Outlook -- References -- Processor Architecture --
NanoController: A Minimal and Flexible Processor Architecture for
Ultra-Low-Power Always-On System State Controllers -- 1 Introduction
-- 2 Related Work -- 3 NanoController Architecture -- 3.1 Data Path --
3.2 Control Path and ISA -- 3.3 Program Code Generation -- 4
Evaluation -- 4.1 Exemplary Control Application -- 4.2 Code Size
Comparison -- 4.3 Performance Comparison -- 4.4 Silicon Area and
Power/Energy Consumption -- 5 Conclusion -- References --
ControlPULP: A RISC-V Power Controller for HPC Processors with
Parallel Control-Law Computation Acceleration -- 1 Introduction -- 2
Related Work -- 3 The ControlPULP Platform -- 3.1 System Architecture
-- 3.2 Power Control Firmware -- 4 Experimental Results -- 4.1 Area
Evaluation -- 4.2 Firmware Control Action -- 4.3 In-band Services --
4.4 System-Level PCF Step Evaluation -- 4.5 Control-Level PCF Step
Evaluation -- 5 Conclusion -- References -- Embedded Software
Systems and Beyond.
CASA: An Approach for Exposing and Documenting Concurrency-
Related Software Properties -- 1 Introduction -- 2 Related Work -- 3
Exposing and Documenting Concurrency-Related Software Architecture
Decisions -- 3.1 Guideline for Identification of Concurrency-Critical
Components (GICCS) -- 3.2 Documentation of Synchronisation
Mechanisms -- 3.3 Testing of Concurrent Software with CASA's Support
-- 4 Integration of CASA in Agile SCRUM -- 5 Evaluation -- 5.1 Setup I:
Reverse Engineering of Synchronisation Decisions in Industrial
Environment -- 5.2 Setup II: Survey with Software Engineering
Professionals -- 5.3 Threats to Validity -- 6 Conclusion -- References
-- High-Level Simulation of Embedded Software Vulnerabilities to EM
Side-Channel Attacks -- 1 Introduction -- 2 Related Work -- 3 EM
Synthesis Flow -- 3.1 Dictionary Construction -- 3.2 Synthesis -- 4
Experiments -- 4.1 Accuracy and Dictionary Cost -- 4.2 Case Study:
Compile-Time Control Flow Prediction -- 5 Summary and Conclusions
-- References -- Deep Learning Optimization I -- A Design Space
Exploration Methodology for Enabling Tensor Train Decomposition
in Edge Devices -- 1 Introduction -- 2 Background and Related Works
-- 2.1 Low-Rank Factorization -- 2.2 Tensor-Train (TT) Format
and T3F Library -- 2.3 Motivation -- 3 Methodology -- 4 Experimental
Results -- 5 Conclusion -- References -- Study of DNN-Based
Ragweed Detection from Drones -- 1 Introduction -- 2 Related Work --

2.1 Weed Detection and Eradication -- 2.2 Deep Learning and Model Compression -- 3 System Architecture -- 3.1 Ground Sampling Distance -- 3.2 Flight Parameter Model -- 3.3 Dataset -- 3.4 Object Detection Network -- 3.5 Semantic Segmentation Network -- 4 Results -- 4.1 Drone Selection -- 4.2 Experimental Setup -- 5 Conclusion and Future Work -- References.

PULP-TrainLib: Enabling On-Device Training for RISC-V Multi-core MCUs Through Performance-Driven Autotuning -- 1 Introduction -- 2 Related Work -- 3 On-Device Training on PULP -- 3.1 The PULP Platform -- 3.2 PULP-TrainLib Library -- 3.3 Accelerating SW Training Primitives -- 3.4 AutoTuner -- 4 Experimental Results -- 4.1 Latency Optimization on PULP-TrainLib -- 4.2 Effect of Tensor Shapes on AutoTuning -- 4.3 TinyML Benchmark Training -- 4.4 Comparison with the State of the Art -- 5 Conclusion -- References -- Extra-Functional Property Estimation -- The Impact of Dynamic Storage Allocation on CPython Execution Time, Memory Footprint and Energy Consumption: An Empirical Study -- 1 Introduction -- 1.1 Contributions -- 2 Method -- 3 Experimental Setup -- 3.1 Rationale -- 3.2 PGO, LTO Sensitivity -- 3.3 Configuration Points -- 3.4 Benchmarking Script -- 3.5 Platform Independence -- 4 Results -- 4.1 Discussion -- 5 Limitations -- 6 Related Work -- 7 Conclusions -- References -- Application Runtime Estimation for AURIX Embedded MCU Using Deep Learning -- 1 Introduction -- 1.1 Performance Modeling Approaches -- 1.2 Structure of the Paper -- 2 Related Work -- 3 Methodology -- 3.1 Mechanistic Model, Based on QEMU -- 3.2 Using Artificial Neural Networks for Performance Estimation -- 4 Training Data -- 4.1 Preprocessing of the Training Data -- 4.2 Analysis of the Training Data -- 5 Evaluation -- 5.1 Hyperparameter Search -- 5.2 Architectures Chosen for the Evaluation with Program Traces -- 6 Conclusion and Outlook -- References -- A Hybrid Performance Prediction Approach for Fully-Connected Artificial Neural Networks on Multi-core Platforms -- 1 Introduction -- 2 Related Work -- 3 Proposed Modeling Approach -- 3.1 Target Application: Fully-Connected ANNs -- 3.2 Modeling ANNs with SDF -- 3.3 Mapping on the Targeted Architecture -- 3.4 Computation Time Model.

3.5 Simulation Using SystemC Models -- 4 Experiments -- 4.1 Testing Configuration -- 4.2 Experiment Results -- 4.3 Exploration of Partitionings and Mappings -- 5 Conclusion -- References -- Deep Learning Optimization II -- A Smart HW-Accelerator for Non-uniform Linear Interpolation of ML-Activation Functions -- 1 Introduction -- 2 Related Work -- 2.1 Lookup Table Approximation -- 2.2 Piecewise Linear Approximation (PWL) -- 2.3 Piecewise Nonlinear Approximation (PWNL) -- 2.4 CORDIC -- 3 Proposed Approach -- 3.1 Overview -- 3.2 HW Generator -- 3.3 Integrated Features -- 3.4 Software Preprocessing -- 3.5 Proposed Architecture -- 3.6 Multiplier Optimizations -- 4 Experimental Results -- 4.1 Optimized Search for Interpolation Points -- 4.2 Optimized Generated Hardware -- 4.3 Comparison with Similar Works -- 5 Conclusion -- 6 Future Work -- References -- Hardware-Aware Evolutionary Filter Pruning -- 1 Introduction -- 2 Related Work -- 3 Fundamentals -- 3.1 Filter Pruning of CNNs -- 3.2 Inference Time Analysis of Convolutional Layers on GPUs -- 4 Design Space Exploration -- 4.1 Search Strategies -- 5 Experiments -- 5.1 Influence of Number of Groups G -- 5.2 VGG-11 and VGG-16 on CIFAR-10 -- 5.3 Retraining with Reduced Data Set During DSE -- 5.4 ResNet-18 and ResNet-101 on ILSVRC-2012 -- 6 Conclusion -- References -- Innovative Architectures and Tools for Security -- Obfuscating the Hierarchy of a Digital IP -- 1 Introduction -- 2 Proposed Approach -- 2.1 Motivation and Background -- 2.2 Hierarchical Obfuscation -- 3

Results -- 3.1 Power-Performance-Area Evaluations -- 3.2 Security
Analysis -- 4 Conclusion -- References -- On the Effectiveness of True
Random Number Generators Implemented on FPGAs -- 1 Introduction
-- 2 Methodology -- 2.1 Digital Noise Sources -- 2.2 Post-processing
Methods -- 3 Experimental Evaluation -- 3.1 Setup -- 3.2 Resource
Utilization.
3.3 Throughput.
