1. Record Nr. UNINA9910555252703321

Titolo	Multi-processor system-on-chip . 1 Architectures / / coordinated by Liliana Andrade, Frederic Rousseau
Pubbl/distr/stampa	London : , : ISTE Hoboken, NJ : , : John Wiley & Sons, Inc., , 2020
ISBN	1-119-81829-X 1-119-81827-3
Descrizione fisica	1 online resource (321 pages)
Disciplina	621.3815
Soggetti	Systems on a chip Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Cover -- Half-Title Page -- Dedication -- Title Page -- Copyright Page -- Contents -- Foreword -- Acknowledgments -- PART 1: Processors -- 1 Processors for the Internet of Things -- 1.1. Introduction -- 1.2. Versatile processors for low-power IoT edge devices -- 1.2.1. Control processing, DSP and machine learning -- 1.2.2. Configurability and extensibility -- 1.3. Machine learning inference -- 1.3.1. Requirements for low/mid-end machine learning inference -- 1.3.2. Processor capabilities for low-power machine learning inference -- 1.3.3. A software library for machine learning inference -- 1.3.4. Example machine learning applications and benchmarks -- 1.4. Conclusion -- 1.5. References -- 2 A Qualitative Approach to Many-core Architecture -- 2.1. Introduction -- 2.2. Motivations and context -- 2.2.1. Many-core processors -- 2.2.2. Machine learning inference -- 2.2.3. Application requirements -- 2.3. The MPPA3 many-core processor -- 2.3.1. Global architecture -- 2.3.2. Compute cluster -- 2.3.3. VLIW core -- 2.3.4. Coprocessor -- 2.4. The MPPA3 software environments -- 2.4.1. High-performance computing -- 2.4.2. KaNN code generator -- 2.4.3. High-integrity computing -- 2.5. Conclusion -- 2.6. References -- 3 The Plural Many-core Architecture - High Performance at Low Power -- 3.1. Introduction -- 3.2. Related works -- 3.3. Plural many-core architecture -- 3.4. Plural programming model -- 3.5.

Plural hardware scheduler/synchronizer -- 3.6. Plural networks-on-chip -- 3.6.1. Scheduler NoC -- 3.6.2. Shared memory NoC -- 3.7. Hardware and software accelerators for the Plural architecture -- 3.8. Plural system software -- 3.9. Plural software development tools -- 3.10. Matrix multiplication algorithm on the Plural architecture -- 4 ASIP-Based Multi-Processor Systems for an Efficient Implementation of CNNs -- 4.1. Introduction -- 4.2. Related works.
4.3. ASIP architecture -- 4.4. Single-core scaling -- 4.5. MPSoC overview -- 4.6. NoC parameter exploration -- 4.7. Summary and conclusion -- 4.8. References -- PART 2: Memory -- 5 Tackling the MPSoC DataLocality Challenge -- 5.1. Motivation -- 5.2. MPSoC target platform -- 5.3. Related work -- 5.4. Coherence-on-demand: region-based cache coherence -- 5.4.1. RBCC versus global coherence -- 5.4.2. OS extensions for coherence-on-demand -- 5.4.3. Coherency region manager -- 5.4.4. Experimental evaluations -- 5.4.5. RBCC and data placement -- 5.5. Near-memory acceleration -- 5.5.1. Near-memory synchronization accelerator -- 5.5.2. Near-memory queue management accelerator -- 5.5.3. Near-memory graph copy accelerator -- 5.5.4. Near-cache accelerator -- 5.6. The big picture -- 5.7. Conclusion -- 5.8. Acknowledgments -- 5.9. References -- 6 mMPU: Building a Memristor-based General-purpose In-memory Computation Architecture -- 6.1. Introduction -- 6.2. MAGIC NOR gate -- 6.3. In-memory algorithms for latency reduction -- 6.4. Synthesis and in-memory mapping methods -- 6.4.1. SIMPLE -- 6.4.2. SIMPLER -- 6.5. Designing the memory controller -- 6.6. Conclusion -- 6.7. References -- 7 Removing Load/Store Helpers in Dynamic Binary Translation -- 7.1. Introduction -- 7.2. Emulating memory accesses -- 7.3. Design of our solution -- 7.4. Implementation -- 7.4.1. Kernel module -- 7.4.2. Dynamic binary translation -- 7.4.3. Optimizing our slow path -- 7.5. Evaluation -- 7.5.1. QEMU emulation performance analysis -- 7.5.2. Our performance overview -- 7.5.3. Optimized slow path -- 7.6. Related works -- 7.7. Conclusion -- 7.8. References -- 8 Study and Comparison of Hardware Methods for Distributing Memory Bank Accesses in Many-core Architectures -- 8.1. Introduction -- 8.1.1. Context -- 8.1.2. MPSoC architecture -- 8.1.3. Interconnect -- 8.2. Basics on banked memory.
8.2.1. Banked memory -- 8.2.2. Memory bank conflict and granularity -- 8.2.3. Efficient use of memory banks: interleaving -- 8.3. Overview of software approaches -- 8.3.1. Padding -- 8.3.2. Static scheduling of memory accesses -- 8.3.3. The need for hardware approaches -- 8.4. Hardware approaches -- 8.4.1. Prime modulus indexing -- 8.4.2. Interleaving schemes using hash functions -- 8.5. Modeling and experimenting -- 8.5.1. Simulator implementation -- 8.5.2. Implementation of the Kalray MPPA cluster interconnect -- 8.5.3. Objectives and method -- 8.5.4. Results and discussion -- 8.6. Conclusion -- 8.7. References -- PART 3: Interconnect and Interfaces -- 9 Network-on-Chip (NoC): The Technology that Enabled Multi-processor Systems-on-Chip (MPSoCs) -- 9.1. History: transition from buses and crossbars to NoCs -- 9.1.1. NoC architecture -- 9.1.2. Extending the bus comparison to crossbars -- 9.1.3. Bus, crossbar and NoC comparison summary and conclusion -- 9.2. NoC configurability -- 9.2.1. Human-guided design flow -- 9.2.2. Physical placement awareness and NoC architecture design -- 9.3. System-level services -- 9.3.1. Quality-of-service (QoS) and arbitration -- 9.3.2. Hardware debug and performance analysis -- 9.3.3. Functional safety and security -- 9.4. Hardware cache coherence -- 9.4.1. NoC protocols, semantics and messaging -- 9.5. Future NoC technology developments -- 9.5.1. Topology synthesis and floorplan awareness -- 9.5.2.