

Multiprocessor System-on-chip

[DOWNLOAD HERE](#)

1;Preface;6 2;Contents;8 3;Chapter 1: An Introduction to Multi-Core System on Chip - Trends and Challenges;10 3.1;1.1 From SoC to MPSoC;10 3.2;1.2 General Structure of MPSoC;11 3.2.1;1.2.1 Processing Elements;11 3.2.2;1.2.2 Interconnection;12 3.2.3;1.2.3 Power Management;12 3.3;1.3 Power Efficiency and Adaptability;13 3.4;1.4 Complexity and Scalability;15 3.5;1.5 Heterogeneous and Homogeneous Approaches;16 3.5.1;1.5.1 Heterogeneous MPSoC;16 3.5.2;1.5.2 Homogeneous MPSoC;17 3.6;1.6 Multi variable Optimization;20 3.6.1;1.6.1 Static Optimization;20 3.6.2;1.6.2 Dynamic Optimization;20 3.6.2.1;1.6.2.1 Centralized Approaches;21 3.6.2.2;1.6.2.2 Distributed Approaches;24 3.7;1.7 Static vs Dynamic Centralized and Distributed Approaches;25 3.8;1.8 Conclusion;27 3.9;References;28 4;Part I: Application Mapping and Communication Infrastructure ;31 4.1;Chapter 2: Composability and Predictability for Independent Application Development, Verification, and Execution;32 4.1.1;2.1 Introduction;32 4.1.2;2.2 Composability and Predictability;34 4.1.2.1;2.2.1 Terminology;34 4.1.2.2;2.2.2 Composable Resources;38 4.1.2.3;2.2.3 Predictable resources;41 4.1.2.4;2.2.4 Composable and predictable resources;42 4.1.3;2.3 Processor tile;45 4.1.3.1;2.3.1 Composability;45 4.1.3.1.1;2.3.1.1 Constant task slots;46 4.1.3.1.2;2.3.1.2 Constant OS slot;47 4.1.3.1.3;2.3.1.3 Two-level application and task scheduling;48 4.1.3.2;2.3.2 Predictability;48 4.1.4;2.4 Interconnect;49 4.1.4.1;2.4.1 Composability;50 4.1.4.2;2.4.2 Predictability;51 4.1.5;2.5 Memory tile;51 4.1.5.1;2.5.1 Predictability;52 4.1.5.1.1;2.5.1.1 Predictable SDRAM back-end;52 4.1.5.1.2;2.5.1.2 Predictable arbitration;55 4.1.5.2;2.5.2 Composability;56 4.1.6;2.6 Experiments;57 4.1.7;2.7 Conclusions;59 4.1.8;References;61 4.2;Chapter 3: Hardware Support for Efficient Resource Utilization in Manycore Processor Systems;64 4.2.1;3.1 Introduction;65 4.2.2;3.2 Learning from Network Processing Applications;67 4.2.2.1;3.2.1 Commercial Network Processors;68 4.2.2.2;3.2.2 Example Networking Applications;69 4.2.2.3;3.2.3 The FlexPath NP Approach;70 4.2.2.4;3.2.4 What Can Other Manycore Domains Learn from Network Processing?;75 4.2.3;3.3 Learning from HPC and Scientific Computing;77 4.2.3.1;3.3.1 Hierarchical Multi-Topology Networks-on-Chip;77 4.2.3.2;3.3.2 Task Management;81 4.2.3.3;3.3.3 Synchronization Subsystem;82 4.2.3.4;3.3.4 What Can Other Manycore Domains Learn from Supercomputing?;83

4.2.4;3.4 Learning from Bio-Inspired, Self-Organizing Systems in Nature;84 4.2.4.1;3.4.1 Collective Behavior of Entities in Natural and Technical Systems;84 4.2.4.2;3.4.2 Technical Realization of Self-Adaptive IP Cores;86 4.2.4.3;3.4.3 What Can Manycore Domains Learn from Nature?;90 4.2.5;3.5 Summary and Conclusions;92 4.2.6;References;93 4.3;Chapter 4: PALLAS: Mapping Applications onto Manycore;95 4.3.1;4.1 PALLAS;96 4.3.2;4.2 Driving Applications;97 4.3.2.1;4.2.1 Content-Based Image Retrieval;97 4.3.2.2;4.2.2 Optical Flow and Tracking;99 4.3.2.3;4.2.3 Stationary Video Background Subtraction;101 4.3.2.4;4.2.4 Automatic Speech Recognition;102 4.3.2.5;4.2.5 Compressed Sensing MRI;103 4.3.2.6;4.2.6 Market Value-at-Risk Estimation in Computational Finance;105 4.3.2.7;4.2.7 Games;106 4.3.2.8;4.2.8 Machine Translation;107 4.3.2.9;4.2.9 Summary;108 4.3.3;4.3 Perspectives on Parallel Performance;109 4.3.3.1;4.3.1 Linear Scaling Not Required;109 4.3.3.2;4.3.2 Measure Real Problems on Real Hardware;110 4.3.3.3;4.3.3 Consider the Algorithms;110 4.3.3.4;4.3.4 Summing Up;111 4.3.4;4.4 Patterns to Frameworks;111 4.3.4.1;4.4.1 Application Frameworks;112 4.3.4.2;4.4.2 Programming Frameworks;114 4.3.4.2.1;4.4.2.1 Efficiency and Portability Through Programming Frameworks;114 4.3.4.2.2;4.4.2.2 Copperhead;115 4.3.5;4.5 Conclusions;116 4.3.6;4.6 Appendices;117 4.3.6.1;4.6.1 Structural Patterns;117 4.3.6.2;4.6.2 Computational Patterns;117 4.3.6.3;4.6.3 Parallel EAN/ISBN : 9781441964601 Publisher(s): Springer, Berlin, Springer Science & Business Media Discussed keywords: Embedded Systems, Multiprocessor, System-on-Chip (SoC) Format: ePub/PDF Author(s): Hbner, Michael - Becker, Jrgen

[DOWNLOAD HERE](#)

Similar manuals:

[Advances In Design And Specification Languages For Embedded Systems](#)

[Advances In Design Methods From Modeling Languages For Embedded Systems And SoC's](#)

[Design Methodologies For Secure Embedded Systems](#)

[Design Technology For Heterogeneous Embedded Systems](#)

[Embedded Systems For Smart Appliances And Energy Management](#)

[Embedded Systems Specification And Design Languages](#)

[Enhancing Embedded Systems Simulation](#)

[From Model-Driven Design To Resource Management For Distributed Embedded Systems](#)

[Languages For Embedded Systems And Their Applications](#)

[Memory Controllers For Real-Time Embedded Systems](#)

[Pro Linux Embedded Systems](#)

[Radiation Effects On Embedded Systems](#)

[Solutions On Embedded Systems](#)

[Spatial Awareness Of Autonomous Embedded Systems](#)

[System-Level Design Techniques For Energy-Efficient Embedded Systems](#)

[Views On Evolvability Of Embedded Systems](#)

[Memory Allocation Problems In Embedded Systems](#)

[Embedded Systems](#)

[Multiplexed Networks For Embedded Systems](#)

[How To Land A Top-Paying Embedded Systems Software Developers Job: Your Complete Guide To Opportunities, Resumes And Cover Letters, Interviews, Salari - Rachel Matthews](#)

[FreeBSD For Developers - Develop Embedded Systems Ver 2011](#)

[Chapter 23, Specifying Behavior Of Embedded Systems - Robert Oshana](#)

[Chapter 03, Overview Of Real-time And Embedded Systems - Robert Oshana](#)

[Chapter 01, DSP In Embedded Systems: A Roadmap - Robert Oshana](#)

[Chapter 04, Overview Of Embedded Systems Development Lifecycle Using DSP - Robert Oshana](#)