ABSTRACT: The applications for multicore and manycore microprocessors as RISC-V are currently useful for the advantages of their friendly nature, compared to previous chips, which have caused a great demand for these multi-core or many-core processors used in parallel computing for fluid emulation mainly on the atmosphere of the earth, and other applications.

The result of this research, is the focus in determining the factors that influence high-performance systems, after reviewing and considering various authors, for its realization. This research focuses on two factors which influence on the data in computer systems shared memory (multi-core and many-core architecture) being these topology and memory consistency.
Factors which influence the performance of Manycore System operating Topology

- Algorithms
- System operating
- Architecture
- Programming Model
- Languages programming
- Memory Cache
- Memory Model
- Technology
- Homogeneous
- Heterogeneous
- Taxonomy
Software-Defined Error-Correcting Codes

Errors in memory often result in system-level crashes.

Current error-correction techniques are costly and are oblivious to the underlying data stored in memory.

SDECC pushes beyond current error-correction capabilities by combining three layers:

- **System-level fault tolerance**
- **Error-correcting codes**
- **Side-information** about data and instructions in memory

→ RISC-V! 😊
WASP-SC

Austin Harris, Rohith Prakash
The University of Texas at Austin SPARK Lab

• Goal: defend against utilization side-channels
  • E.g. shared memory controllers, hardware accelerators
  • Normalization (e.g. partitioning, worst-case) infeasible

• Solution: shape victim’s utilization to be statistically indistinguishable across different inputs
  • Optimally minimizes slowdown within provably configurable privacy bounds

• Modify Rocket to have cores sharing SHA3 accelerator
  • Send commands through queue with our traffic shaping defense
Sub-microsecond Adaptive Voltage Scaling in a 28nm RISC-V SoC

Demo: Running user-mode programs in Linux on RISC-V silicon to demonstrate integrated power management
The Death of Moore’s Law Will Spur Innovation

As transistors stop shrinking, open-source hardware will have its day

By Andrew bunnie Huang
Posted 31 Mar 2015 | 15:00 GMT

Exascale Computing: Embedded Style

TeraFlop
Embedded

PetaFlop
Departmental

ExaFlop
Data Center

Courtesy Kogge et als 2008
UNNI: An open source core for easy transition from ARM Cortex-M0 to RISC-V

• Von Neumann architecture with 2-stage pipeline
• Optimized for ASIC implementation
• Written in SystemVerilog
• Low latency interrupt handling with tail-chaining and pre-emption

Mikael Korpi
OKiM Technologies
Full-Featured RISC-V Debug Solution

Tim Newsome <tim@sifive.com>

• It works in silicon!
• Download directly to flash

Demo Setup:

Implementations
• SiFive CorePlex in silicon
• IQ-Analog NanoRisc5 on FPGA

Open Source
• Rocket Chip implementation
• gdb and OpenOCD code
• Black box testsuite

More Information
• Debug list at https://riscv.org/mailing-lists/
Syntacore RISC-V cores demos
Alexander Redkin

5th RISC-V workshop
Nov 29-30 2016

info@syntacore.com
www.syntacore.com
Syntacore introduction

IP company

1. Develops and licenses energy-efficient programmable cores
   ▫ With RISC-V ISA

2. Full service to specialize these for the customer needs
   ▫ Workload analysis/characterization
   ▫ Workload-specific customization
     ▪ with tools/compiler support
   ▫ IP hardening at the required library node
   ▫ SoC integration and SW migration support
Baseline SCRx cores

SCRx: a family of the state-of-the-art RISC-V compatible synthesizable processor cores

- SCR1: RV32IC[EM]
- SCR3: RV32IMC[E] \(<=\) demo
- SCR4: RV32IMCF[D]
- SCR5: RV[32|64]IM[A]CFD \(<=\) demo

Stable, configurable, available for evaluation
Baseline: every core can be extended/customized
Thank you!
ASIP Designer - Automating ASIP Design
Architecture Definition, Optimization, and Implementation

- ASIP Designer creates full SDK
  - Compiler-in-the-loop optimization
- Process starts with pre-existing example models
  - RISC-V Starting point
- ASIP Designer generates synthesizable RTL
  - Performance/Power/Area
- Analysis seeds refinement/optimization
  - ASIP model is refined
  - SDK is automatically adapted
  - All elements stay in-sync

1. SDK Generation
2. Architectural Optimization
3. HW Generation
SHAVE: Software/Hardware Assurance Verified End-to-End

- Create practical, end-to-end assurance cases for mission critical software/hardware systems that run on COTS hardware.

- Case study: implement a crypto extension to RISC-V (like AES-NI), build a thin firmware layer and small application on top, and create an assurance case.

Sponsored by the Air Force Research Laboratory (AFRL) and developed with funding from the Defense Advanced Research Projects Agency (DARPA) under contract number FA8650-16-C-7665. Any views, opinions, findings, conclusions and/or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force, the Department of Defense or the U.S. Government.
LibreChainEDA

Open Source IC Design Tool Flow Scripts
Complete FOSS from concept to FPGA
100% Open Source EDA Tools
IP-Xact Based tool flows
Design for Reuse best practices
One stop shopping for installation