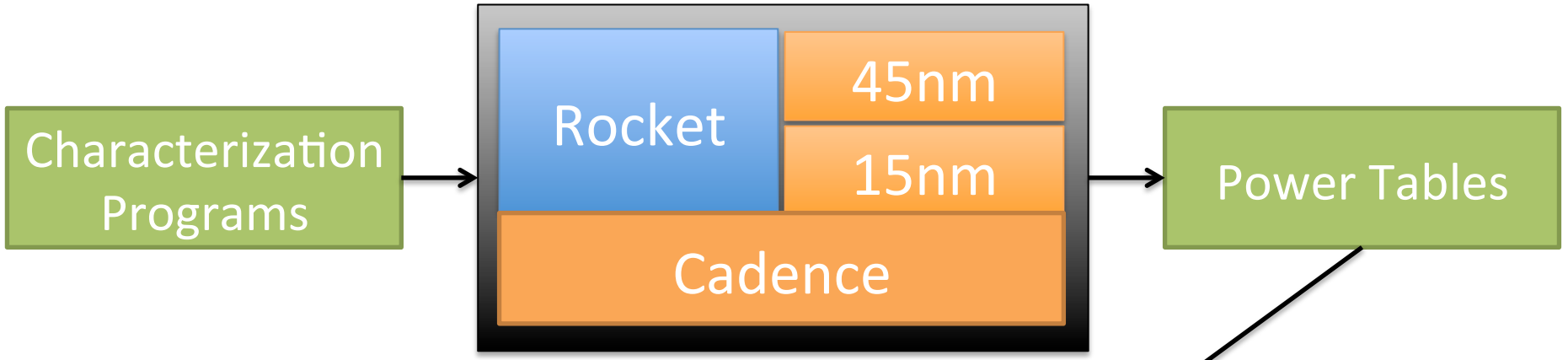
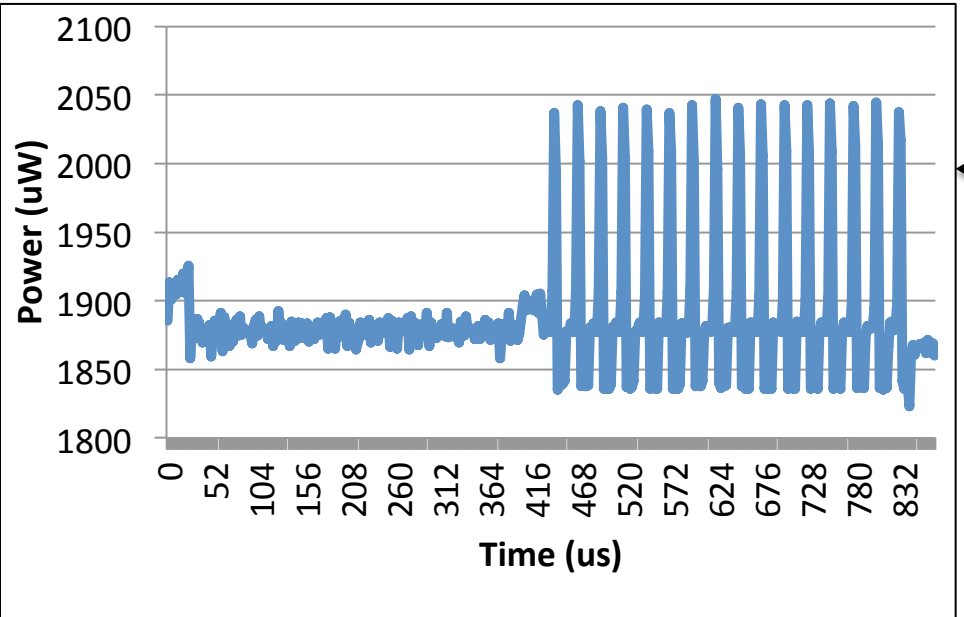
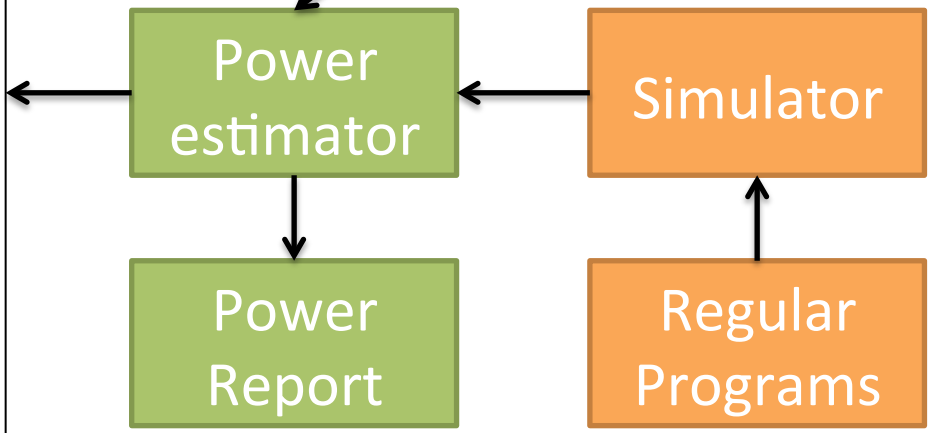
Power Models for RISC-V Processor

Rodolfo Azevedo and Carlos Petry
University of Campinas - Brazil







Sponsored by LGE and FAPESP

ASPIRE UC Berkeley

BOOM: An Industry-Competitive, Synthesizable, Parameterized RISC-V Processor Written in Chisel

- Implements RISC-V (RV64G) + privileged ISA
- Superscalar
- 10,000 lines of Chisel
- A part of the rocket-chip SoC ecosystem
- Open-source!
  - https://ucb-bar.github.io/riscv-boom
- Design document available at:
  - https://ccelio.github.io/riscv-boom-doc
- I accept pull requests!

Chris Celio

Advised by Krste Asanovic & Dave Patterson

Berkeley Chip Projects (Palmer Dabbelt)

Low Energy DSP
- ST 28FDSOI
- Rocket
- Hwacha

Rack-Scale Server
- TSMC 28HPM
- BOOM
- V Extension

Radio Baseband
- TSMC 16FF+
- Rocket
- FFT Unit

Agile Hardware Development:
- Rocket Chip as the base
- Chip specific parts in Chisel
  - Hwacha
  - Accelerators
  - Power Management Unit
  - Multi-Clock
- Automated physical design
- Build fast, custom chips quickly

X-FILES/DANA: RISC-V Hardware/Software for Neural Networks

Schuyler Eldridge, Han Dong, Thomas Unger, Marcia Sahaya Louis, Leila Delshad Tehrani, Jonathan Appavoo, and Ajay Joshi

Hardware and Software

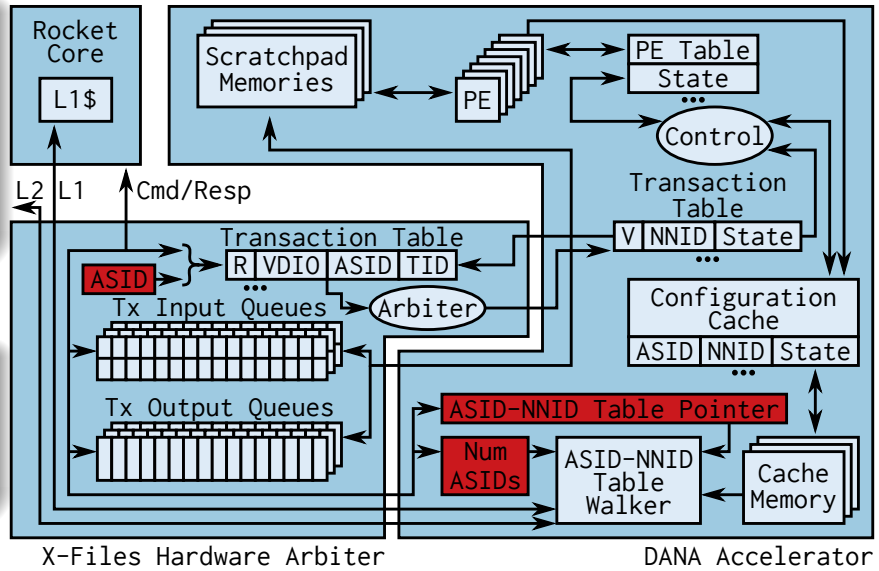
- **X-FILES:** Hardware/software for neural network management
- **DANA:** An X-FILES backend for neural network acceleration

Operating System Support

- RISC-V Proxy Kernel systemcalls
- RISC-V Linux Port (ongoing)

Open Source Availability

- Chisel hardware, C libraries [2]



[1] Eldridge, S., Waterland, A., Seltzer, M., Appavoo, J., and Joshi, A. (2015). Towards general-purpose neural network computing. In *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques (PACT)*.

[2] Boston University Integrated Circuits and Systems Group (2016). X-files/dana github repository. Online: `https://github.com/bu-icsg/xfiles-dana`.

BOSTON UNIVERSITY

AXIOM Gamma 4K open source modular camera with RISC-V inside

- First truly open, professional 4K film-making camera
- Xilinx Zynq FPGA SoC and Kintex-7 FPGA
- CMV12000 Super-35 300 FPS 4K sensor
- Full modularity and extensibility
- Offered as computer vision platform
- Antmicro's Nvidia TK1 module with Android and CUDA support
- Antmicro responsible for design and development of FPGA firmware and software

RISC-V in AXIOM

- Chisel implementation of a 3-stage pipeline RISC-V Z-Scale CPU
- Drives the communication between the camera modules and controls/configures the image sensor
- Example of how the RISC-V architecture can be successfully used in practical, high-profile applications

Learn more at the Antmicro Open Source portal: http://antmicro.com/OpenSource/axiom/

af inventions embedded electronics

die Angewandte ARTSCIENCE

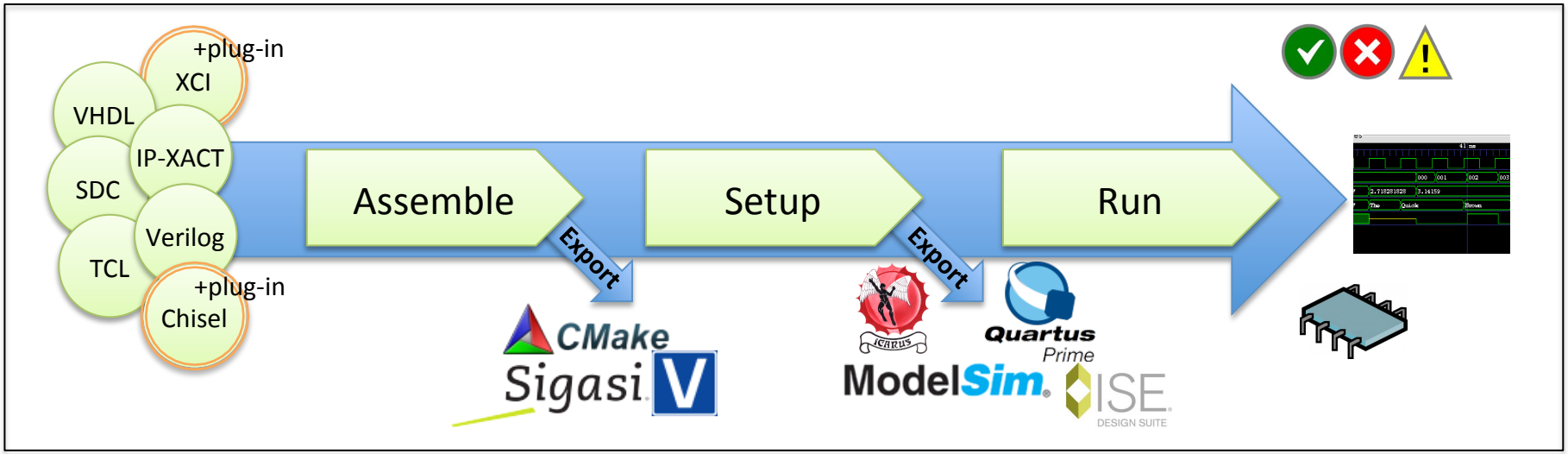
antmicro EMBEDDED SYSTEMS

apertus° open source cinema

PRÄZISIONS-ENTWICKLUNG pe deniz DENZ

FuseSoC - a package manager and build system for HDL code.

The main purpose is to increase reuse of IP cores and be an aid for creating, building and simulating SoC solutions.
The standard FuseSoC core library consists of more than 100 cores.
FuseSoC can be used as an end-to-end flow or integrate with your existing build system.



FuseSoC is:

- Modular
- Extendable
- Unassuming
- Standard-compliant
- Free software
- Battle-proven

FuseSoC makes it easier to:

- Reuse existing cores.
- Create compile-time or run-time configurations.
- Run regression tests against multiple simulators.
- Port designs to new targets.
- Let other projects use your code.
- Set up continuous integration .

Try it yourself at:
https://github.com/olofk/fusesoc
Or run: *pypi install fusesoc*

OlofKindgren
https://www.linkedin.com/in/olofkindgren
http://olofkindgren.blogspot.com/

qamcom FOSSi Foundation

MINIMA PROCESSOR

Microwatt RISC-V

- Energy per operation minimum at ~0.35-0.4V for modern silicon processes
- Since E = C*V^2, Energy-per-Instruction can be 20x lower than at nominal VDD
- Key challenge: yield loss due to process variance
- The Minima Solution: adaptive Timing Event Processing (TEP) under software control
  - TEP adapts at run time, limiting the need for massive margins
  - Always operate at minimum energy for given task, data, ambient condition (under OS controlled)

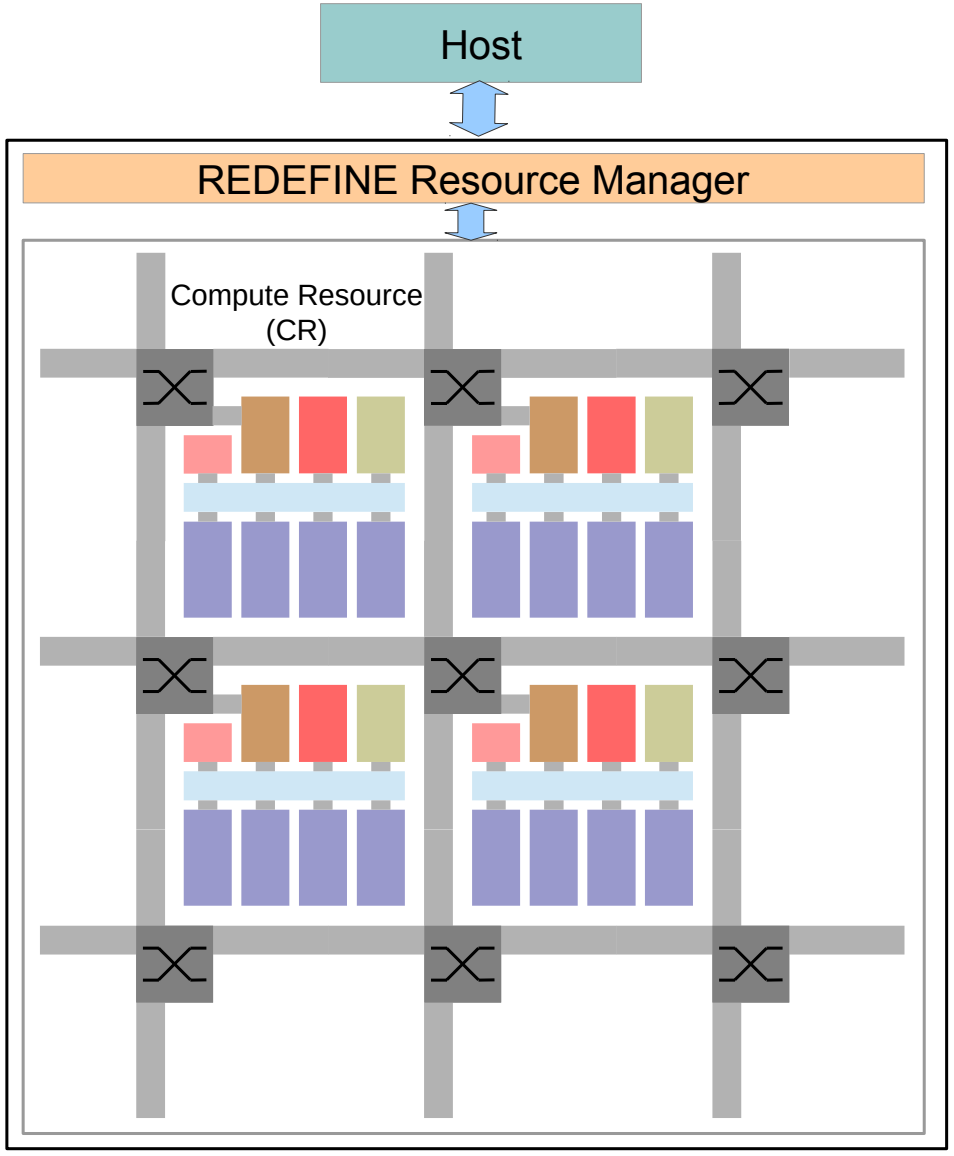
Preliminary power and performance for Rocket (28nm FDSOI, default configuration, without caches)					
VDD	***Clock Frequency***	***P_{DYN}***	***P_{LEAK}***	***P_{TOTAL}***	***Area***
0,4V	***15 MHz***	***7.5 µW***	***4.8 µW***	***12.3 µW***	***0.22 mm^2***

REDEFINE: A RISC-V ISA Based Runtime Reconfigurable Massively Parallel Processor for High Performance Applications

Ranjani Narayan, Chandan Haldar, Madahava Krishna, Kavita Madhu, S. K. Nandy

Indian Institute of Science

Morphing Machines



- Problems addressed:
  - *GPP: Programmability at the cost of power, granularity of dyadic operations*
  - *ASICs: Performance at the cost of flexibility*
- REDEFINE, our solution:
  - *Programming: HLL front-end*
  - *Performance: ASIC-like, with Domain Customized H/W*
  - *Scalable: Packet-switched Network on Chip*
  - *Reconfigurable: Dynamically created heterogeneous cores*
  - *RISC V based ISA (RV32 I, M, F), extended by REDEFINE-specific instructions*
- USP of REDEFINE:
  - *Multi-grain Dataflow execution:*
    - *Asynchronous and Self-scheduling*
    - *Exploitation of dynamic parallelism*
  - *Dynamic provisioning of resources:*
    - *Minimal communication interference*
    - *Opportunity for creation of "Safe-zones"*
  - *Compile-time WCET analysis:*
    - *Determinacy, Dependability*

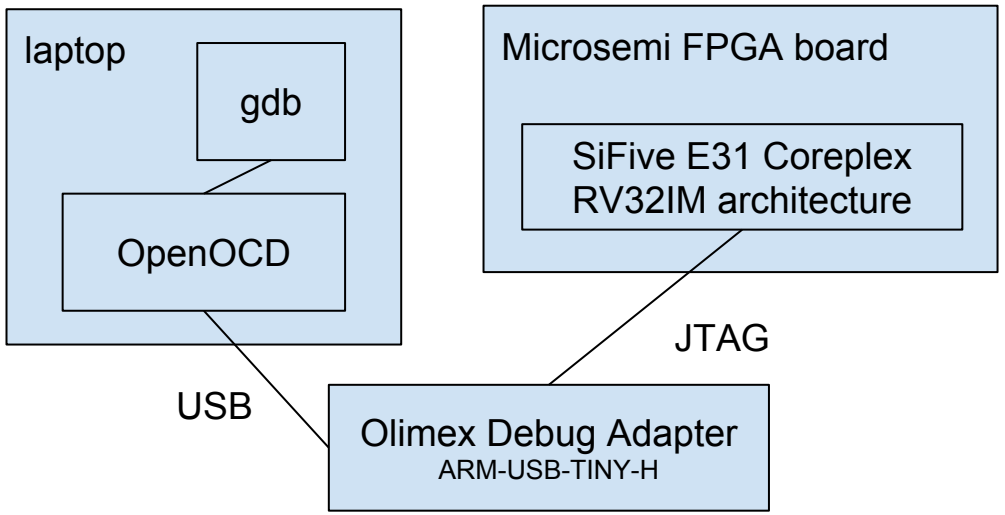
Router | Inter-router Connections | Power Control Unit | Distributed Shared Memory Bank | Runtime Support | Debug Unit | RISC-V CE (CFU) | Intra-CR Connection

RISC-V Debug Solution

Tim Newsome ‹tim@sifive.com›

- It works!
- Easy to add to new cores

Demo Setup:

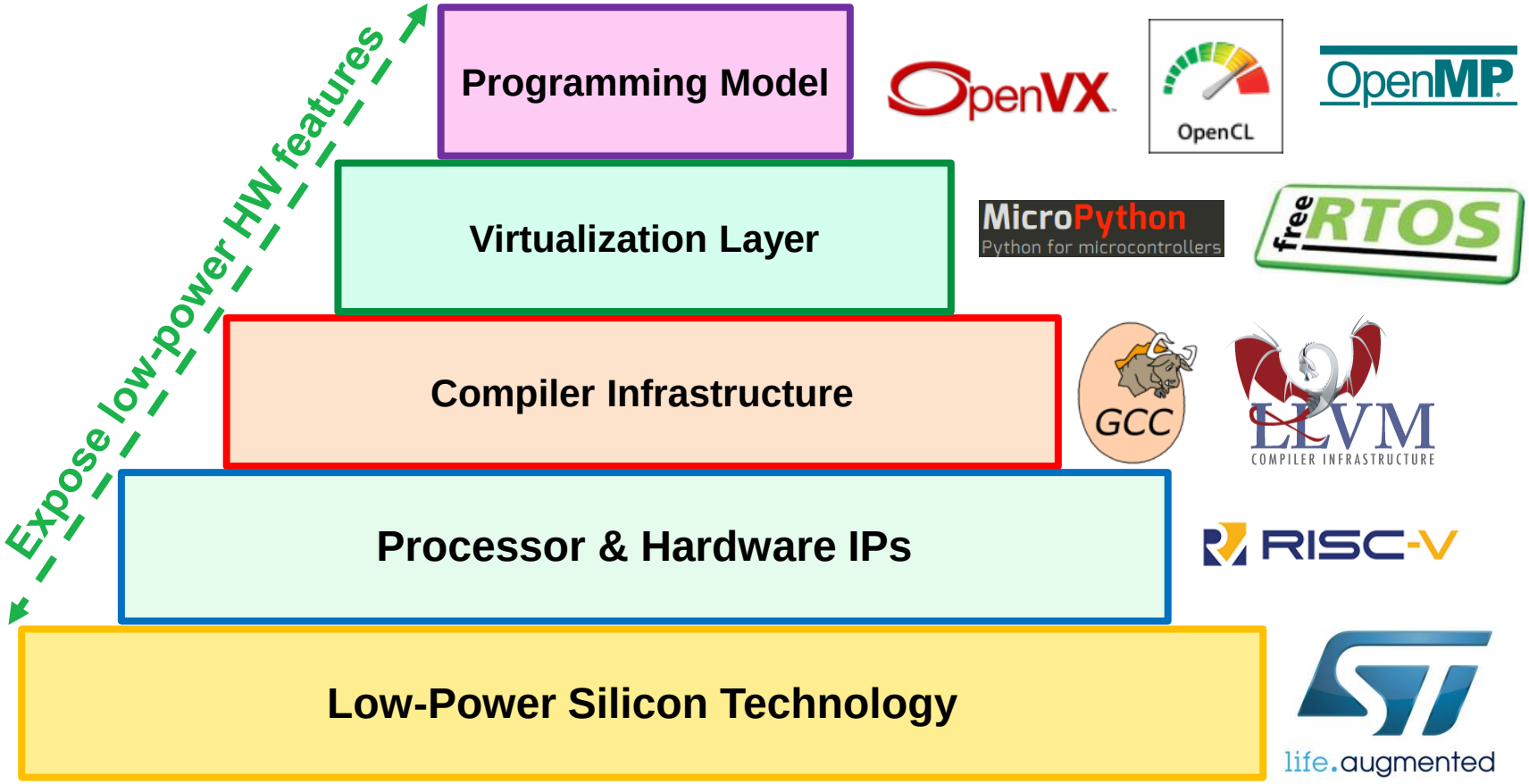


- Resource Usage
  - 3,900 gates of infrastructure
  - 600 gates per RV32 Rocket Core with 2 hwbp
- Open Source
  - Rocket Chip implementation
  - gdb and OpenOCD code
  - Black box testsuite
- More Information
  - Debug list at https://riscv.org/mailing-lists/

PULP: Parallel Ultra-Low-Power platform

Parallel + Programmable + Heterogeneous IoT computing platform

1 MOPS - 2000 MOPS @ 1mW-10mW active power

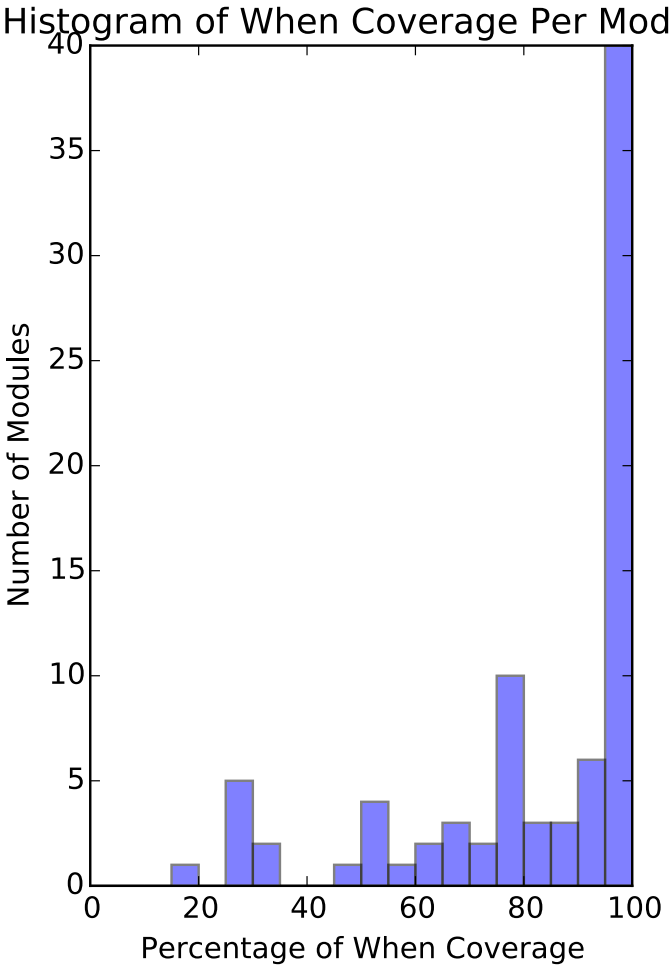


Automatic Path Coverage Tracking for RISC-V Processors

- Rocket-Chip Generator
  - Written in Chisel
  - Generates many differently-parameterized designs
  - Executes large suite of assembly tests
  - **What percentage of design is executed?**
- Path Coverage
  - Percentage of executed true/false paths in Chisel's if-else construct
  - "Sanity check"
  - 100% coverage != correct design
  - Generated Verilog loses information so existing coverage tools won't work
- Automatic Path Coverage Tracking
  - Instrumented Chisel compiler
  - Total percentage coverage, histogram of per-module coverage, scatter-plot of path execution frequency
  - List of unexecuted paths with source location information
- Case study: Torture
- Case study: Hwacha

Adam Izraelevitz

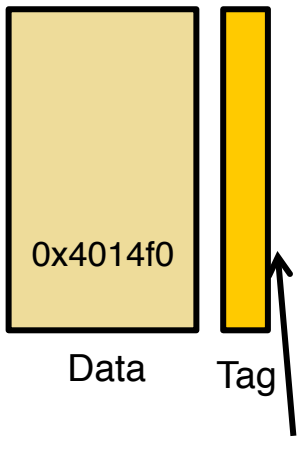
Colin Schmidt



Wildcat: RISC-V for Education
Martin Schoeberl

- A RISC-V simulation/implementation for teaching
- Instruction set simulator in about 200 LoC
- Code readability important
- Substitute MIPS in standard textbooks (CO&D, CAQA)
- Use at Technical University of Denmark
- Teaching material to share
  - Slides for the RISC-V instruction set
  - Text substitute for MIPS ISA in textbook
  - Assembler and C exercises and (simple) tool setup (or VM)
- Work-in-progress
- https://github.com/schoeberl/wildcat

P-Taxi: Simple 8-bit Meta-Data Processor

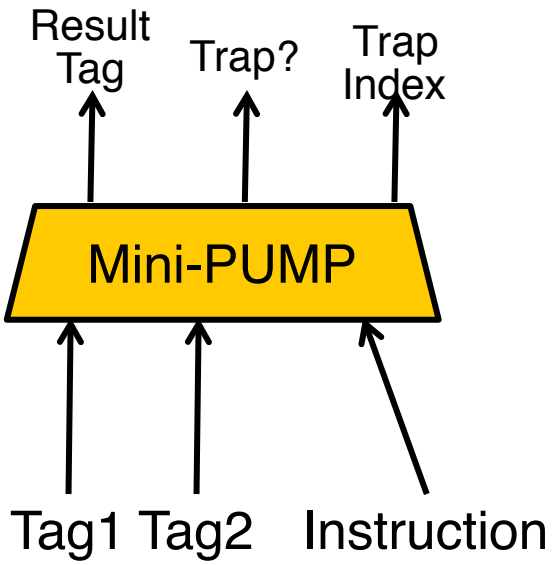


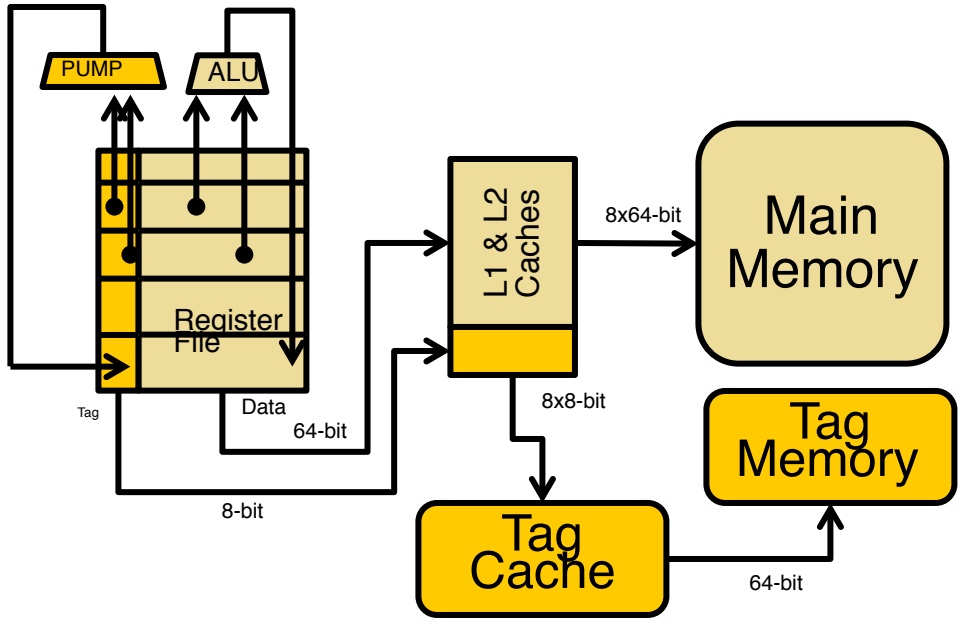
- **Pointer Types**
  - Code Pointer
  - Return Address
  - Data Pointer
  - Function Pointer
- **Data Types**
  - Instruction
  - Raw Data
  - Entry Point
  - Return Point

Type information is stored in protected metadata

- **Example: Return address**
  - Only "call" instruction can create return address
  - "Return" instruction will only respect a return pointer

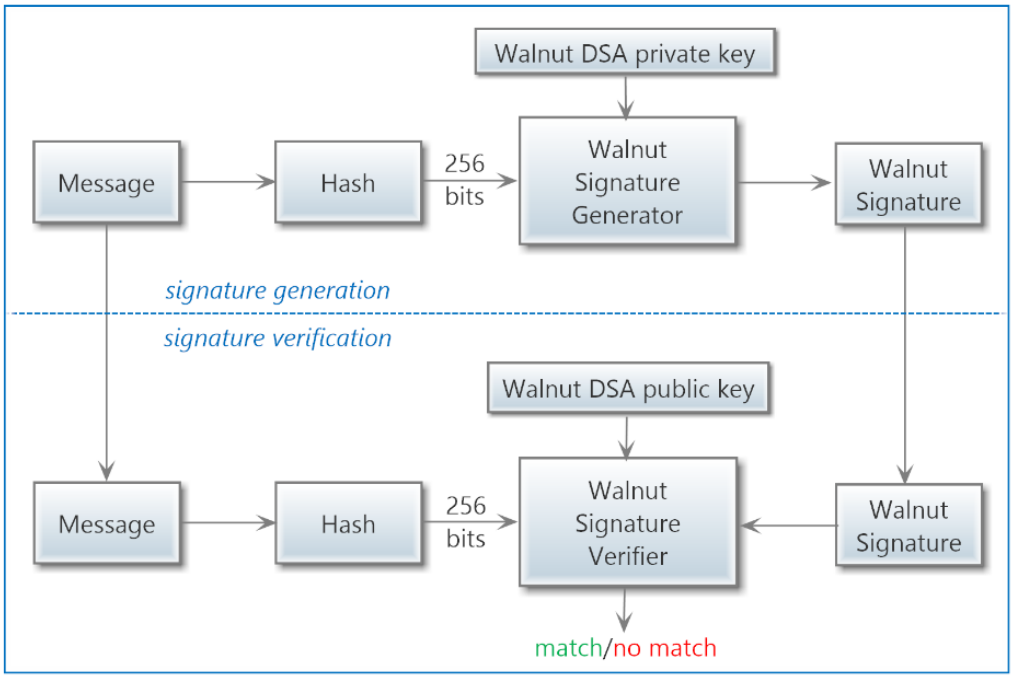
- The original Taxi changed the SPIKE definition of all relevant instructions
- P-Taxi uses a small PUMP to allow programmable policy definitions





High performance Digital Signature Algorithm called "Walnut DSA™" running on Codasip RISC-V compliant Codix-BK

- Group Theoretic Cryptography (GTC)-based signature verification
- Runs on Codix-BK, a Z-scale class 3-stage design, targeted to a Microsemi SmartFusion2 SoC FPGA
- Codix-BK custom instruction is employed to implement a quad Galois Field multiplier yielding a 3x improvement in run time



SECURE RF
Securing the Internet of Things®

codasip

Microsemi believes in RISC-V

- Microsemi is a major sponsor of the RISC-V tutorial
- We are committed to the creation of a thriving RISC-V ecosystem
  - Several 3rd party partners have run their RISC-V solutions on SmartFusion2 or Igloo2
  - Designs using Microsemi FPGAs with RISC-V cores are shipping
- Microsemi is defining a RISC-V roadmap
- We'd love to meet you!
  - Share your involvement with RISC-V
  - Share your ideas on how to evolve the RISC-V ecosystem
  - Learn about our FPGAs which are ideal for RISC-V cores & your design
  - Learn about the development board featured in the tutorial
- The first 100 visitors will receive a Microsemi laptop bag

f32c: a retargetable, FPGA-optimized core

- Executes either RV32 or MIPS ISA subsets
- Scalar, 5-stage pipeline
- 1.63 DMIPS/MHz, 3.06 CoreMark/MHz
  - Beats MicroBlaze, Nios, Cortex-M3 by 20% to 40%
- Performance optimized: < 1000 6-input LUTs
- Area optimized: < 650 6-input LUTs
- 115 MHz @ Lattice XP2 (90 nm, 4-input LUTs)
- 185 MHz @ Artix-7 (28 nm, 6-input LUTs)
- Direct-mapped caches, branch predictor...
- Hands-on demo with FPGArduino!

Edge Computing based on RISC-V Ecosystem

Lei Zhang, Randolph Wang, Zhiwei Xu, Ninghui Sun

Institute of Computing Technology, CAS

(1) RISC-V is open and free

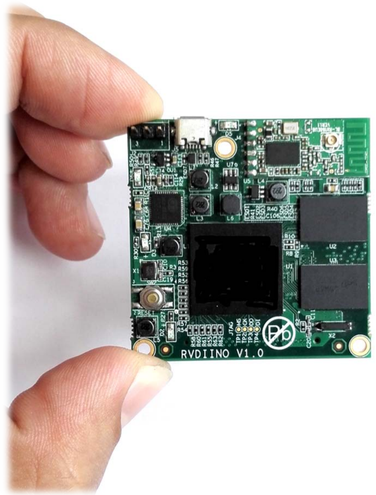
(2) more fragmented ecosystem

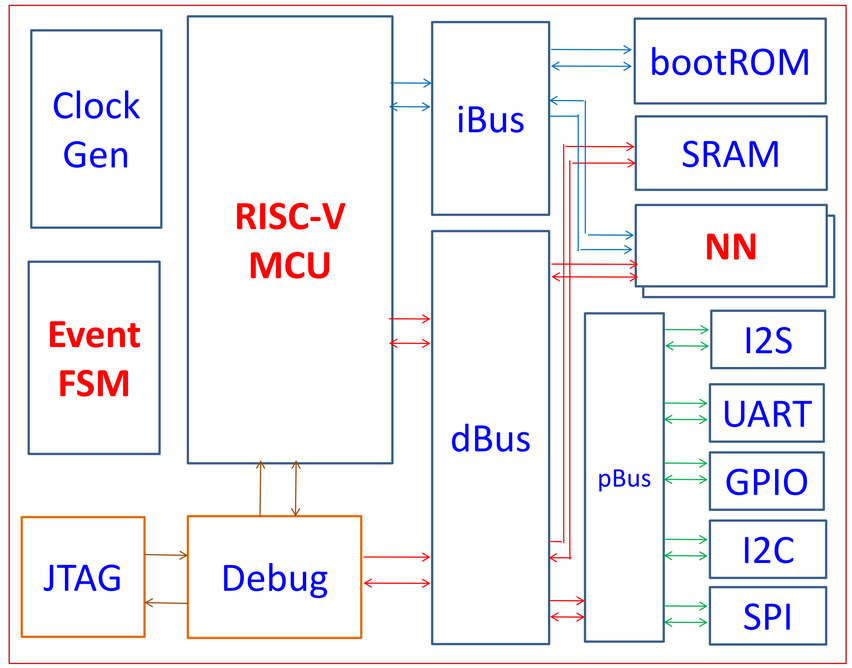
(3) no forward compatibility burdens

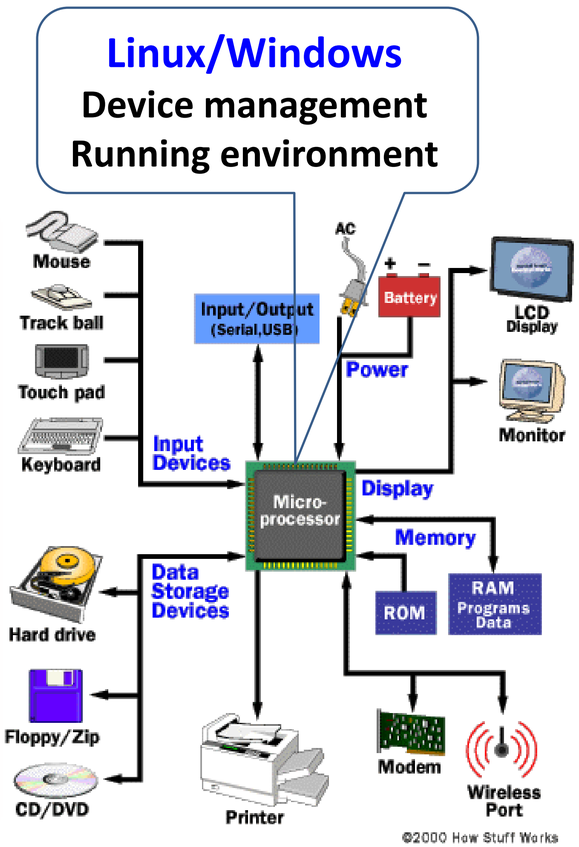
RISC-V MCU with Always-on Sensing

(1) 3-stage pipeline MCU with neuron network acc

(2) working on Arduino compatible - RVDuino







1. Device automatic access : plug-and-play

2. App Management : un/install, update

3. Runtime environment : virtual machine like abstraction

4. Device2Device interaction : buffer/pipe based D2D communication

5. REST service based device driver : API, SDK

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES