Open Source HPC: An Open Source Supercomputing Platform

Farzad Fatollahi-Fard David Donofrio John Shalf

{ffard, ddonofrio, jshalf}@lbl.gov

Lawrence Berkeley National Lab

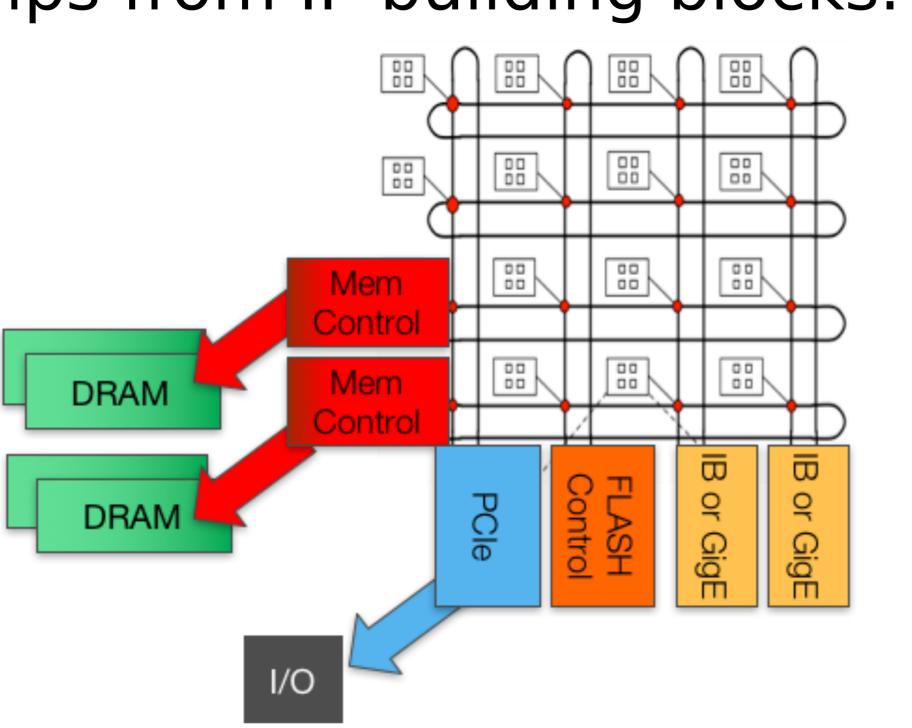
John Leidel Xi Wang Yong Chen

{john.leidel, xi.wang, yong.chen}@ttu.edu

Texas Tech University

Motivation

The idea of SoCs is starting to be applied to HPC, building chips from IP building blocks.



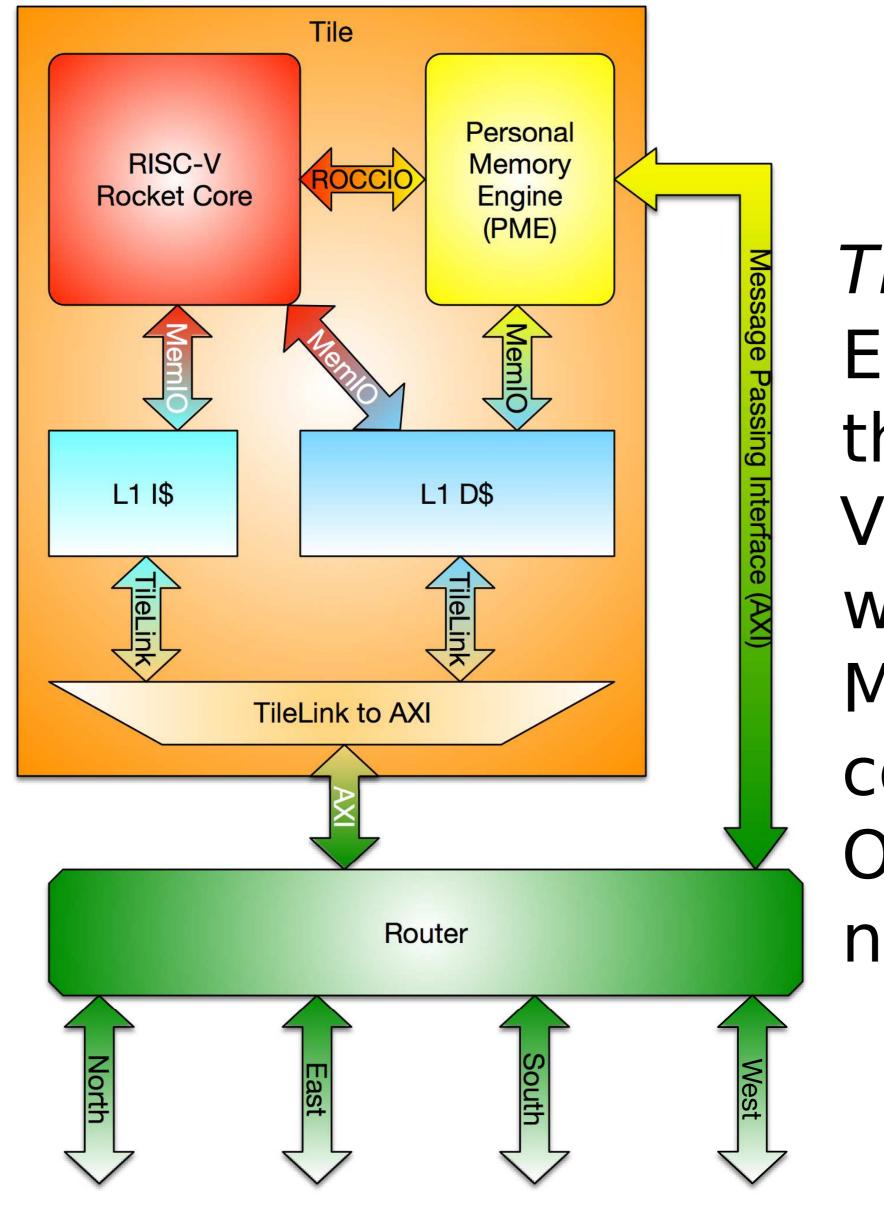
Commercial, closed-source IP is a major drag on innovation in all technology spaces. Open-source hardware has the potential to ignite multiple paths in the semiconductor industry: increasing diversity by driving costs down, lowering the barrier to entry, and opening the door for customization.



Goal

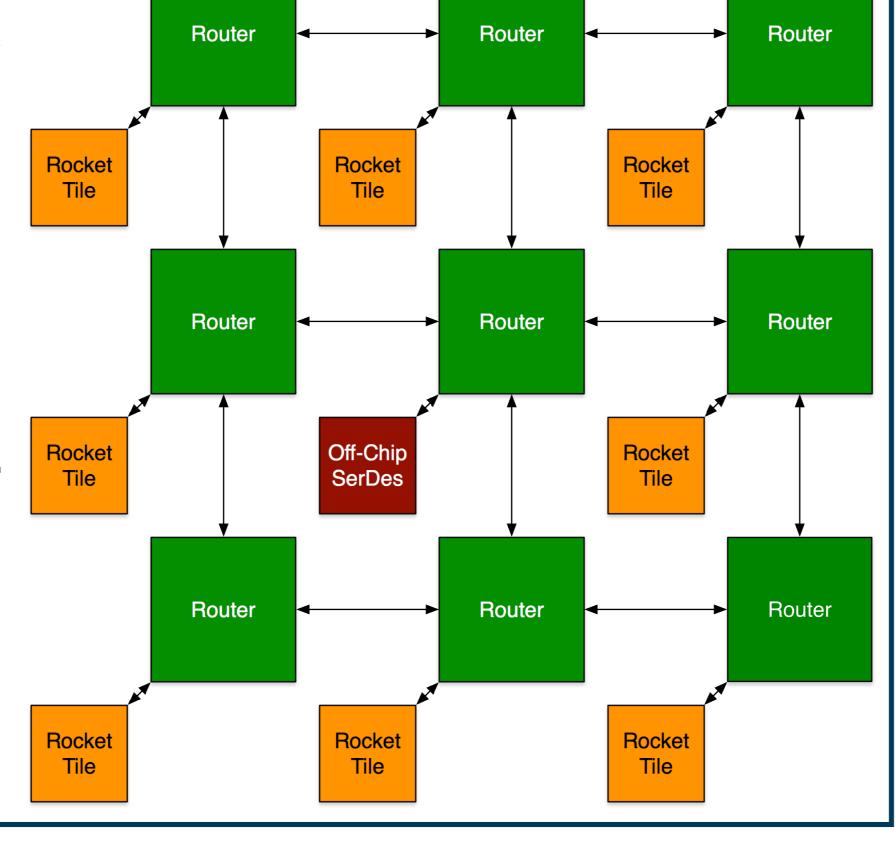
Using pre-verified open source IP, we want to create an open source silicon chip complete with a full software toolchain, including specialized instructions

Hardware Architecture

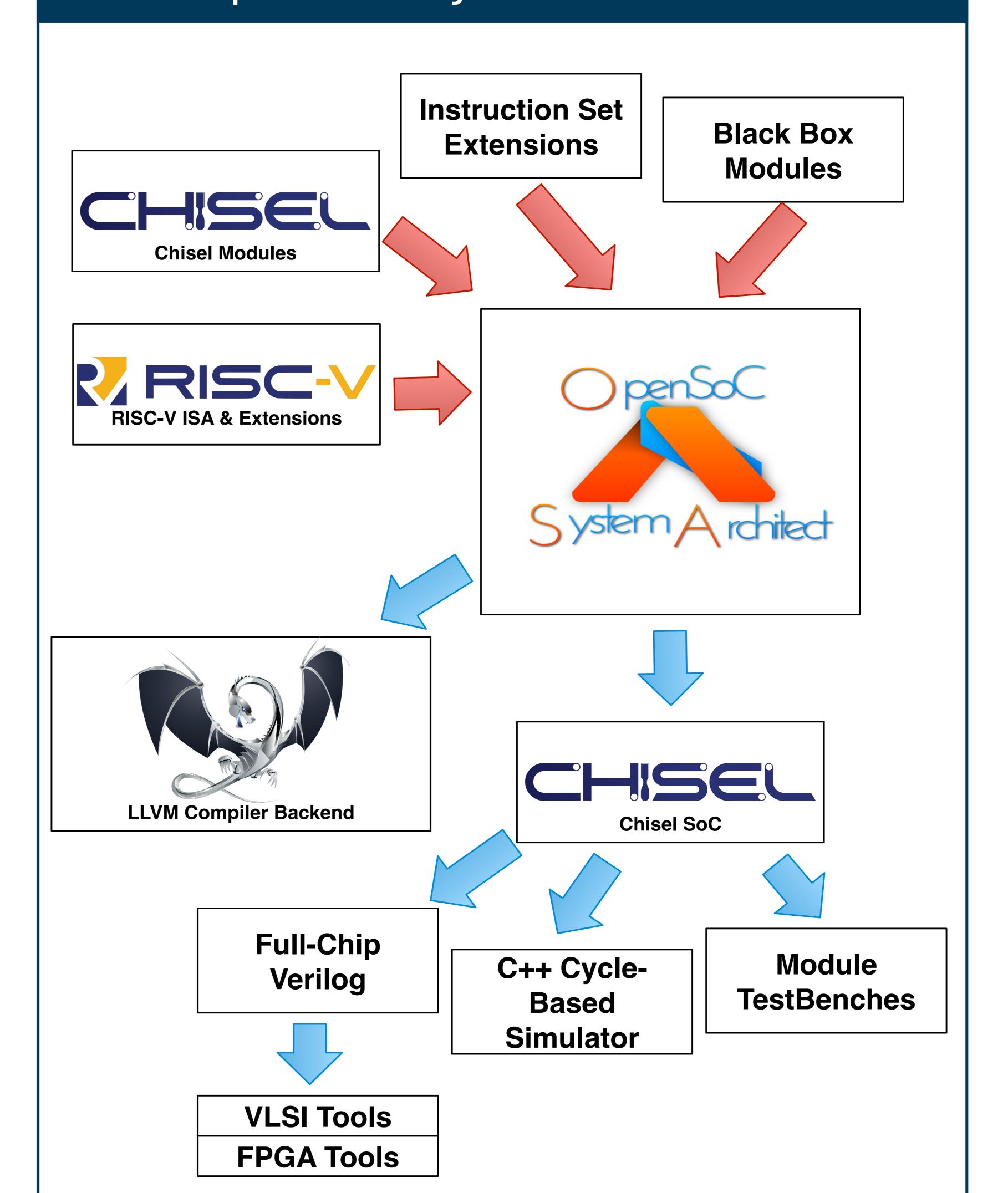


Tiled Architecture Each tile will contain the open source RISC-V based Rocket core with a custom Personal Memory Engine (PME), connected to an OpenSoC Fabric network.

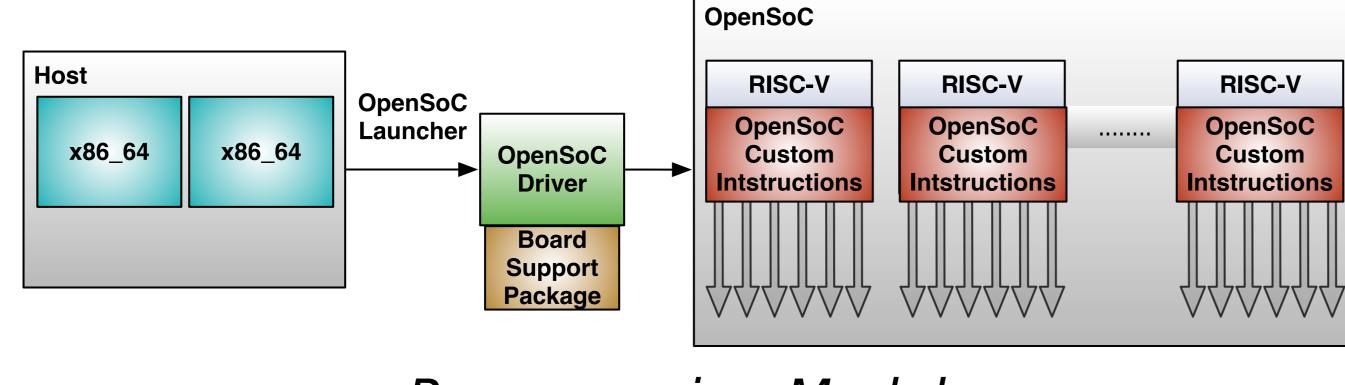
Configurable Network The tiles will be arranged in a mesh networking using the open source OpenSoC Fabric, with a node for a connection to main memory and off-chip communication.



OpenSoC System Architect



Software Infrastructure



Programming Models

The programming model relies on a MIMD approach where a single host (x86 64) application will launch multiple RISC-V threads. Each RISC-V thread may have additional, extended instruction support for application-specific computing workloads.

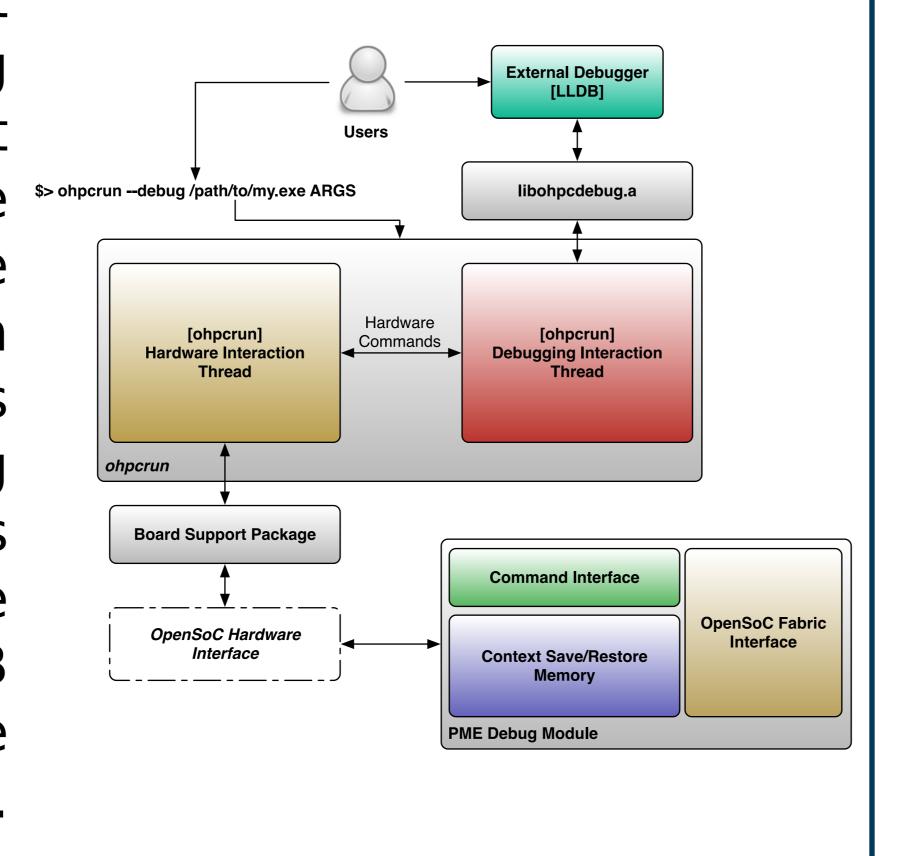
C, C++, ObjC Clang Frontend ► LLVM Optimizer LLVM Codegen Pragmas & **Library Calls** .text pensol System Architect

Compiler & Tool Chain

The compiler/tool chain will be based upon LLVM. The initial support will include C, C++ and, ObjectiveC source compilation using the RISC-V LLVM target. In addition to basic RISC-V support, the environment will also provide auto-generated instruction, register and inline intrinsic extensions for application-specific architecture support.

Execution Environment

The execution environment provides execution/debugging capabilities using an abstract set of interfaces. The \$> ohpcrun --debug /path/to/my.exe ARGS OHPCRUN tool provides the ability to initiate application workloads on the device as well as provide a messaging interface for client debuggers to interact with an active workload. The LLVM LLDB client debugger also will be utilized.



Further Reading & Acknowledgements

For further reading, please visit our website:

opensoc.community

We'd like to acknowledge the Laboratory for Physical Science and the Department





