Extending OpenPiton framework towards the HPC domain: first steps

Xabier Abancens, Mohsin Shahbaz, Guillem López, Noelia Oliete, Alireza Monemi, Neiel Leyva, Lluc Alvarez, Teresa Cervero, Miquel Moretó, John D. Davis
name.surname@bsc.es
Barcelona Supercomputing Center (BSC), Barcelona, Spain

Motivation

• HPC applications are demanding more specialized accelerators to tackle their increasing complexity and requirements
• Europe is promoting the use of the open source RISC-V ISA
• OpenPiton framework is not yet suited for the HPC domain: NoC, bus width, fixed caches' block sizes, single issue CPU (OpenSPARC)...

DVINO as a Tile in OpenPiton

OpenPiton[1]: Open source manycore research framework that comprises different tools and modules to build, test and implement RTL designs
DVINO[2]: Drac Vector IN-Order system composed by a RISC-V Scalar Core with a coupled accelerator. Developed at Barcelona Supercomputing Center (BSC)

In order to make OpenPiton suitable for the HPC domain, we include several modifications, listed below:

DVINO integration in OpenPiton:
• Provide straightforward compatibility with all in-house RISC-V cores:
  - In-order (currently operative)
  - Deeper pipeline in-order and out-of-order (under development)
• Adaptation to Ariane L1 cache subsystem
• New Performance Monitoring Unit to obtain metrics through CSRs

Scalar core's new features:
• Floating point operations (FD extension v2.2)
• Compress instruction extension (C extension v2.0)
• 6 pipeline stages

FPGA contributions

• Extending FPGAs support for the Xilinx FPGAs: Alveo U280 and U55C
• Supporting a custom FPGA Shell[3]: the framework is now compatible with a flexible, adaptable, configurable and open source FPGA Shell, which provides a seamless communication between the host and accelerators implemented in the FPGA

Conclusions and Future work

We exploit the open source OpenPiton framework to add new features and integrate our in-house CPU series aiming to make steps towards the HPC domain. These steps include modifying the following areas:

• Computational elements, the Tiles
• Verification environment
• FPGA implementation

The main upcoming work will be focused on:
• Including parametric L1 data cache blocks
• Increasing NoC size
• Adding custom L1 data and instruction cache subsystem
• Replacing in-order core by out-of-order core

References