Eithne: A framework for benchmarking micro-core accelerators

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
EITHNE: A FRAMEWORK FOR BENCHMARKING MICRO-CORE ACCELERATORS

What are micro-cores?
- Micro-cores have small instructions sets and tiny amounts (c. 32kB) of on-chip memory, resulting in very low power consumption.
- The low power consumption and simplicity of core mean that a large number can be placed on a chip.
- The HPC community is more and more interested in micro-core architectures as they provide massive parallelism on-chip. For example, the RISC-V based European Processor Initiative (EPI), and a combination of micro-cores with normal technology for “posits”.
- There are lots of hard-processor examples including the Adapteva Epiphany and the PEZY-SC2 (Shoubu system B).
- There is also an increasing number of soft-core examples, such as the GRVI Phalanx.

Which one to choose?
There are a number of factors to consider for the selection of a soft-core for use in a micro-core accelerator:
- Performance
- Power consumption
- Chip area - complex instruction sets require large decoding logic
- Scalability - maximum clock frequency can be limited by the complexity of core design
- Code density - on-chip RAM limited

Lots and lots of choice...

Objectives
- Provide a framework to support benchmarking of multiple hard and soft micro-core accelerators from a single codebase
- Measurement:
  - FLOPs
  - Power consumption (Watts)
  - Code size
  - Support multiple benchmarks

Sample Results
For expediency, we highlight a small sample of results here:
- Performance and power consumption for a single core of a subset of the supported devices
- Eithne supports multi-core devices and metrics can be created for a group / cluster of cores
- The results highlight the benefits of a framework that supports a number of different micro-cores and communication links

Framework overview
The framework provides a set API functions that enable a “plug-in” architecture to support multiple benchmarks and devices

Host
Benchmark
Control
Communications
Memory Management

Device type #1
Core
Core
Core
Core

Device type #2
Core
Core
Core
Core

Device kernel code
void kernel_init Khalin(tARGETFID id, KhalinSharedMem buffer) {
  KhalinKernel kernels[2];
  kernels[0] = sgesl;
  kernels[1] = sgefa;
  EITHNE_REGISTER_SCALAR (vars, RESULT, EITHNE_INTEGER, info);
  EITHNE_REGISTER_ARRAY (vars, A, EITHNE_FLOAT_ARRAY, a, N*LDA);
  EITHNE_REGISTER_ARRAY (vars, IPVT, EITHNE_INTEGER_ARRAY, ipvt, N);
  EITHNE_REGISTER_ARRAY (vars, JOB, EITHNE_INTEGER, job);
}

Communications / Control code
EITHNE_SEND (vars, TARGET_ID, A);
EITHNE_RECV (vars, TARGET_ID, A);

Sample Results graph

EITHNE: A FRAMEWORK FOR BENCHMARKING MICRO-CORE ACCELERATORS

The definition of “device” is flexible: most often this is a micro-core accelerator but it could be a thread running on the host.

Host benchmark code
buffer = EITHNE_ALLOC MEM (FILE) + "WFILE"
EITHNE_INIT_HOST (vars, HOST_ID, buffer + EITHNE_DATA_OFFSET, buffer);
EITHNE_INIT_CKORS (4);
EITHNE_RECV (vars, TARGET_ID, A);

Device kernel code
void kernel_init Khalin(tARGETFID id, KhalinSharedMem buffer) {
  KhalinKernel kernels[2];
  kernels[0] = sgesl;
  kernels[1] = sgefa;
  EITHNE_REGISTER_SCALAR (vars, RESULT, EITHNE_INTEGER, info);
  EITHNE_REGISTER_ARRAY (vars, A, EITHNE_FLOAT_ARRAY, a, N*LDA);
  EITHNE_REGISTER_ARRAY (vars, IPVT, EITHNE_INTEGER_ARRAY, ipvt, N);
  EITHNE_REGISTER_ARRAY (vars, JOB, EITHNE_INTEGER, job);
}

Further work
- Implement additional benchmarks
- Benchmark additional RISC-V soft-cores e.g. RISC-V, SwertV
- Kernels implemented using OpenMP
- MPI-based communications

Currently supported devices
- RISC-V
  - PicorV32 (soft-core)
  - VectorBlox Orca (soft-core)
  - RISCY (NXP NV32/M1)
- Xilinx MicroBlaze (soft-core)
- ARM
  - Cortex-M1 (soft-core)
  - Cortex-A9
- Adapteva Epiphany III
- Intel x86-64

Available on GitLab:
https://gitlab.com/mjamieson/eithne