Christof Teuscher
ECE 410/510: Hardware for AI and ML

Codefest #3: Decide the SW/HW boundary, pick tools, code toy example

Portland State University
Department of Electrical and Computer Engineering (ECE)
www.teuscher-lab.com
teuscher@pdx.edu

Next week

Monday
- CUDA
- Mapping deep neural nets (CNN) on GPUs
- TPUs

Wednesday
- No in-person codefest. Individual project time instead.
- Work from home or get together somewhere in small working groups.

Week 3 challenges

Challenge #10: Identify computational bottlenecks

1. Ask your favorite LLM to identity "computational bottlenecks" in the FrozenLake code from https://github.com/ronanmmurphy/Q-Learning-Algorithm
2. Do the suggestions make sense? How well is it able to identity bottlenecks?
3. Ask it to propose a HW implementation of the biggest bottleneck.
4. Ask it to generate System Verilog code for the HW implementation.

Challenge #11: GPU acceleration

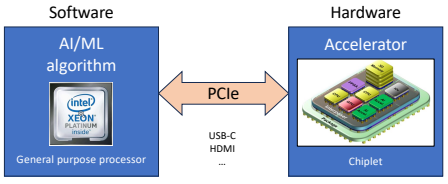
5. Ask your favorite LLM to optimize the FrozenLake code from https://github.com/ronanmmurphy/Q-Learning-Algorithm for a GPU.
6. Benchmark both the pure Python and the GPU-accelerated versions and compare. How much speed-up do you get?

Project page

Link on Canvas:

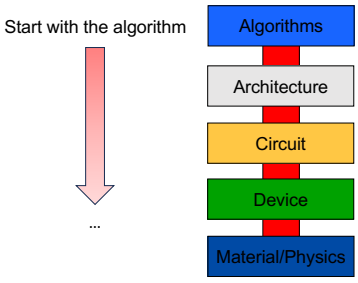
https://docs.google.com/document/d/1_HSDXhJEF1F2Qu76zJOlygvPm_F9NqX5VZ0f8CGXI34

Project vision



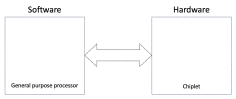
Design, test, and benchmark a co-processor chiplet that accelerates parts of some AI/ML code. Start with a blank slate for your design.
[Alternative]: design a stand-alone chiplet.

Compute stack



Main goals for today

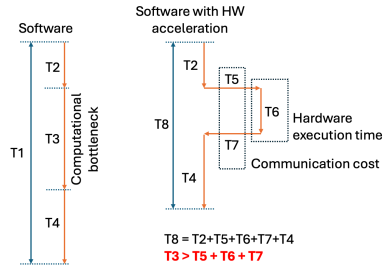
Pick a HW/SW boundary



Pick a workflow/toolchain

- PyMTL (Mamba)
- PyRTL
- Cocotb
- MyHDL
- pyUVM
- yosys
- …
- Candence
- Synopsis

Back-of-the-envelope estimate



T8 = T2+T5+T6+T7+T4

T3 > T5 + T6 + T7

Workflow/toolchain (1)

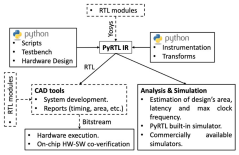
PyMTL (Mamba):

- https://pymtl.github.io
- User group: https://groups.google.com/g/pymtl-users
- An open-source hardware modeling, generation, simulation, and verification framework.
- MyHDL allows a subset of Python code to be translated to Verilog or VHDL. It offers co-simulation options where native Python code runs alongside a compiled simulation model of your hardware.
- Hardware-software co-simulation using PyMTL3:
  - Create your hardware model in PyMTL3
  - Develop software that will interact with the hardware
  - Set up the co-simulation environment
  - Run and analyze results

Workflow/toolchain (2)

PyRTL:

- https://ucsbarchlab.github.io/PyRTL
- PyRTL is an open-source Python-based hardware development toolkit.
- PyRTL provides a minimal set of hardware primitives, expressed as a Python class, which can then be extended with other classes and libraries as appropriate.
- Allows for fast design iteration in a variety of domains, including cryptography and machine learning.
- Paper:
  - Agile Hardware Development and Instrumentation with PyRTL
  - https://doi.org/10.1109/MM.2020.2997704



Workflow/toolchain (3)

Cocotb:

- https://www.cocotb.org
- cocotb is an open source coroutine-based co-simulation testbench environment for verifying VHDL and SystemVerilog RTL using Python.

MyHDL:

- https://www.myhdl.org
- MyHDL turns Python into a hardware description and verification language, providing hardware engineers with the power of the Python ecosystem.
- MyHDL designs can be converted to Verilog or VHDL.

pyUVM:

- https://github.com/pyuvm/pyuvm
- pyuvm is the Universal Verification Methodology (UVM) implemented in Python instead of SystemVerilog. pyuvm uses cocotb to interact with simulators and schedule simulation events.

yosys:

- https://github.com/YosysHQ/yosys
- Yosys is a framework for RTL synthesis and more. It currently has extensive Verilog-2005 support and provides a basic set of synthesis algorithms for various application domains.

Iterative, rapid prototyping codesign process

https://www.pacific-research.com/iterative-product-development

Communication interfaces

- **Advanced Interface Bus** (AIB): Developed by Intel, AIB is an open-source, die-to-die interconnect technology that enables high-bandwidth, low-power communication between chiplets. It supports both parallel and serial communication modes.
- **Universal Chiplet Interconnect Express** (UCIe): An industry-standard specification backed by major companies like AMD, Intel, Arm, and TSMC. UCIe aims to establish an open ecosystem for chiplet integration with standardized physical interfaces, protocols, and power management.
- **High Bandwidth Memory (HBM) Interface**: While primarily for connecting memory chiplets, this interface uses through-silicon vias (TSVs) and microbumps to achieve very high bandwidth connections.
- **Bunch of Wires (BoW):** A simpler, more cost-effective interface for medium-bandwidth connections between chiplets, developed through the Open Compute Project.
- TSMC's **Integrated Fan-Out (InFO)** and **Chip-on-Wafer-on-Substrate (CoWoS):** These are packaging technologies that include specific interconnect interfaces for connecting multiple chiplets.

Unclear of IP is available for any of these…