Efficiently accelerating AI workloads with RISC-V

Guillem Solé

Software Lead

Esperanto Technologies

Email: guillem.sole@esperantotech.com

About Esperanto Technologies Inc.

- Esperanto is a startup company based in Mountain View, California with an office in Barcelona

- Esperanto is building one of the world's most advanced chips to accelerate Machine Learning
  - Designed with the world leading 7nm TMSC silicon technology
  - Goal is to have industry leading Tera Operations per Watt for our target workloads
  - **Based on the RISC-V instruction set**

- Technical features
  - Over a thousand 64-bit RISC-V cores on one chip
  - Each RISC-V core has its own vector floating point accelerator
  - Network on chip allows all processors to reside in same address space
  - Multiple levels of cache
  - High bandwidth DRAM interfaces
  - Energy efficient design techniques

Why we chose the RISC-V instruction set as a base

- For a startup time to market and money are key to success
- Can’t spend a lot of effort and time working on non-key factors
- If your value is not in the definition of the base instruction set, don’t spend time on it!
  - Lots of time lost in taking decisions
    - VLIW?
    - RISC vs CISC
    - Variable vs fixed encoding
    - …
  - Result is likely going to be worse than already existing solutions
  - Take something that is already robustly designed and save time
- ARM is too expensive if you are not going to take advantage of its ecosystem
- RISC-V has an excellent base instruction set and is free

RISC-V Compilers were available from Day 0 of our design

- Developing and tuning a compiler is a lot of time
- Compiler is a key part of the whole solution for AI systems
- Having a compiler working on day 0 is key to correctly drive the chip design
  - Allows designers to know how the hot spots will look like sooner
- All the effort on the compiler can be focused on the specific features of the chip
- Bug fixes and improvements come for free along the way
- RISC-V has GCC and LLVM working

RISC-V met our instruction extension requirements

- Esperanto needed to have the ability to extend the base ISA with its own proprietary extensions

- ISA extension requirements
  - Can be easily extended
    - Extension mechanisms that require big decoders or complex front end changes are not desired
  - Don’t require to negotiate with other parts how to extend it to hit desired time to market
    - Getting to a common ground might take too much time
    - There are a lot of quarters from closing an ISA to having a product in market
  - Don’t enforce to disclose the extensions
    - Extensions might be the key component of a company

- RISC-V meets all the extension requirements

Why RISC-V enables efficient domain specific solutions

- For a solution to success, it needs to minimize overhead not related to the specific domain
- RISC-V base ISA is really small
  - Decoder area/power is negligible
- RISC-V memory ordering enables low weight solutions
  - Reduces area/power dedicated to track loads/stores
  - Reduce design and validation effort
- Not allowing Self Modified Code is a win
  - SMC prevents some optimizations or makes them too costly
  - At Intel many proposals died because of that
- Thanks to its simplicity, RISC-V enables designs where majority of area/power/design effort are in the pieces that add value to the company

RISC-V at Esperanto

- RISC-V enabled Esperanto to have a general purpose AI solution
  - Majority of other AI solutions use fixed function hardware with systolic-array-like architectures
  - Esperanto's vision is that flexibility is a must have for future AI workloads

- Power/Area overhead related to RISC-V support is small
  - Decoders, scalar ALU, …

- RISC-V simple design allows Esperanto to focus on energy efficient design techniques

- Esperanto adds its own proprietary extensions on top of RISC-V
  - Key feature to efficiently run AI workloads
  - Majority of the core power is spent on the ALUs and not in control

Want to learn more? We're hiring, come talk with us.