Open ESP
The Heterogeneous Open-Source Platform for Developing RISC-V Systems

Luca P. Carloni with Davide Giri

Columbia University
IN THE CITY OF NEW YORK

FOSDEM' 20,
Brussels Feb 1, 2020

Open Source Release of ESP

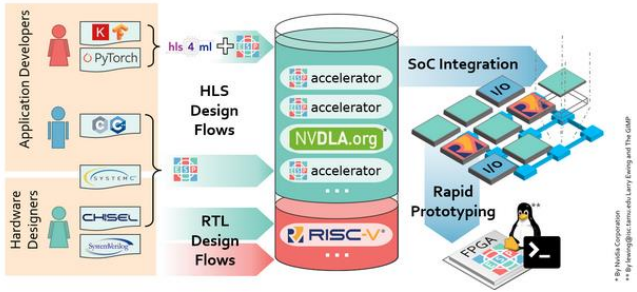
https://www.esp.cs.columbia.edu

Home Resources News Press Team Contact

ESP
the open-source SoC platform

The ESP Vision

ESP is an open-source research platform for heterogeneous system-on-chip design that combines a flexible tile-based architecture and a modular system-level design methodology.



ESP provides three accelerator flows: RTL, high-level synthesis (HLS), machine learning frameworks. All three design flows converge to the ESP automated SoC integration flow that generates the necessary hardware and software interfaces to rapidly enable full-system prototyping on FPGA.

Overview

Latest Posts

Video proceedings of the ESP talk at the RISC-V Summit

The video of the ESP talk at the RISC-V summit 2019 is now available on Youtube.

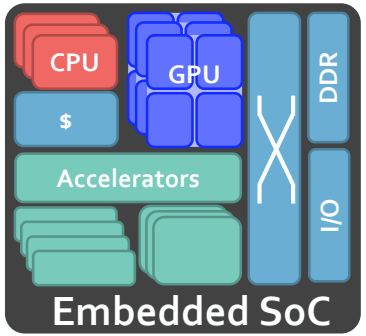
Read more

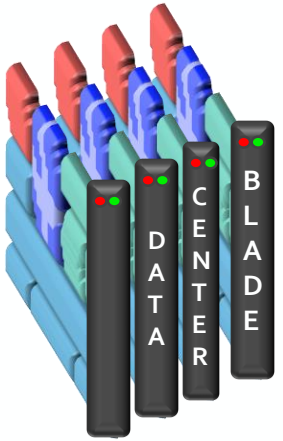
Published: Jan 25, 2020

Two accepted talks at FOSDEM 2020 in Brussels

We will give two talks in the RISC-V developer room at FOSDEM in Brussels (Belgium) on February 1st.

Why ESP?





Heterogeneous systems are pervasive

Integrating **accelerators** into a SoC is hard

Doing so in a **scalable** way is very hard

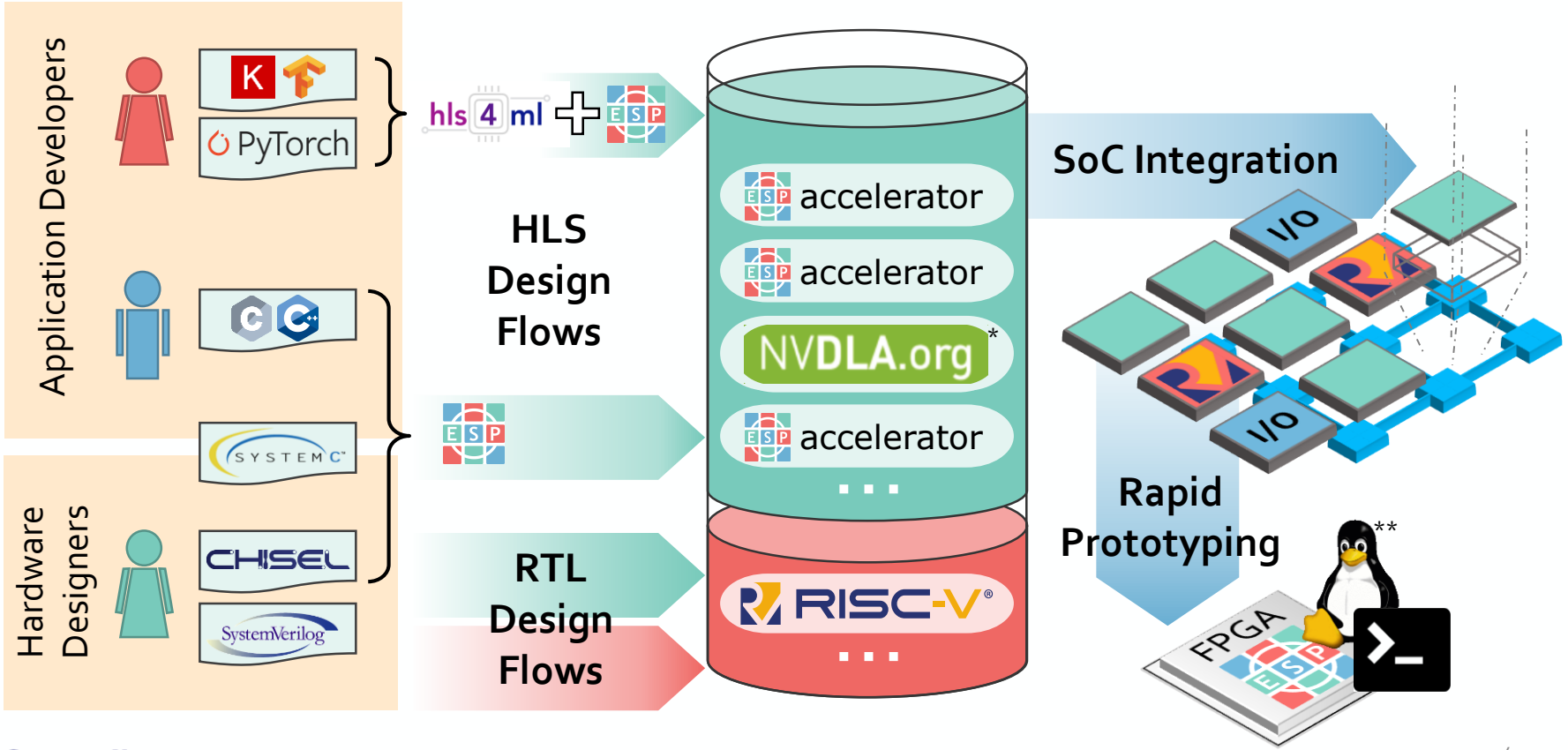
Keeping the system **simple to program** while doing so is even harder

ESP makes it easy

ESP combines a **scalable architecture** with a **flexible methodology**

ESP enables **several accelerator design flows**
and takes care of the hardware and software integration

ESP Vision: Domain Experts Can Design SoCs

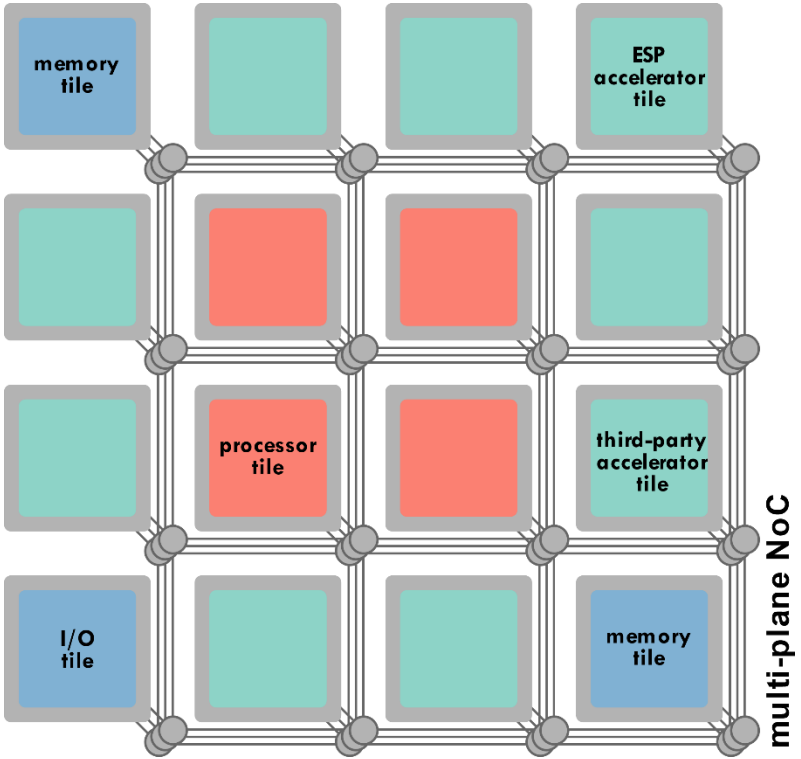


* By Nvidia Corporation
** By lewing@isc.tamu.edu Larry Ewing and The GIMP

ESP Architecture

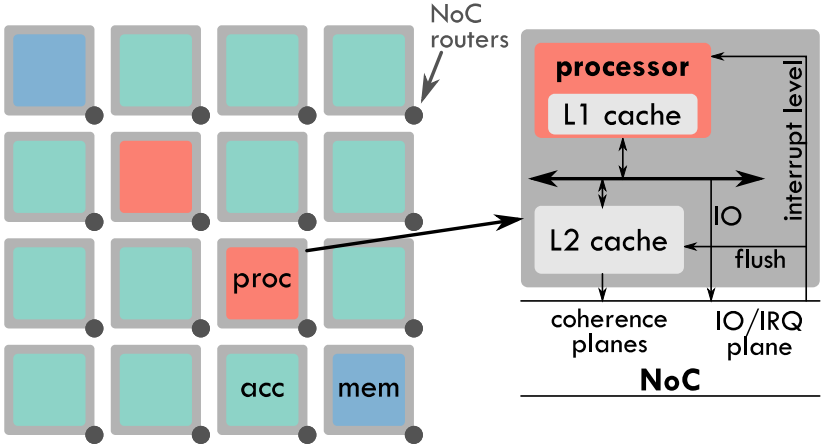
- RISC-V Processors
- Many-Accelerator
- Distributed Memory
- Multi-Plane NoC

The ESP architecture implements a **distributed** system, which is **scalable**, **modular** and **heterogeneous**, giving processors and accelerators similar weight in the SoC



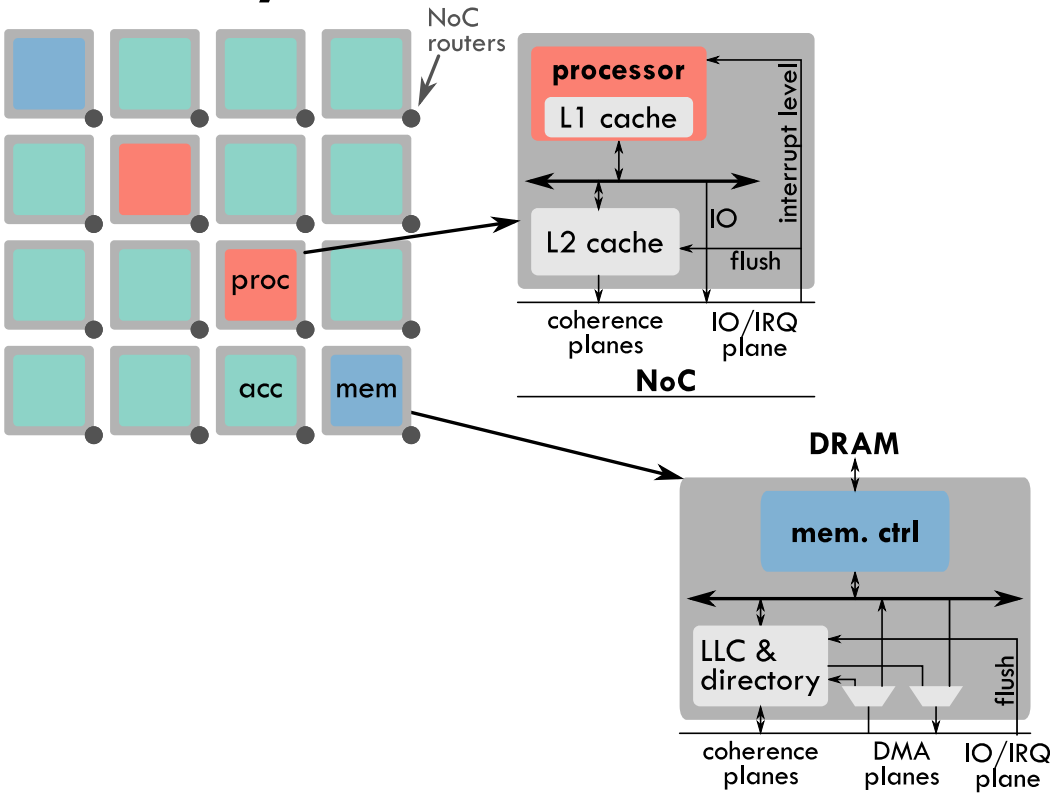
ESP Architecture: Processor Tile

- Processor off-the-shelf
  - RISC-V Ariane (64 bit)
    SPARC V8 Leon3 (32 bit)
  - L1 private cache
- L2 private cache
  - Configurable size
  - MESI protocol
- IO/IRQ channel
  - Un-cached
  - Accelerator config. registers, interrupts, flush, UART, …



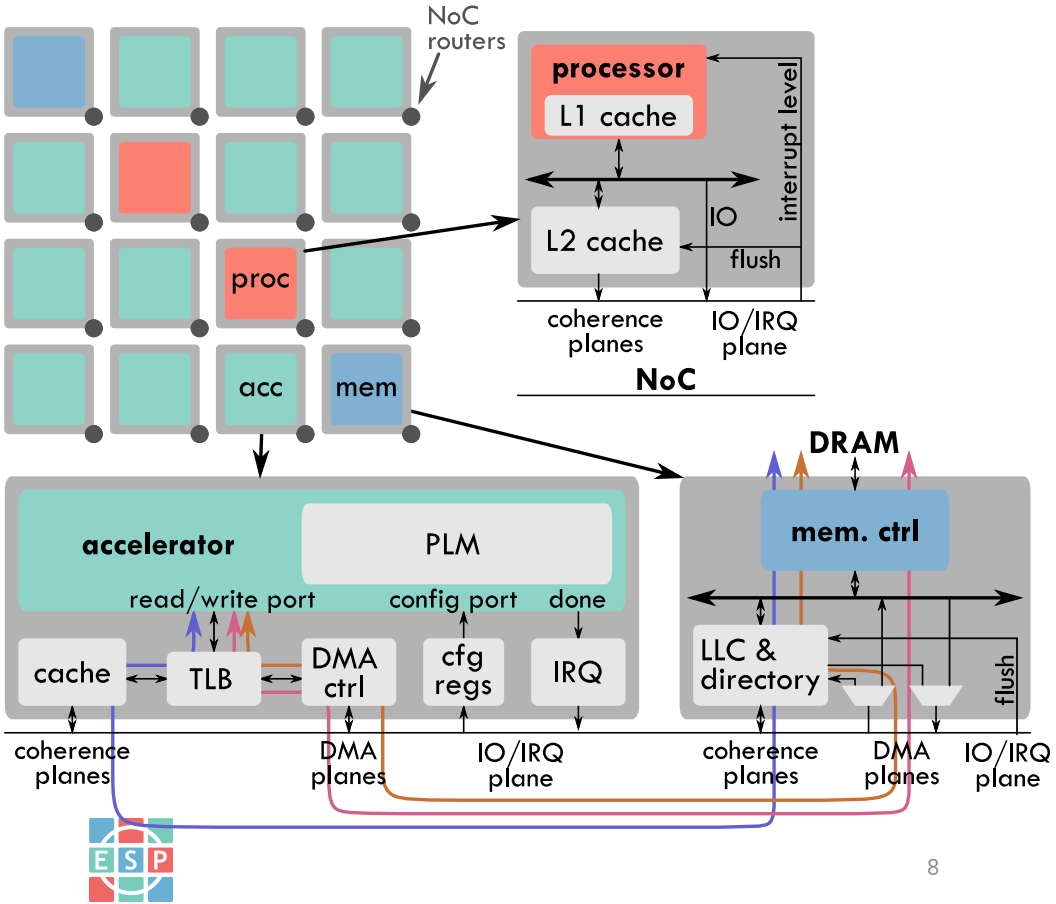
ESP Architecture: Memory Tile

- External Memory Channel
- LLC and directory partition
  - Configurable size
  - Extended MESI protocol
  - Supports coherent-DMA for accelerators
- DMA channels
- IO/IRQ channel

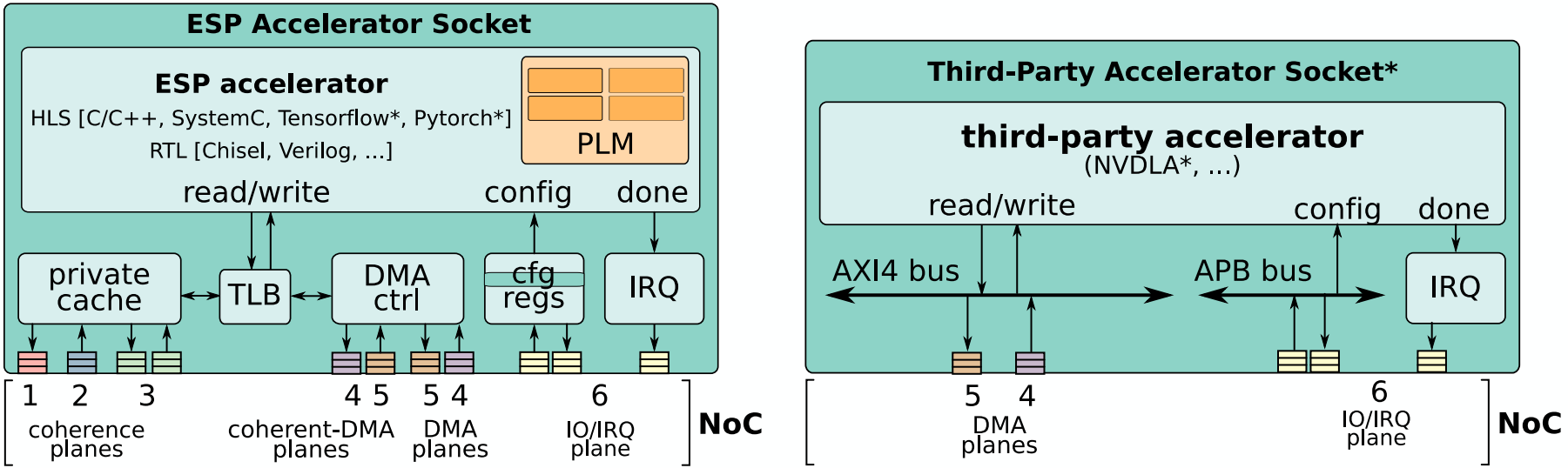


ESP Architecture: Accelerator Tile

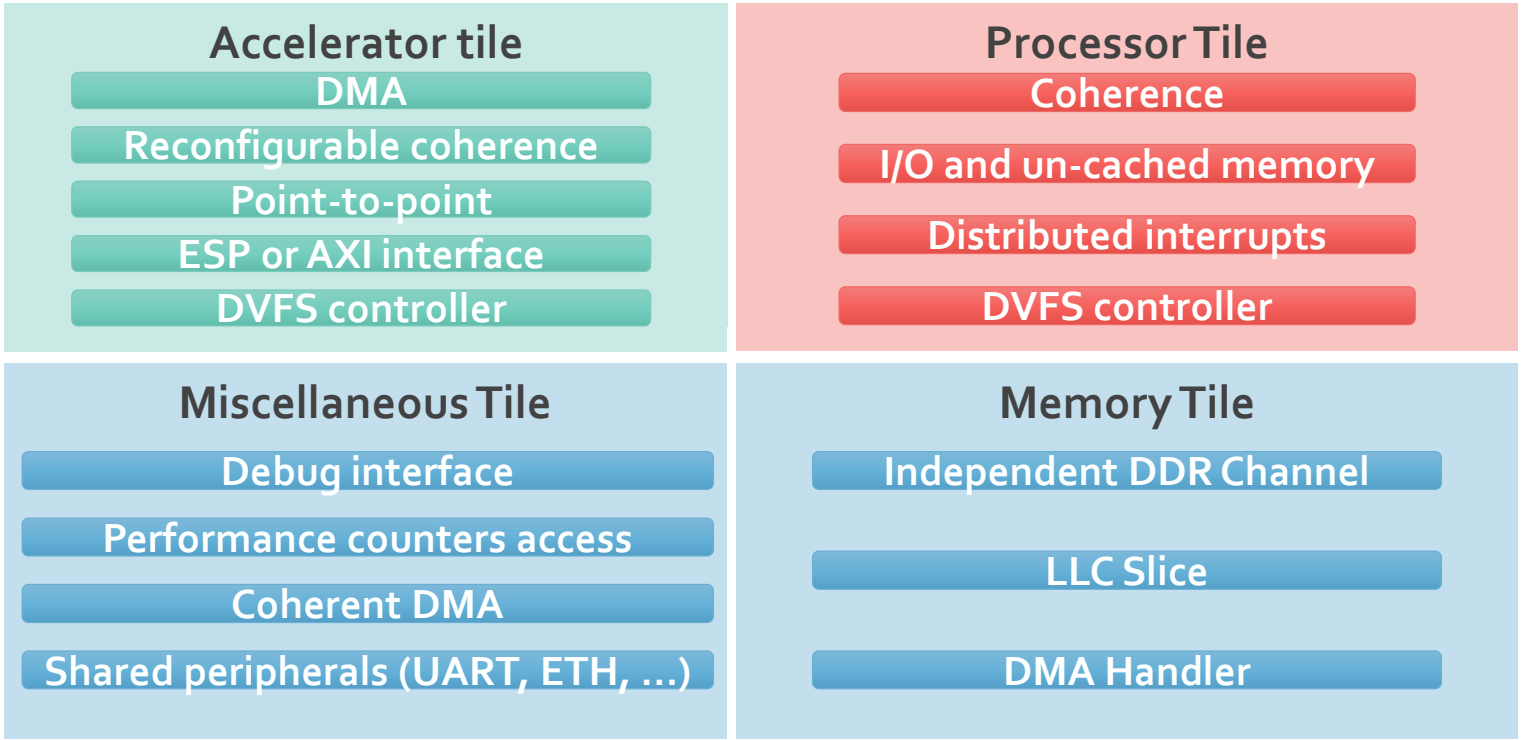
- Accelerator Socket w/ Platform Services
  - Direct-memory-access
  - Run-time selection of coherence model:
    - Fully coherent
    - LLC coherent
    - Non coherent
  - User-defined registers
  - Distributed interrupt



ESP Accelerator Socket

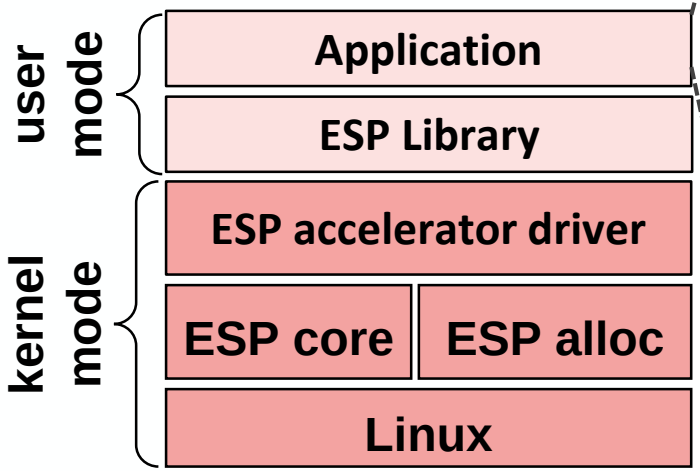


ESP Platform Services



ESP Software Socket

- ESP accelerator API
  - Generation of device driver and unit-test application
  - Seamless shared memory




```
/*
 * Example of existing C application
 * with ESP accelerators that replace
 * software kernels 2, 3 and 5
 */
{
  int *buffer = esp_alloc(size);

  for (...) {

    kernel_1(buffer,...); /* existing software  */

    esp_run(cfg_k2);      /* run accelerator(s) */
    esp_run(cfg_k3);

    kernel_4(buffer,...); /* existing software  */

    esp_run(cfg_k5);
  }

  validate(buffer);       /* existing checks    */

  esp_cleanup();          /* memory free        */
}
```

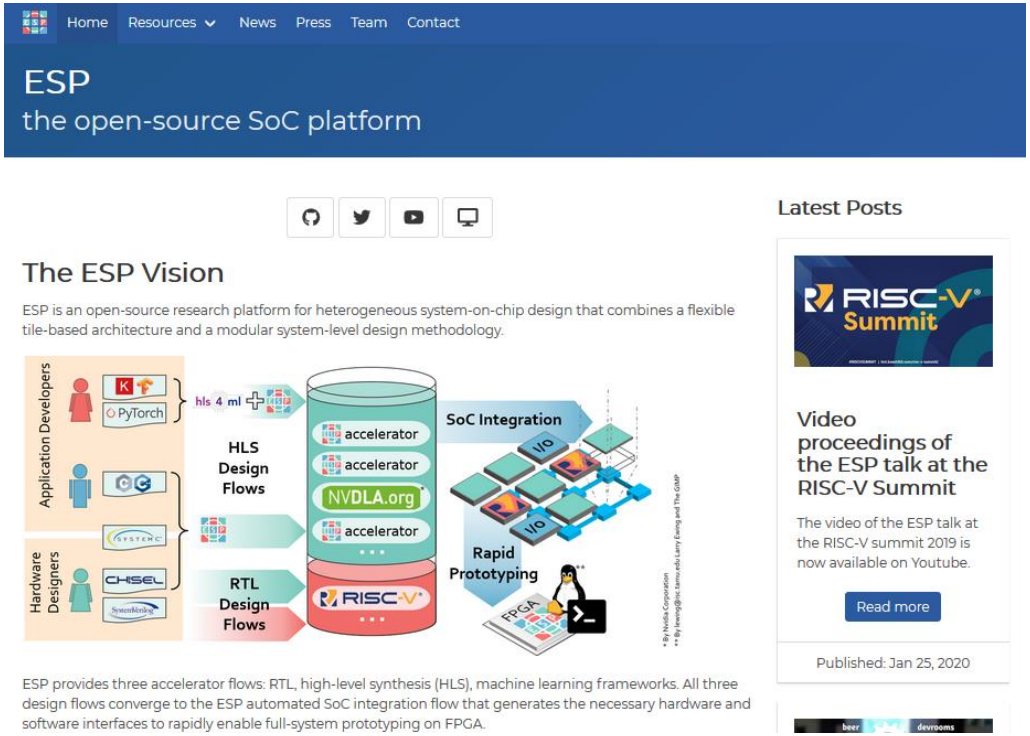
ESP
The open-source heterogeneous system-on-chip platform

SystemC and C/C++ Accelerator Design Flow

In Summary: ESP for Open-Source Hardware

https://www.esp.cs.columbia.edu

- We contribute **ESP** to the OSH community in order to support the realization of
  - **more scalable** architectures for SoCs that integrate
  - **more heterogeneous** components, thanks to a
  - **more flexible** design methodology, which accommodates different specification languages and design flows
- ESP was conceived as a heterogeneous integration platform from the start and tested through years of teaching at Columbia University
- We invite you to **use ESP** for your projects and to **contribute to ESP**!



Thank you from the ESP team!

https://esp.cs.columbia.edu

https://github.com/sld-columbia/esp

System Level Design Group

COMPUTER SCIENCE

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK