RISC-V® Summit

December 8-10 | Virtual Event

CORE-V MCU SoC

Florian Zaruba, OpenHW Group

florian@openhwgroup.org

CONTRIBUTE.
COLLABORATE.
COMMERCIALIZE.

OPENHW GROUP™

Brought to you by
informatech

riscvsummit.com #RISCVSUMMIT

CORE-V MCU SoC, Open Source, 22nm MCU with Embedded FPGA

Florian Zaruba florian@openhwgroup.org

T @be4web T @openhwgroup

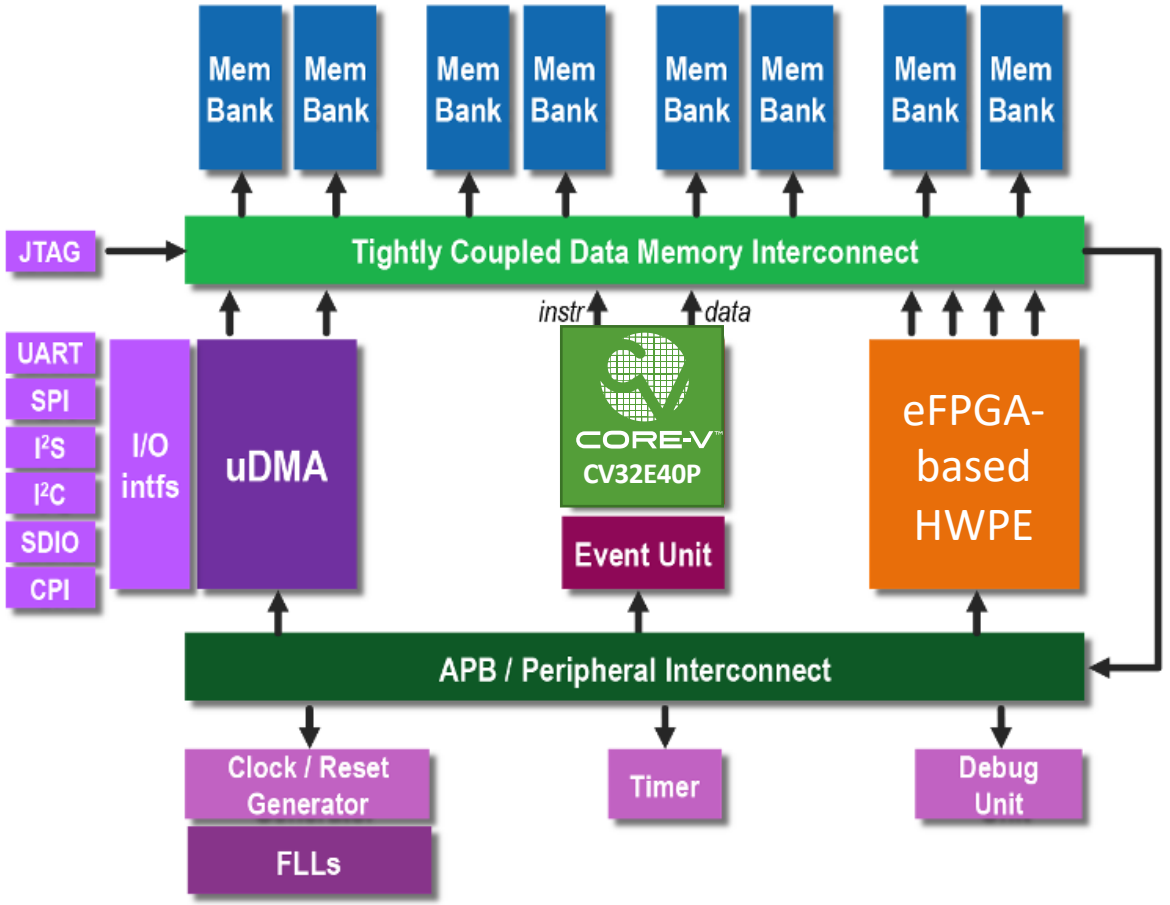
www.openhwgroup.org

History

- Both platforms originate from the PULP Project: Tuned for energy-efficiency
- CORE-V MCU is derived from PULPissimo
  - Efficient micro-controller
  - Improved CV32E40P
  - Standardized OBI protocol
  - APB for µDMA subsystem
- CORE-V APU is derived from Ariane's SoC
  - UNIX-capable system
  - Minimal infrastructure to demonstrate the core
  - AXI-based
  - Xilinx Peripherals (SPI, DW converter)

CORE-V MCU

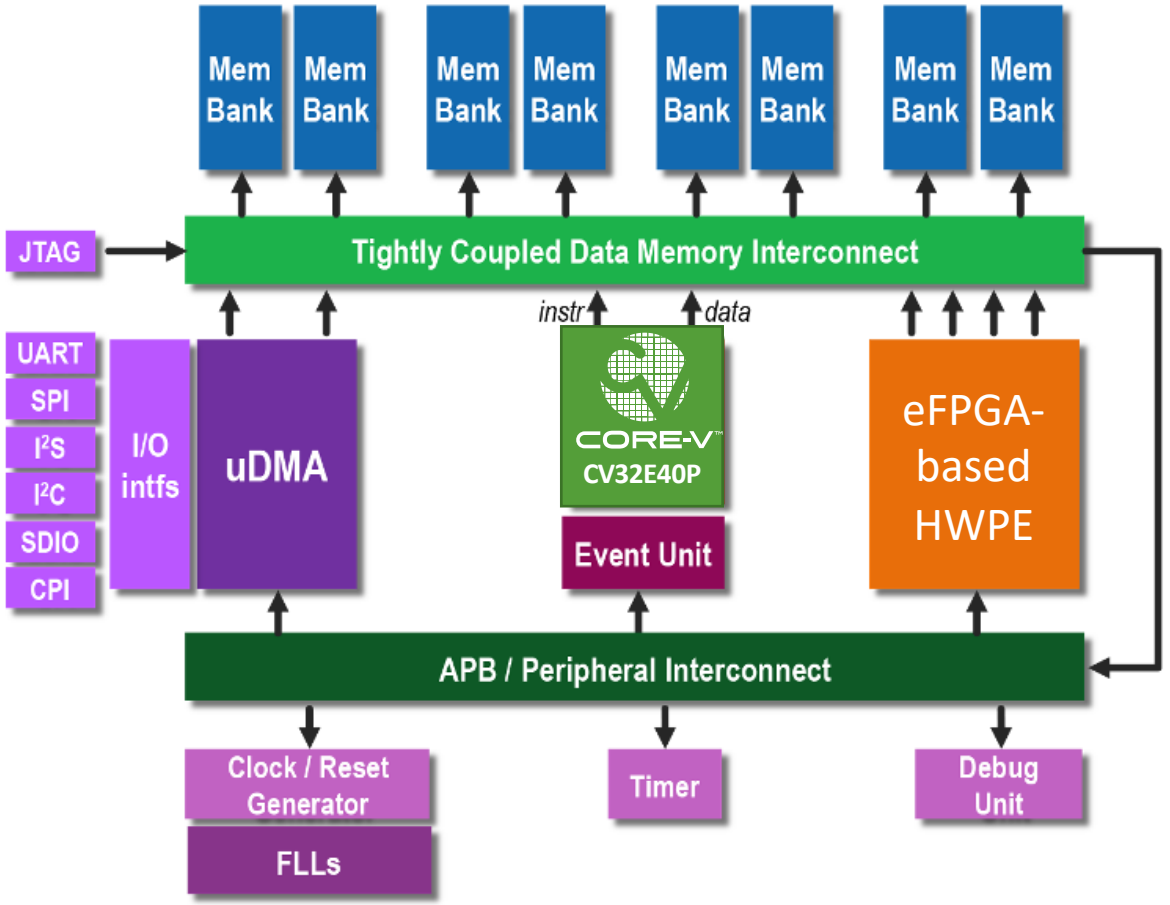
- Shared memory
  - Unified Data/Instruction Memory
- Support for Accelerators
  - i.e., HW Processing Engine (HWPE)
  - Direct shared memory access
  - Programmed through APB bus
- µDMA for I/O subsystem
  - Can copy data directly from I/O to memory without involving the core
- Used as controller in larger systems



https://github.com/openhwgroup/core-v-mcu

CORE-V MCU

- Rich set of peripherals for edge AI sensors (vision and time series):
  - QSPI (up to 280 Mbps)
  - Camera Interface
  - I2C/I2S
  - JTAG, GPIOs
  - Interrupt controller
  - boot ROM
- Custom accelerators for ML use cases



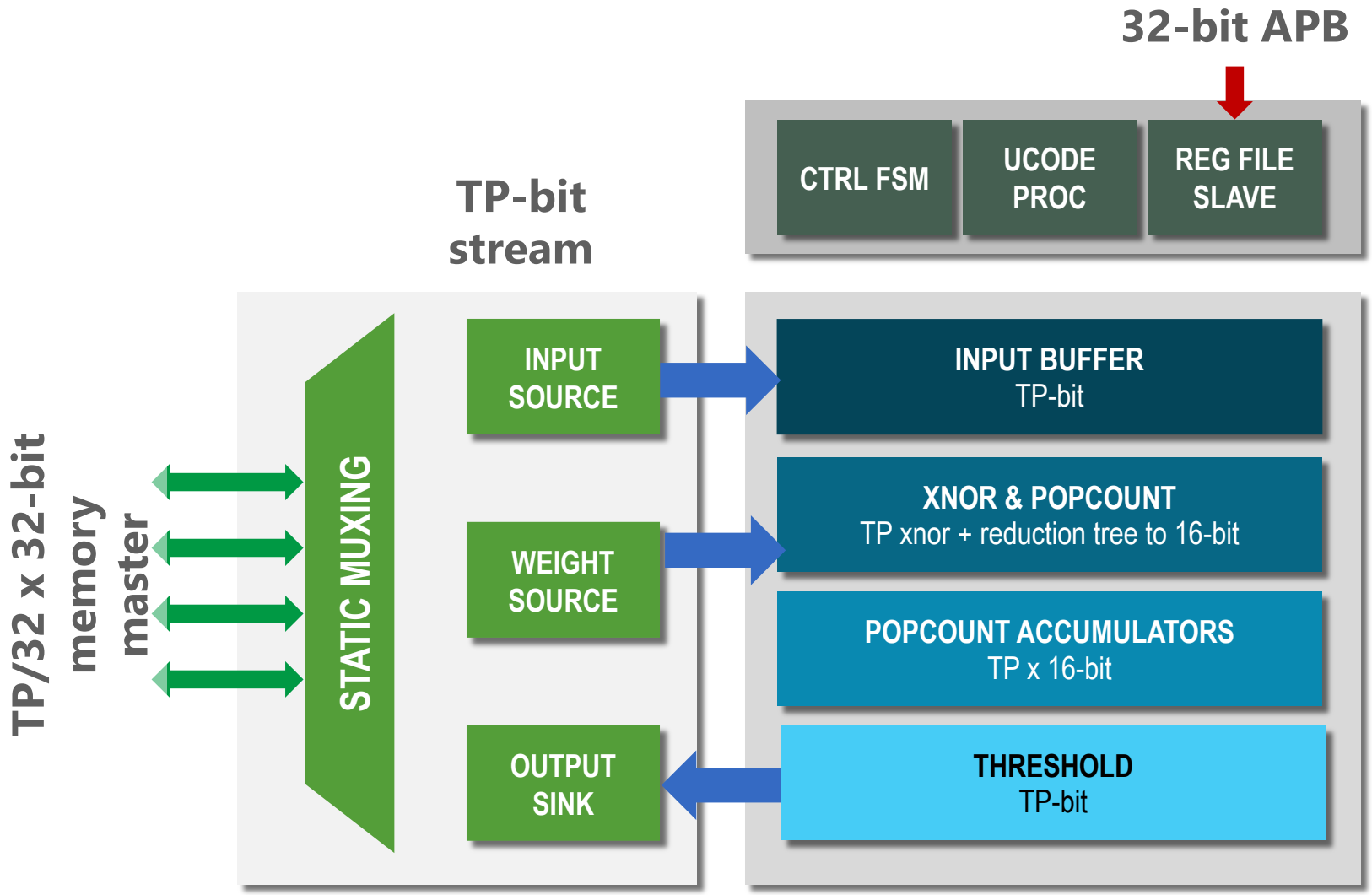
https://github.com/openhwgroup/core-v-mcu

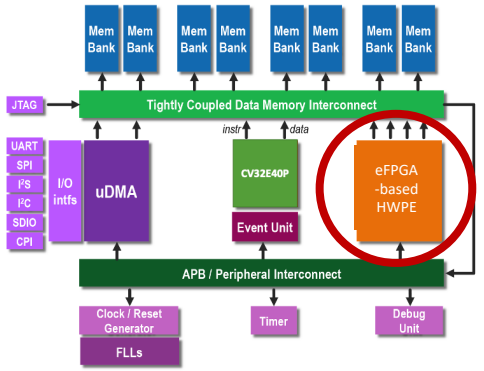
PULP Interrupt Controller

- Generates up to 32 requests
  - Events vs. interrupts
- Mapped on the APB bus
- Receives events in a FIFO from the SoC event generator
  - Unique interrupt ID
- Mask, pending, ack, event id registers
- Special set, clear, read, write operation registers

- Sources:
  - Timers
  - GPIO
  - Custom accelerator
  - Events from the uDMA
- CV32E40P
  - Support for standardized and “fast” RISC-V interrupt mechanisms

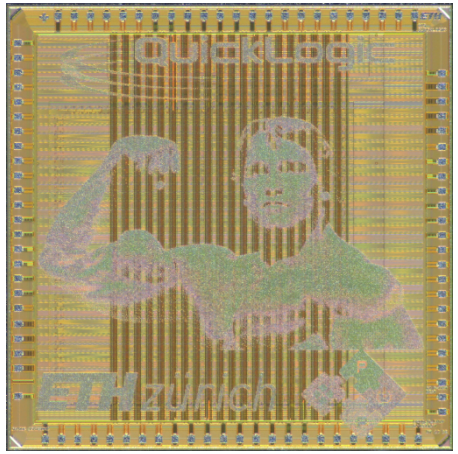
Example for a Custom Accelerator



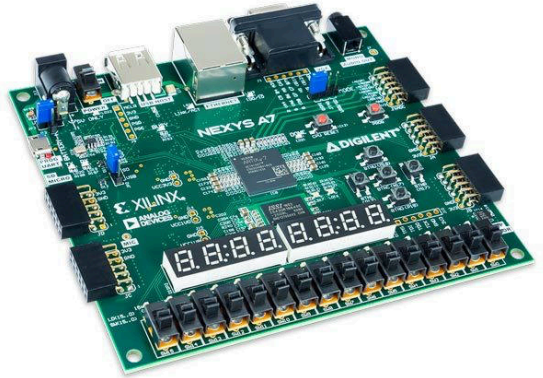


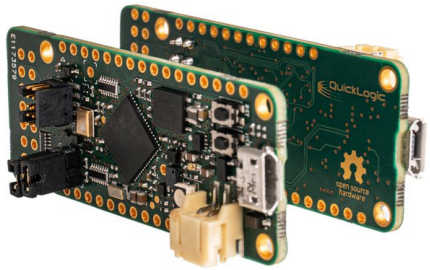
CORE-V MCU Physical Design

- Two realizations:
  - FPGA - Digilent Nexys A7 or Genesys 2
  - ASIC - GLOBALFOUNDRIES 22FDX
- ASIC coming early next year
  - Including Quicklogic eFPGA



- Prototype your custom accelerator for AI/ML applications
  1. FPGA
  2. eFPGA





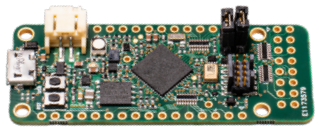
QuickLogic®

OpenHW Group BHAG

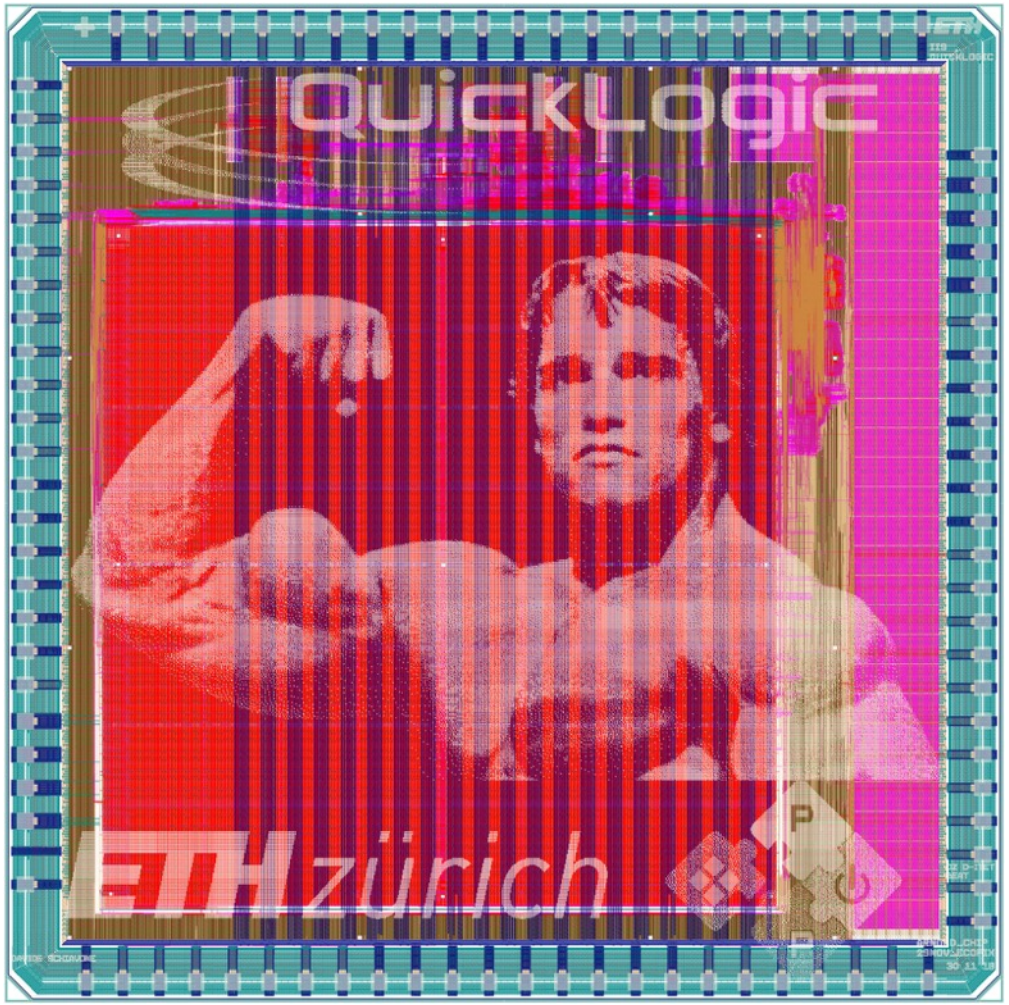
BIG HAIRY AUDACIOUS GOAL

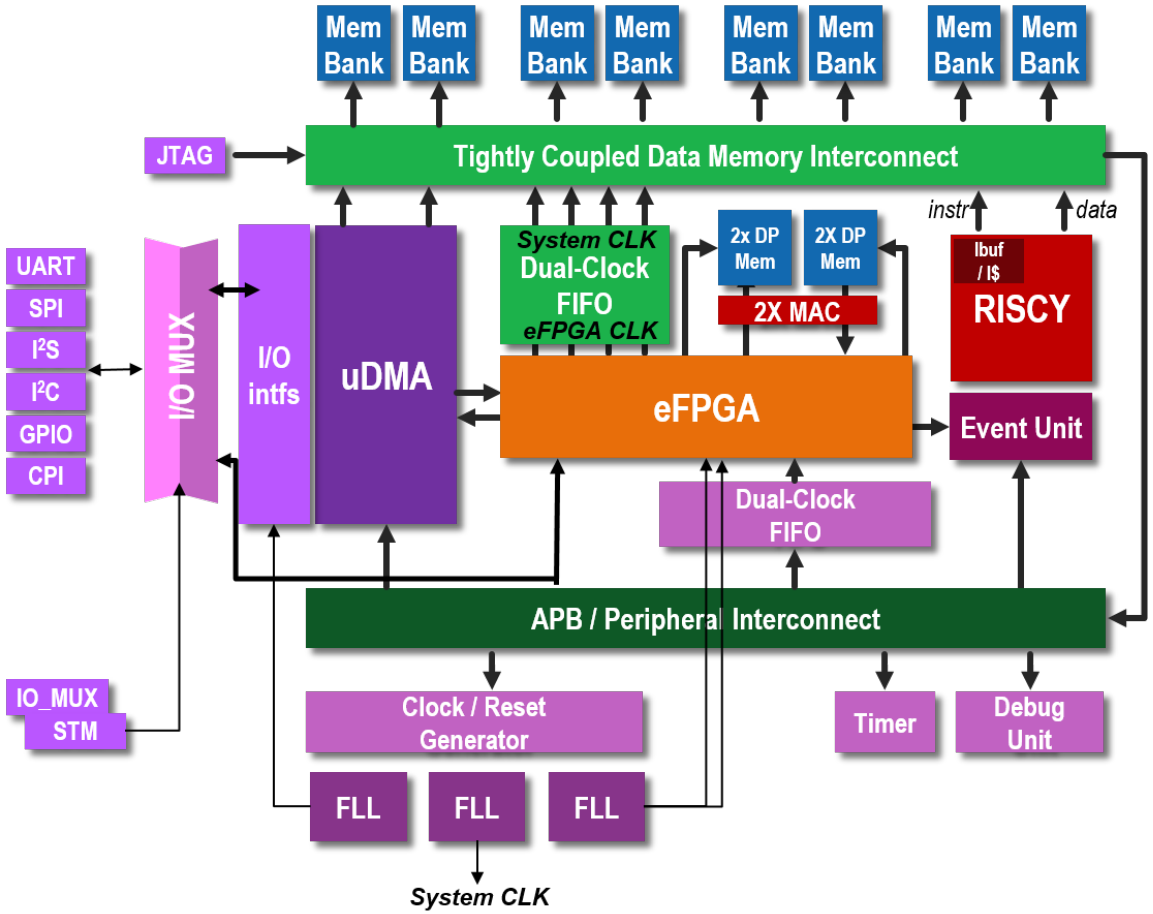
CORE-V™ APU & MCU SoCs

- Production Ready
- Using CORE-V CVA6 & CVE4 Cores
- Deep Sub-Micron SoCs
- On evaluation boards
- Running Linux / FreeRTOS
- Tapeout CORE-V MCU ~Q1'2021



Case Study: ETH Zurich “Arnold” Test Chip Platform

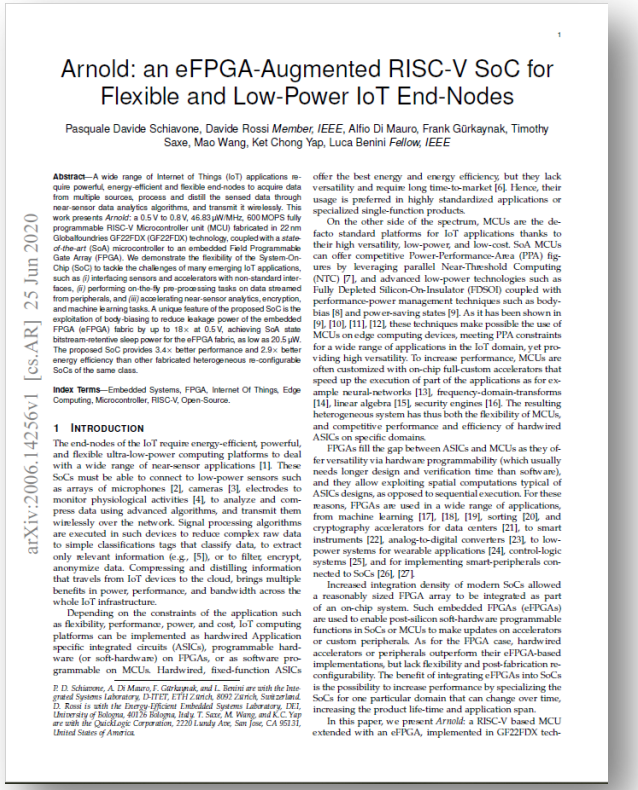




- ~600MHz, 3mm x 3mm die size on GF22FDX
- RISC-V with QuickLogic ArcticPro 2 eFPGA

Arnold – Heterogenous, Energy-Efficient Architecture

- Features
  - RISC-V General Purpose Processor
  - 512 KiB on-chip Memory
  - Broad set of peripheral I/O with memory access via µDMA
  - Tightly coupled eFPGA that supports
    - Direct connection to I/O
    - Shared memory accelerator interface
    - I/O filtering functions
    - Config and control interface to/from system
- Benefits
  - Energy efficient architecture enables flexibility to implement hardware partitioning of software requirements
  - Lower unit cost than vs discrete MCU / discrete FPGA implementations
  - OTA hardware upgrades
  - Lower NRE cost vs 'spinning an ASIC' for each derivative



Arnold: an eFPGA-Augmented RISC-V SoC for Flexible and Low-Power IoT End-Nodes

Pasquale Davide Schiavone, Davide Rossi *Member, IEEE*, Alfio Di Mauro, Frank Gürkaynak, Timothy Saxe, Mao Wang, Ket Chong Yap, Luca Benini *Fellow, IEEE*

Abstract—A wide range of Internet of Things (IoT) applications require powerful, energy-efficient and flexible end-nodes to acquire data from multiple sources, process and distill the sensed data through near-sensor data analytics algorithms, and transmit it wirelessly. This work presents *Arnold*, a 0.5 V to 0.8 V, 46.83 µW/MHz, 600 MOPS fully programmable RISC-V Microcontroller unit (MCU) fabricated in 22 nm Globalfoundries GF22FDX (GF22FDX) technology, coupled with a *state-of-the-art* (SoA) microcontroller to as embedded Field Programmable Gate Array (FPGA). We demonstrate the flexibility of the System-On-Chip (SoC) to tackle the challenges of many emerging IoT applications, such as *(i)* interfacing sensors and accelerators with non-standard interfaces, *(ii)* performing on-the-fly pre-processing tasks on data streamed from peripherals, and *(iii)* accelerating near-sensor analytics, encryption, and machine learning tasks. A unique feature of the proposed SoC is the exploitation of body-biasing to reduce leakage power of the embedded FPGA (eFPGA) fabric by up to 19× at 0.5 V, achieving SoA state bitstream-retentive sleep power for the eFPGA fabric, as low as 20.5 µW. The proposed SoC provides 3.4× better performance and 2.9× better energy efficiency than other fabricated heterogeneous re-configurable SoCs of the same class.

Index Terms—Embedded Systems, FPGA, Internet Of Things, Edge Computing, Microcontroller, RISC-V, Open-Source.

1 Introduction

The end-nodes of the IoT require energy-efficient, powerful, and flexible ultra-low-power computing platforms to deal with a wide range of near-sensor applications [1]. These SoCs must be able to connect to low-power sensors such as arrays of microphones [2], cameras [3], electrodes to monitor physiological activities [4], to analyze and compress data using advanced algorithms, and transmit them wirelessly over the network. Signal processing algorithms are executed in such devices to reduce complex raw data to simple classifications tags that classify data, to extract only relevant information (e.g., [5]), or to filter, encrypt, anonymize data. Compressing and distilling information that travels from IoT devices to the cloud, brings multiple benefits in power, performance, and bandwidth across the whole IoT infrastructure.

Depending on the constraints of the application such as flexibility, performance, power, and cost, IoT computing platforms can be implemented as hardwired Application specific integrated circuits (ASICs), programmable hardware (or soft-hardware) on FPGAs, or as software programmable on MCUs. Hardwired, fixed-function ASICs offer the best energy and energy efficiency, but they lack versatility and require long time-to-market [6]. Hence, their usage is preferred in highly standardized applications or specialized single-function products.

On the other side of the spectrum, MCUs are the de-facto standard platforms for IoT applications thanks to their high versatility, low-power, and low-cost. SoA MCUs can offer competitive Power-Performance-Area (PPA) figures by leveraging parallel Near-Threshold Computing (NTC) [7], and advanced low-power technologies such as Fully Depleted Silicon-On-Insulator (FDSOI) coupled with performance-power management techniques such as body-bias [8] and power-saving states [9]. As it has been shown in [9], [10], [11], [12], these techniques make possible the use of MCUs on edge computing devices, meeting PPA constraints for a wide range of applications in the IoT domain, yet providing high versatility. To increase performance, MCUs are often customized with on-chip full-custom accelerators that speed up the execution of part of the applications as for example neural-networks [13], frequency-domain-transforms [14], linear algebra [15], security engines [16]. The resulting heterogeneous system has thus both the flexibility of MCUs, and competitive performance and efficiency of hardwired ASICs on specific domains.

FPGAs fill the gap between ASICs and MCUs as they offer versatility via hardware programmability (which usually needs longer design and verification time than software), and they allow exploiting spatial computations typical of ASICs designs, as opposed to sequential execution. For these reasons, FPGAs are used in a wide range of applications, from machine learning [17], [18], [19], sorting [20], and cryptography accelerators for data centers [21], to smart instruments [22], analog-to-digital converters [23], to low-power systems for wearable applications [24], control-logic systems [25], and for implementing smart-peripherals connected to SoCs [26], [27].

Increased integration density of modern SoCs allowed a reasonably sized FPGA array to be integrated as part of an on-chip system. Such embedded FPGAs (eFPGAs) are used to enable post-silicon soft-hardware programmable functions in SoCs or MCUs to make updates on accelerators or custom peripherals. As for the FPGA case, hardwired accelerators or peripherals outperform their eFPGA-based implementations, but lack flexibility and post-fabrication re-configurability. The benefit of integrating eFPGAs into SoCs is the possibility to increase performance by specializing the SoCs for one particular domain that can change over time, increasing the product life-time and application span.

In this paper, we present *Arnold*: a RISC-V based MCU extended with an eFPGA, implemented in GF22FDX tech-

P. D. Schiavone, A. Di Mauro, F. Gürkaynak, and L. Benini are with the Integrated Systems Laboratory, D-ITET, ETH Zürich, 8092 Zürich, Switzerland. D. Rossi is with the Energy-Efficient Embedded Systems Laboratory, DEI, University of Bologna, 40136 Bologna, Italy. T. Saxe, M. Wang, and K.C. Yap are with the QuickLogic Corporation, 2220 Lundy Ave, San Jose, CA 95131, United States of America.

https://arxiv.org/pdf/2006.14256.pdf

CORE-V™ MCU SoC
Tapeout Late Q1 2021

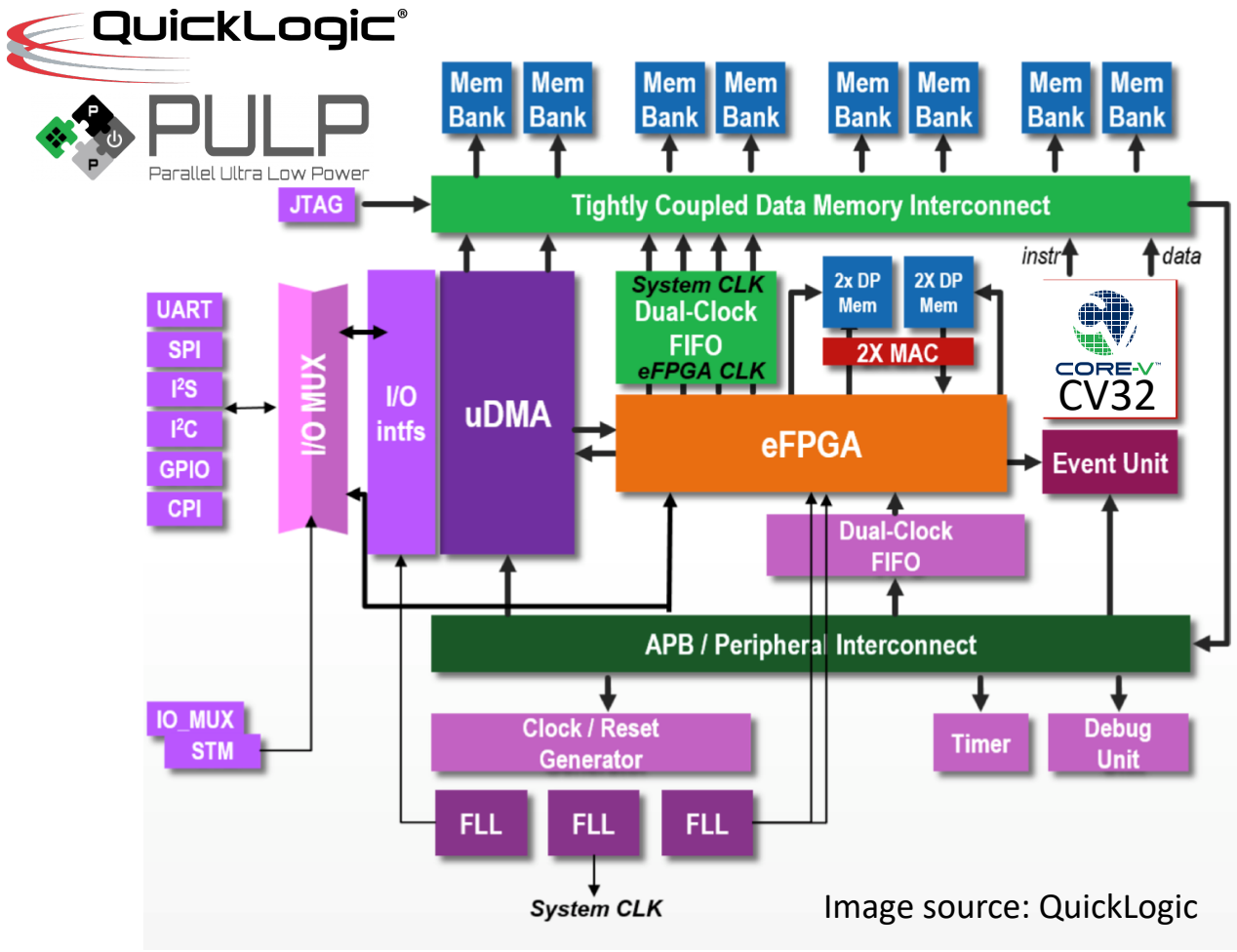
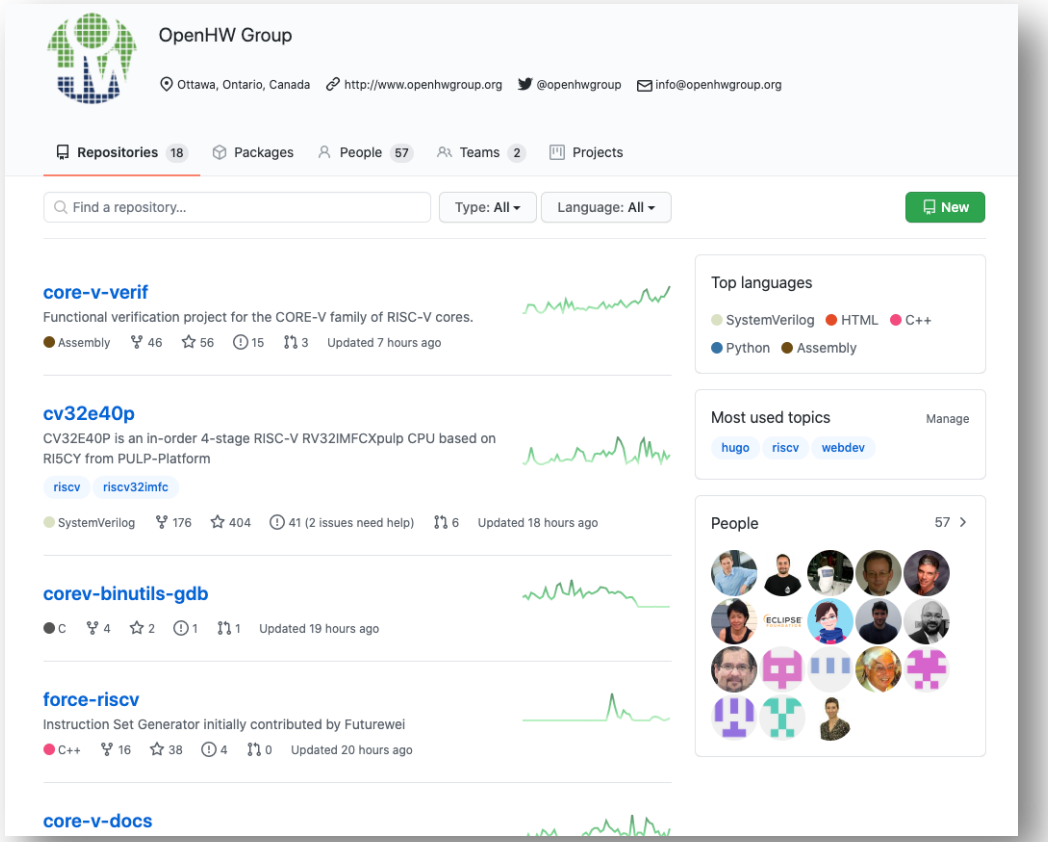


Image source: QuickLogic

- Project announced at OSDForum Sept 2020
- Real Time Operating System (e.g. FreeRTOS) capable ~600+MHz CV32 MCU host CPU
- Embedded FPGA fabric with hardware accelerators from QuickLogic
- Multiple low power peripheral interfaces (SPI, GPIO, I2C, HyperRAM, CAMIF, etc.) for interfacing with sensors, displays, and connectivity modules
- Built in 22FDX with GF

Software

- The SDK contains all the tools and runtime support for CORE-V based microcontrollers
- Provides:
  - HAL
  - `crt0` and linker scripts
  - higher level functions (API)
- CORE-V toolchain and SW ecosystem under development



Get involved: https://github.com/openhwgroup/core-v-sw

Current Status

- CORE-V MCU CV32E40P running on Genesys 2 with simple runtime
  - Code built with Embecosm's CORE-V GCC development tools
  - GDB/OpenOCD debug
  - Come join us and contribute at: https://github.com/openhwgroup/core-v-mcu
- CORE-V MCU with Ashling RiscFree/Opella-XD JTAG probe running on Genesys 2

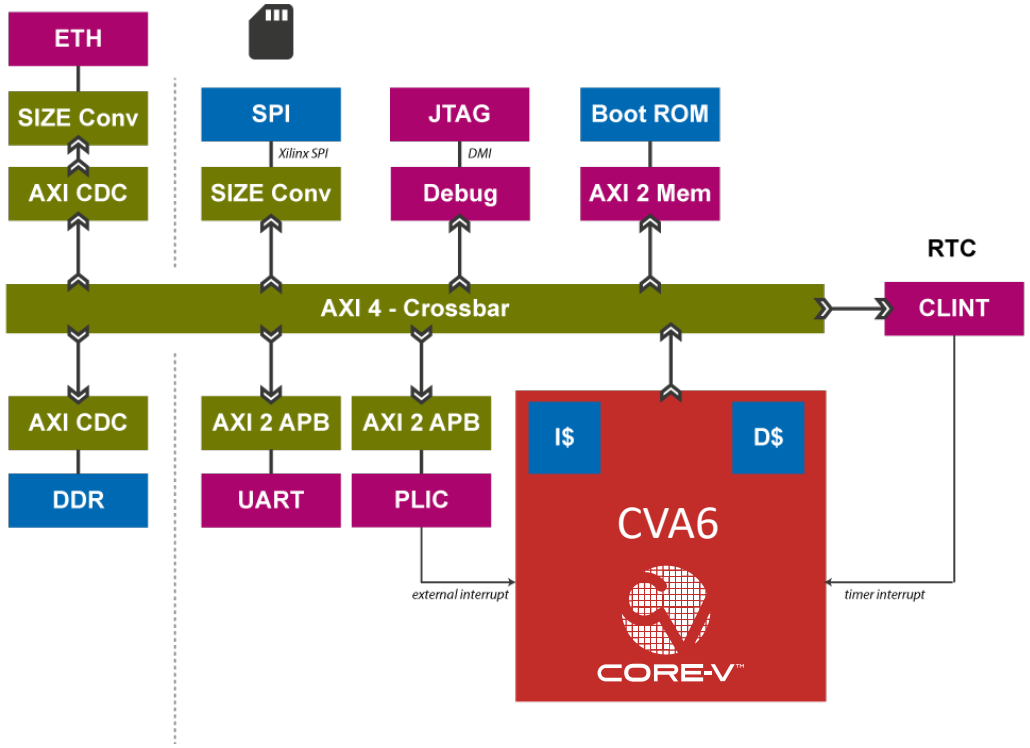
Next Steps

- Further testing and debugging of the implementation
- Integration of more system peripherals
- More complex SDK
- Architectural enhancements, such as implementing Open Bus Interface with CV32 now supports
- Integration/validation of proprietary and open-source tools (hardware, software)
- Training materials, demos

Looking for collaborators to help!

CORE-V APU

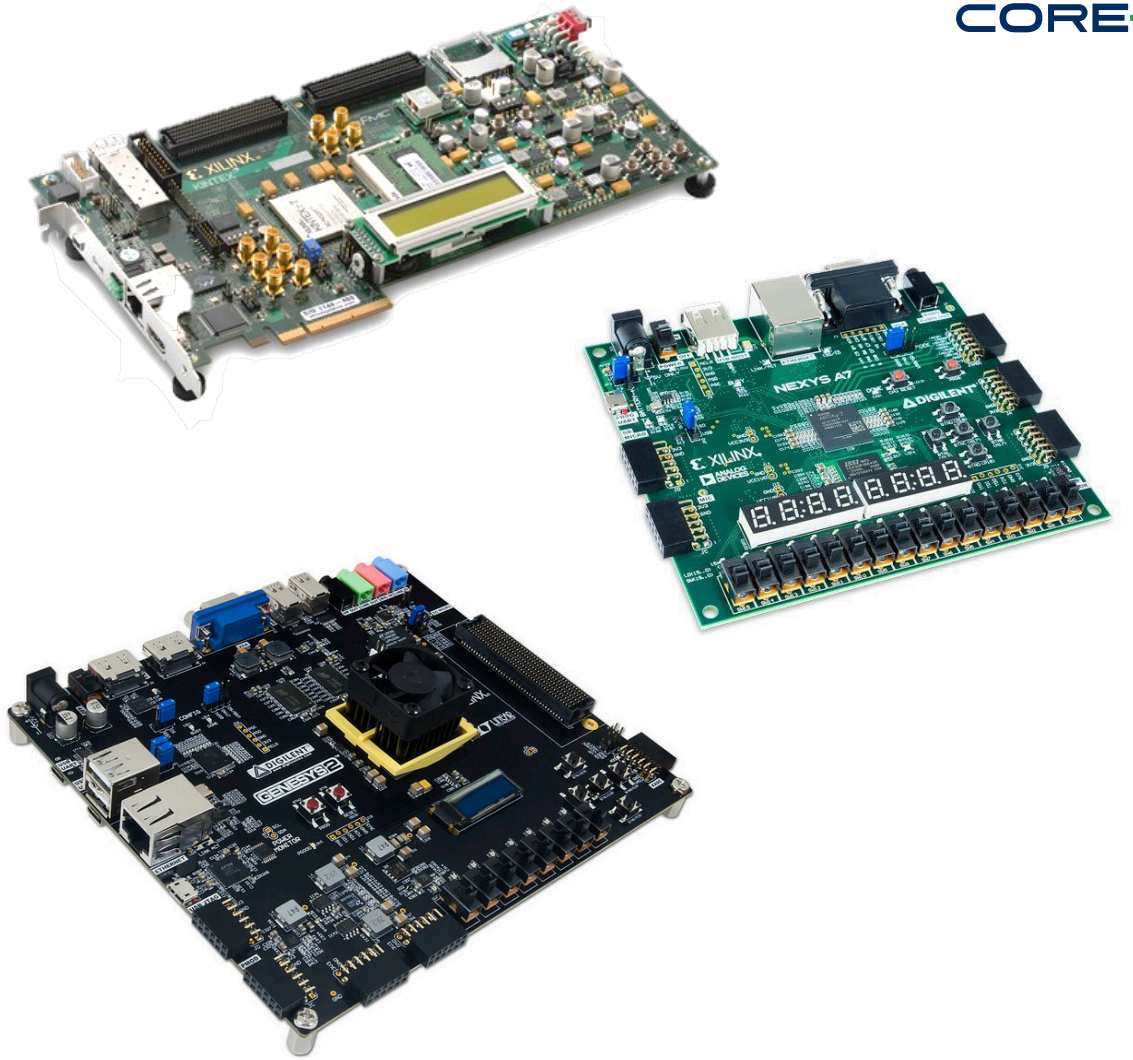
- Minimum set of peripherals to boot Linux
- Code is on SD Card
- Zero-stage bootloader is in boot ROM
- UART as main UI
- Ethernet for network connectivity



https://github.com/openhwgroup/cva6

CORE-V MCU & APU FPGA

- Default Board: **Genesys 2**
  - Wide-spread adoption
  - Rich set of board peripherals
  - Reasonably cheap (discounts through OHW, academics)
- Alternative: **Nexys A7**
  - Cheaper
  - Higher availability
- “User ports”:
  - VC707, KC705
  - Ultrascale Boards



RISC-V Debug

- Draft specification 0.13
- Defines debug registers for
  - run/halt/single-step
  - reading/writing GPR, FPR and CSRs
  - Querying hart status
- JTAG interface
- OpenOCD support
- Our choice: **Execution Based**

- Standard allows for different debug probes to be used
- IDE integration

ASHLING

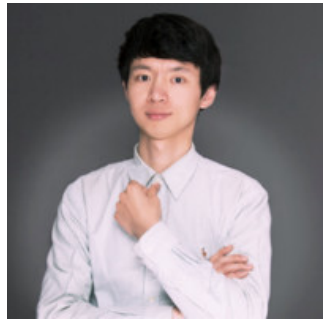
Debug (Detailed)

- Special Debug Mode
  - Less intrusive, leverage existing pipeline
  - DPC, debug-interrupt, `dret` and `ebreak`
- Halt, Resume, (Single-)step
  - Only required command: Abstract read (read floating-point register, register and CSR)
  - Debug Transport Protocol (orthogonal)
    - Currently only JTAG specified
- **SystemVerilog** reference implementation available
  - https://github.com/pulp-platform/riscv-dbg

Meet the Team: HW and SW Task Groups

- Hardware TG
  - Hugh Pollitt-Smith (CMC)
  - Tim Saxe (QuickLogic)
- Software TG
  - Jeremy Bennet (Embecosm)
  - Yunhai Shang (Alibaba T-Head)




OpenHW Director of Engineering: Florian Zaruba

Get on Board!

- Development is in the open!
- CORE-V MCU
  - https://github.com/openhwgroup/core-v-mcu
- CORE-V APU
  - Directly in the core's repository
  - https://github.com/openhwgroup/cva6
- Thank you and be sure to visit the OpenHW booth here at the Virtual RISC-V Summit with demos from…