OSDForum

OPEN SOURCE DEVELOPER FORUM

CORE-V™

ETH Zurich Arnold Test Chip & CORE-V MCU Project Proposal

Brian Faith faith@quicklogic.com

Tim Saxe saxe@quicklogic.com

T @QuickLogic_Corp

www.quicklogic.com

Sep 2020

QuickLogic®

Open Source FPGA Tooling: Our Journey from Resistance to Adoption

Brian Faith
CEO

Sept 2020

QuickLogic at a Glance

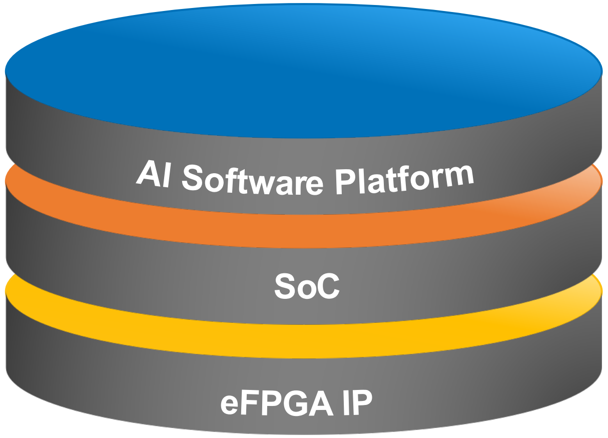
Snapshot

- **Founded:** 1989, public since 1999
- **Ticker:** QUIK (NASDAQ)
- **Headquarters:** San Jose, CA

What We Do

- QuickLogic is a platform company that enables our customers to quickly and easily create intelligent ultra-low power endpoints to build a smarter, more connected world
- We develop ultra-low power, multi-core semiconductor platforms and hardware- and software-based IP for AI, voice and sensor processing applications

End-to-End Solutions


AI Software Platform

SoC

eFPGA IP

Target Markets

- Hearables and wearables
- Consumer and industrial IoT
- Smartphones and tablets
- Consumer electronics
- AI-enabled devices
- Aerospace and defense devices

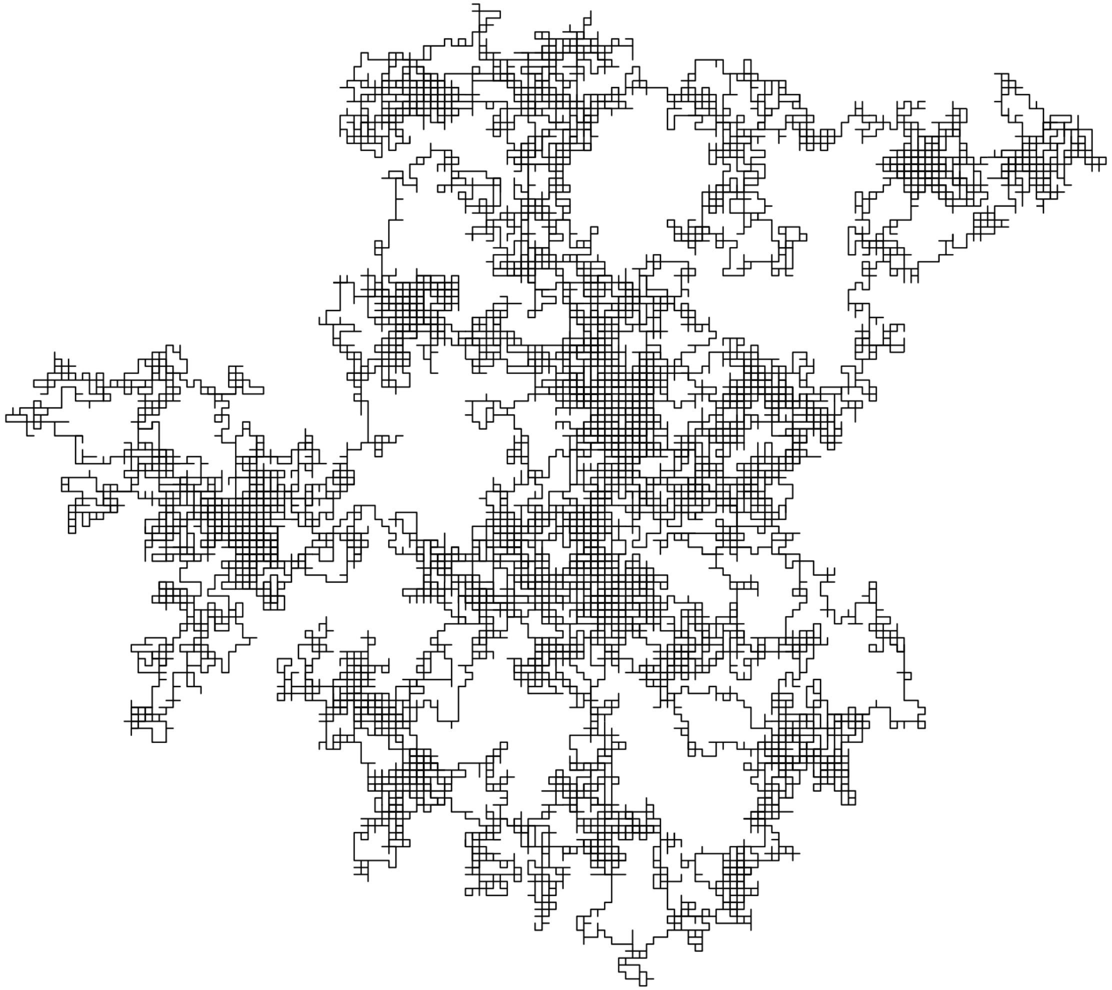
Resistance

- Over the past 30+ years, 60+ Programmable Logic companies have come and gone
- With so many innovative architectures and great products, why did so many fail?
- Almost all pointed back to challenges with ***software***

And, for those of us who have had robust software…

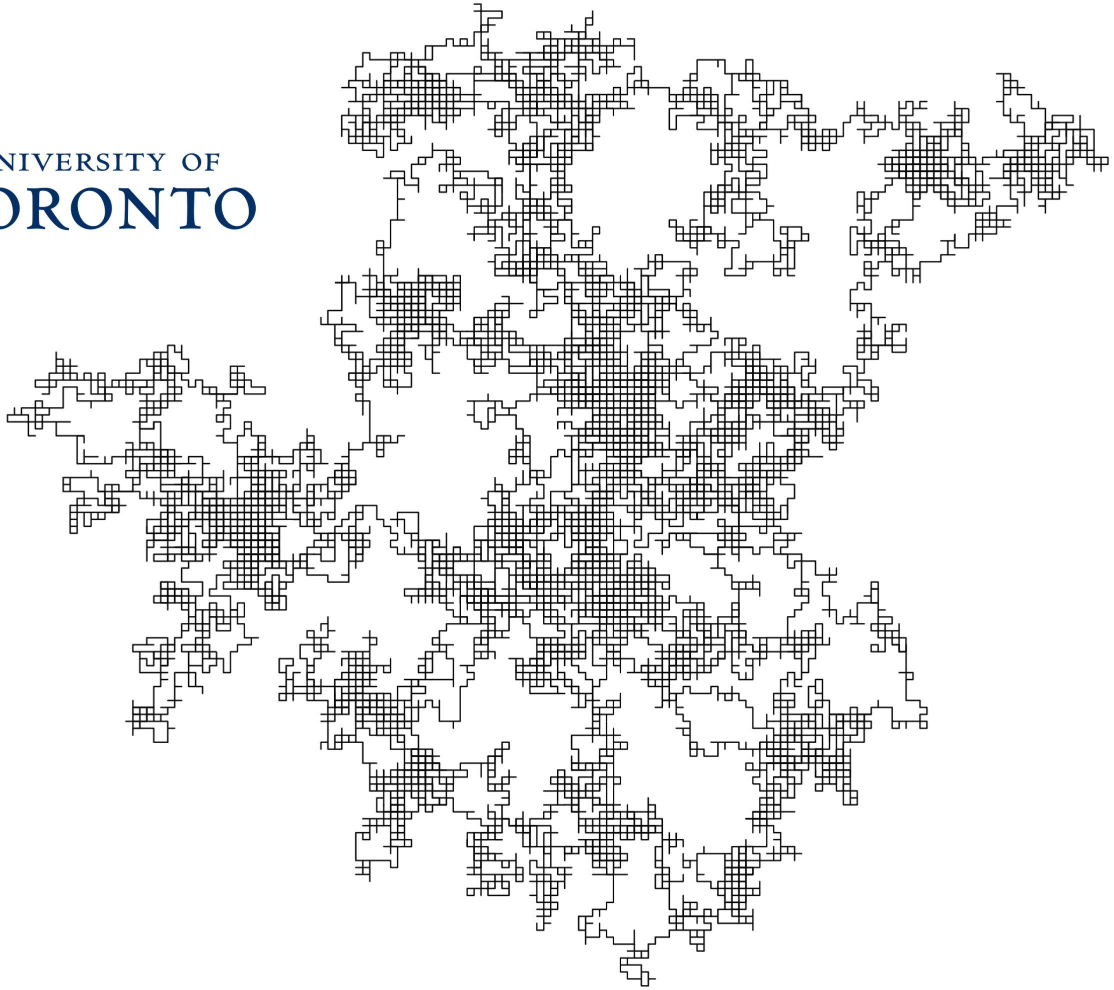
The Walled Garden

A Somewhat Random Path



A Somewhat Random Path

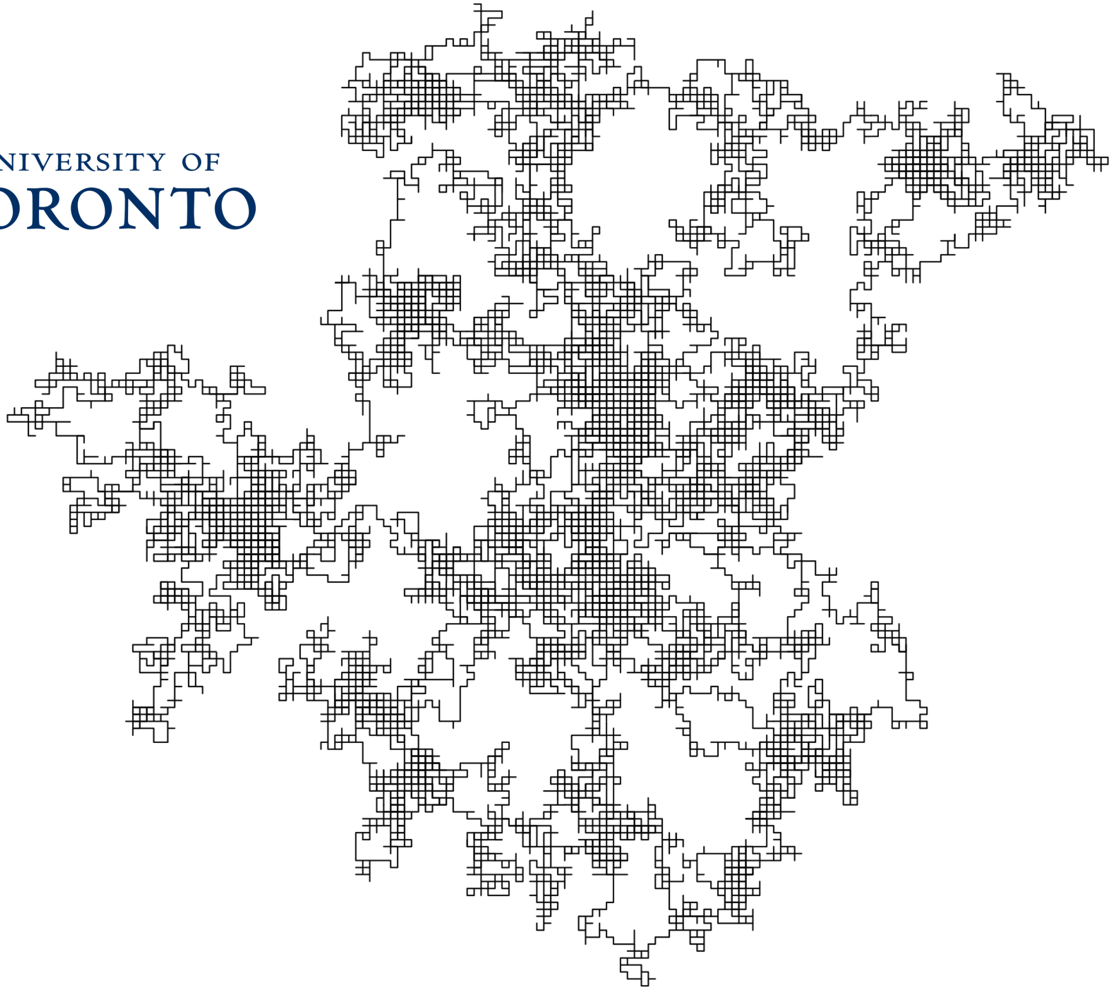
UNIVERSITY OF TORONTO



A Somewhat Random Path

Google

UNIVERSITY OF TORONTO



A Somewhat Random Path

Google

UNIVERSITY OF TORONTO

Risk breaking up my walled garden?

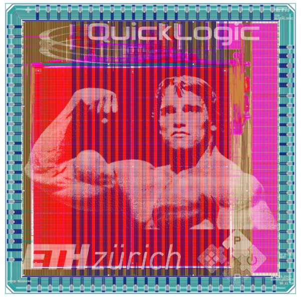
No, thank you.

A Somewhat Random Path

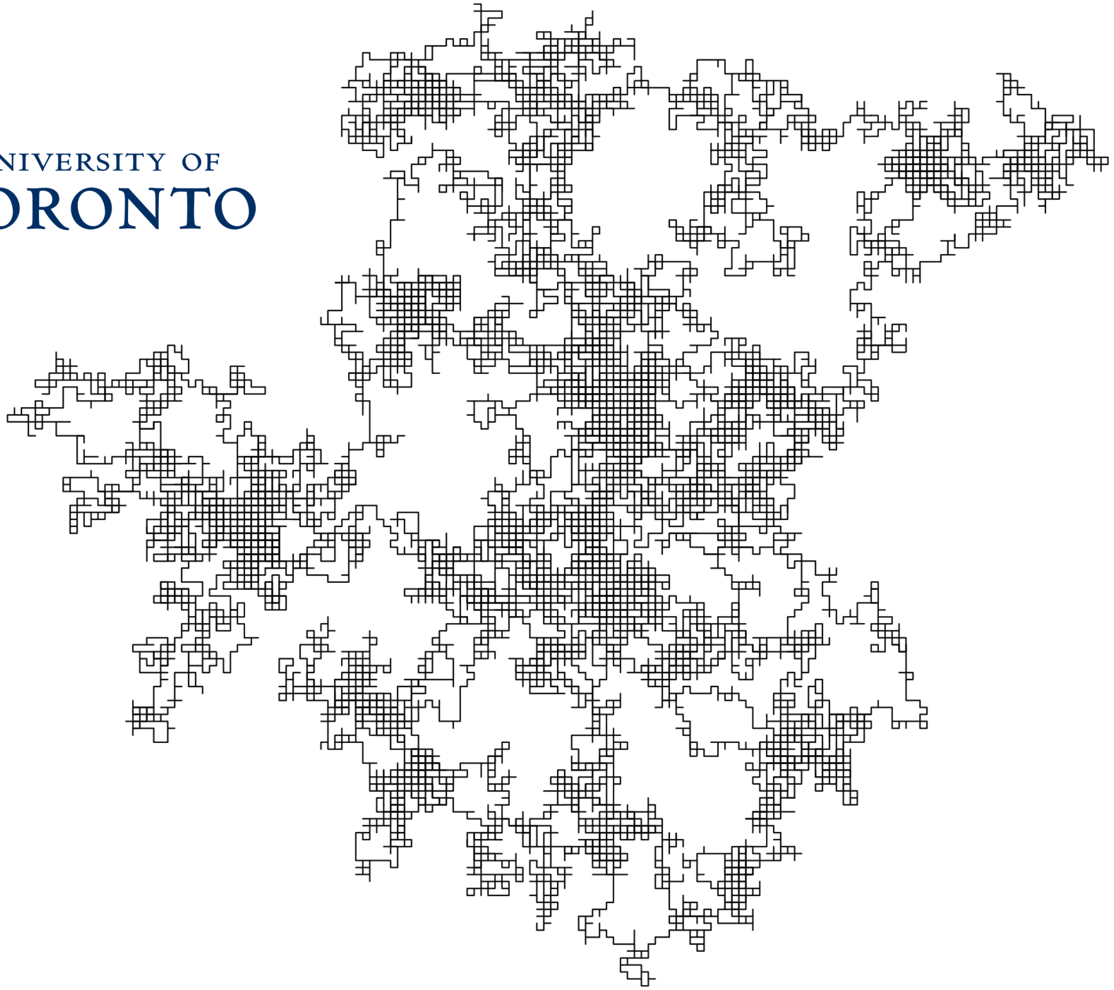
Google

UNIVERSITY OF TORONTO

ETH zürich



RISC-V MCU with 22FDX eFPGA

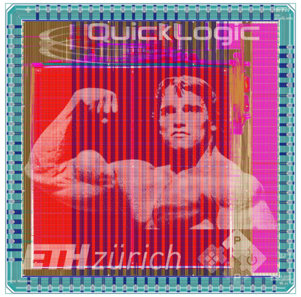


A Somewhat Random Path

Google

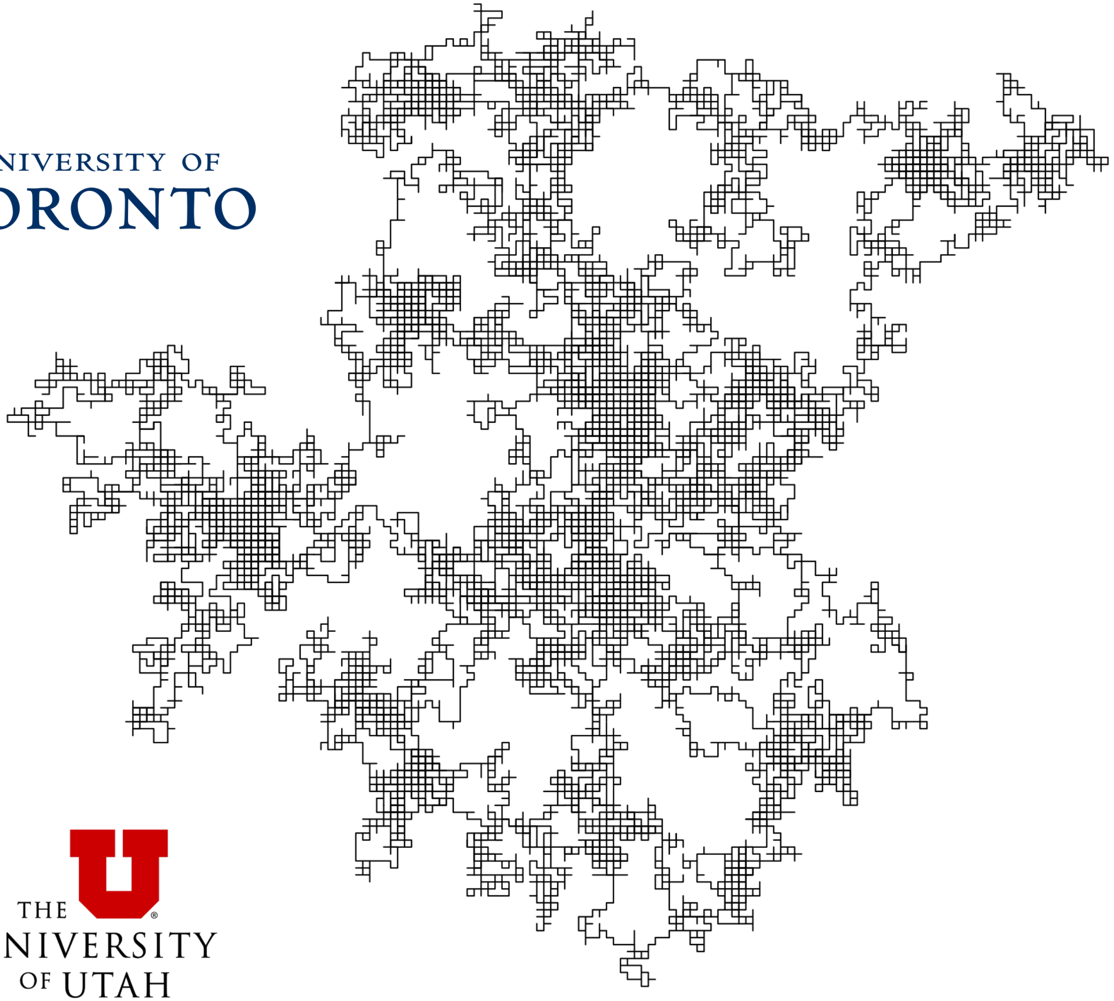
UNIVERSITY OF TORONTO

ETH zürich



RISC-V MCU with
22FDX eFPGA

THE UNIVERSITY OF UTAH

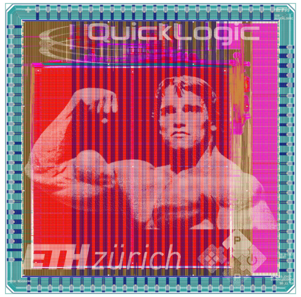


A Somewhat Random Path

Google

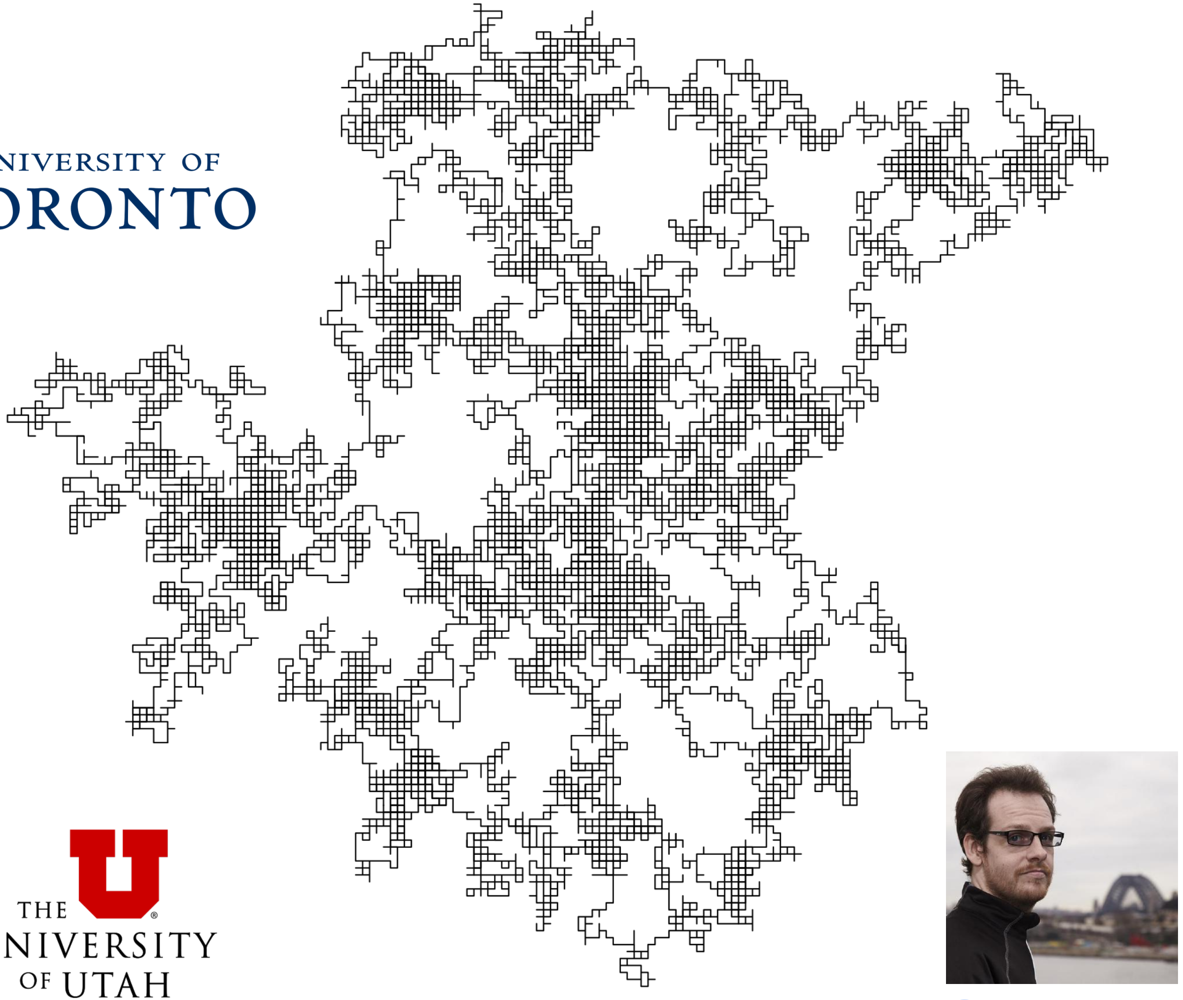
UNIVERSITY OF TORONTO

ETH zürich



RISC-V MCU with 22FDX eFPGA

THE UNIVERSITY OF UTAH



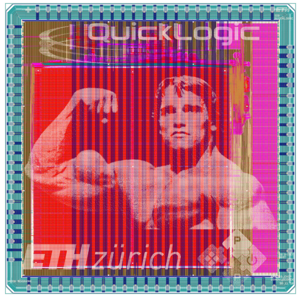
Google

A Somewhat Random Path

Google

UNIVERSITY OF TORONTO

ETH zürich



RISC-V MCU with
22FDX eFPGA

THE UNIVERSITY OF UTAH

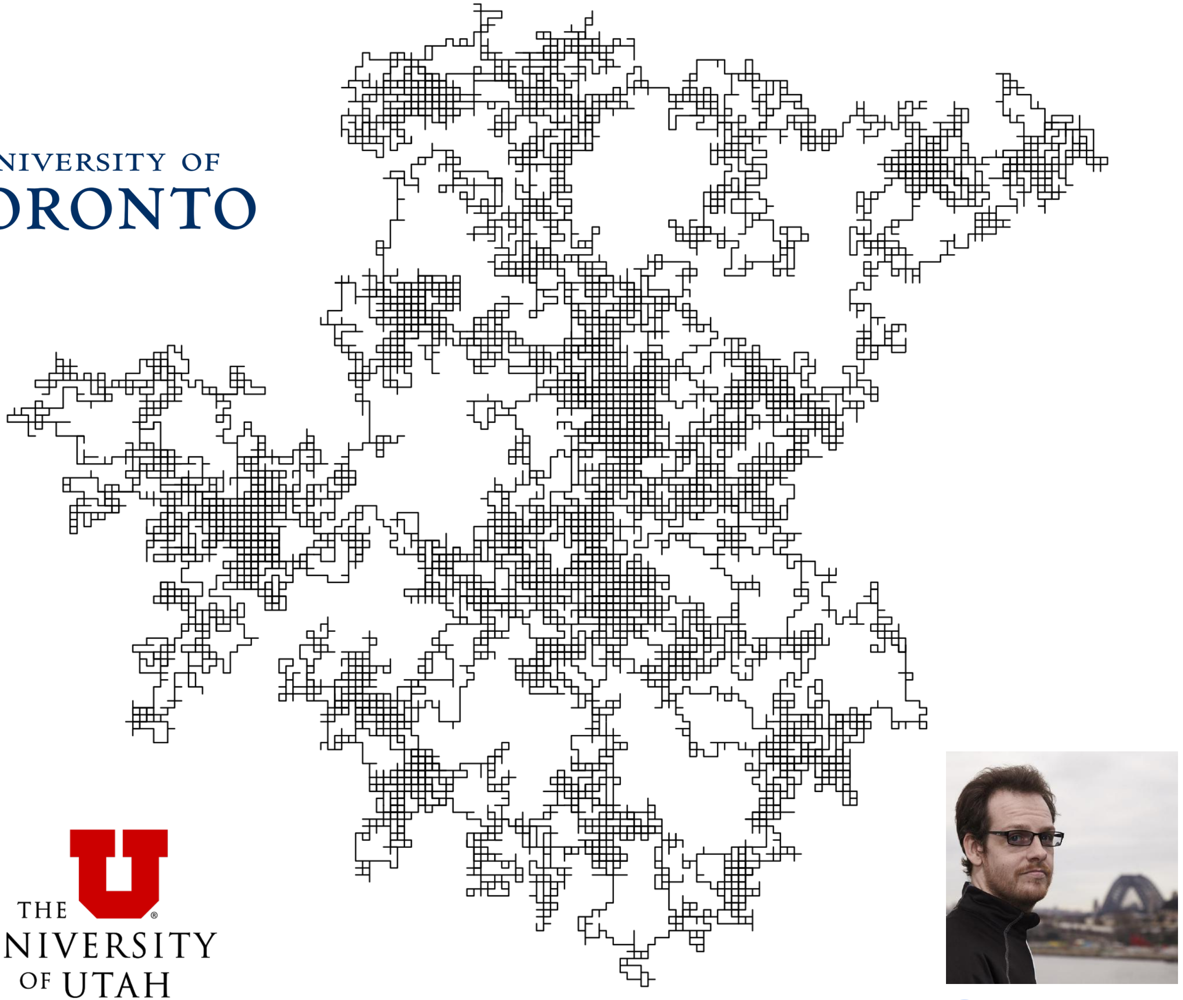
Hmmm. Maybe there's something here.

Google

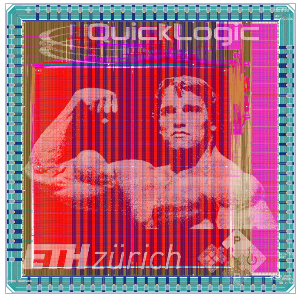
A Somewhat Random Path

Google

UNIVERSITY OF TORONTO



ETH zürich



RISC-V MCU with 22FDX eFPGA

THE UNIVERSITY OF UTAH

Google

FOSSi Foundation

Adoption

QuickLogic®

QORC

QuickLogic Open Reconfigurable Computing

Disrupting the Programmable Logic Status Quo

- First Programmable Logic company to ***actively contribute to*** a fully open source suite of development tools for its FPGA devices and embedded FPGA (eFPGA) technology
- Full RTL-to-bitstream support with complete architecture and accurate timing support
- Changing the paradigm of using existing hardware devices or reference designs that are inflexible and forces developers to adapt rather than having devices adapt to developers needs
- Developed in collaboration with open source industry influencers:

Google

antmicro
EMBEDDED SYSTEMS

Disrupting the Programmable Logic Status Quo

- First Programmable Logic company to ***actively collaborate to create***
  - Fully open source suite of development tools
    for our FPGA devices and embedded FPGA (eFPGA) technology
- Full RTL-to-bitstream support with complete architecture and accurate timing support

Changing the Paradigm…

- Let the community create and adapt hardware to developers' needs
  - Instead of forcing developers to adapt to existing hardware devices or reference designs
- Work in collaboration with open source industry influencers:

Google

antmicro
EMBEDDED SYSTEMS

Fully Committed to Open Source FPGA Tools

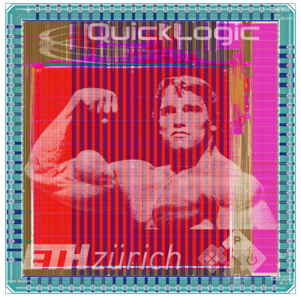
Google

UNIVERSITY OF TORONTO

QORC

SymbiFlow

ETH zürich



RISC-V MCU with 22FDX eFPGA

antmicro EMBEDDED SYSTEMS

RENODE™

THE UNIVERSITY OF UTAH

Google

FOSSi Foundation

Fully Committed to Open Source FPGA Tools

Google

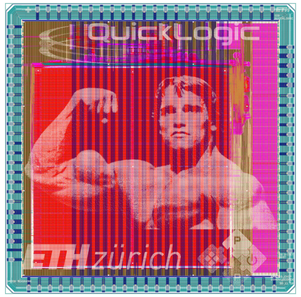
UNIVERSITY OF TORONTO

CROWD SUPPLY

QORC

SymbiFlow

ETH zürich



RISC-V MCU with 22FDX eFPGA

antmicro EMBEDDED SYSTEMS

RENODE™

THE UNIVERSITY OF UTAH

Google

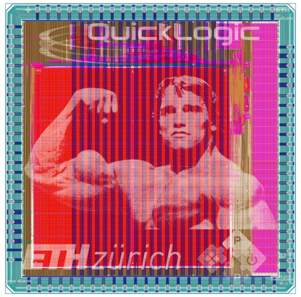
FOSSi Foundation

Fully Committed to Open Source FPGA Tools

Google

UNIVERSITY OF TORONTO

ETH zürich



RISC-V MCU with
22FDX eFPGA

THE UNIVERSITY OF UTAH

QORC

CHIPS ALLIANCE

CROWD SUPPLY

SymbiFlow

antmicro
EMBEDDED SYSTEMS

RENODE™

Google

FOSSi Foundation

Fully Committed to Open Source FPGA Tools

Google

UNIVERSITY OF TORONTO

OPENHW GROUP™
PROVEN PROCESSOR IP

CHIPS ALLIANCE

CROWD SUPPLY

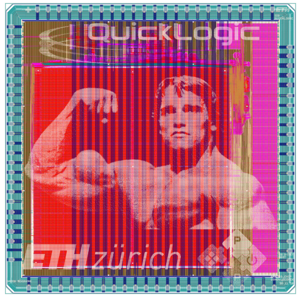
QORC

SymbiFlow

antmicro
EMBEDDED SYSTEMS

RENODE™

ETH zürich



RISC-V MCU with 22FDX eFPGA

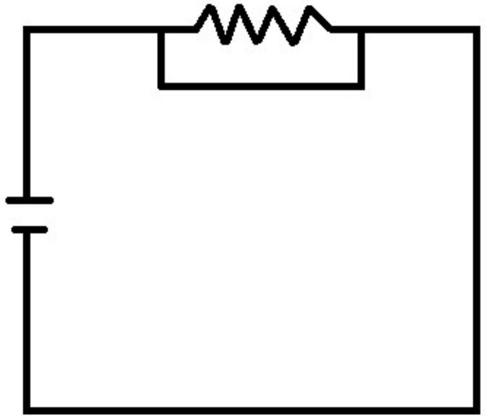
THE UNIVERSITY OF UTAH

Google

FOSSi Foundation

Our Journey

- The Open Source Community’s persistence and passion is remarkable
- Open Source FPGA Tooling is ready for prime time
- There will be a time in the not-so-distance future where they are the norm vs exception



ETH Zurich Arnold Test Chip & CORE-V MCU Project Proposal

Brian Faith faith@quicklogic.com

Tim Saxe saxe@quicklogic.com

T @QuickLogic_Corp

www.quicklogic.com

CORE-V™ Family of RISC-V Cores

- Initial contribution of open source RISC-V cores from ETH Zurich PULP Platform
  - Very popular, industry adopted cores
- OpenHW Group becomes the official committer for these repositories

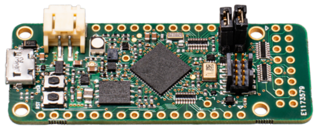
Core	Bits/Stages	Description
CVE4 (RI5CY)	Embedded 32 bit 4-stage	An Embedded class 4-stage core that implements, the RV32IMFCXpulp, has an optional 32-bit FPU supporting the F extension and instruction set extensions for DSP operations, including hardware loops, SIMD extensions, bit manipulation and post-increment instructions.
CVA6 (Ariane)	Application 32 & 64 bit 6-stage	An Application class 6-stage, single issue, in-order CPU implementing RV64GC extensions with three privilege levels M, S, U to fully support a Unix-like (Linux, BSD, etc.) operating system. It has configurable size, separate TLBs, a hardware PTW and branch-prediction (branch target buffer, branch history table and a return address stack).

OpenHW Group BHAG

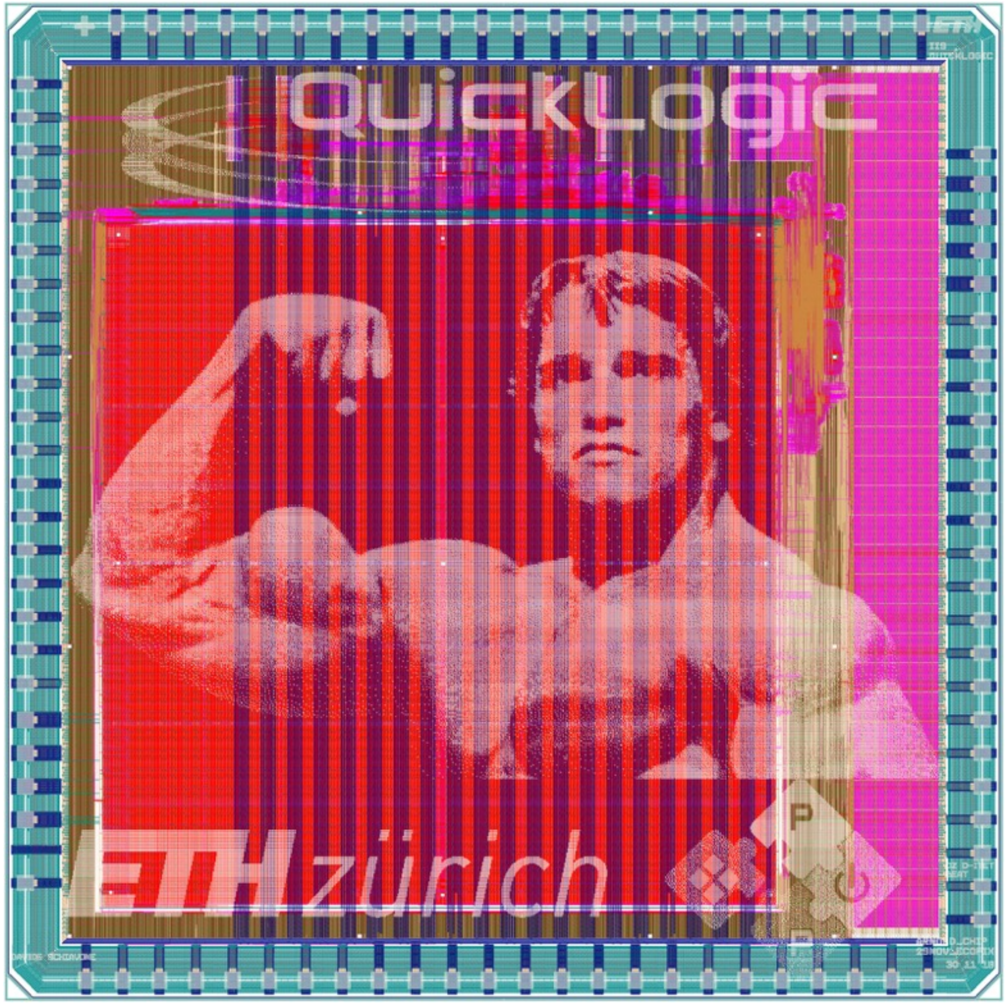
BIG HAIRY AUDACIOUS GOAL

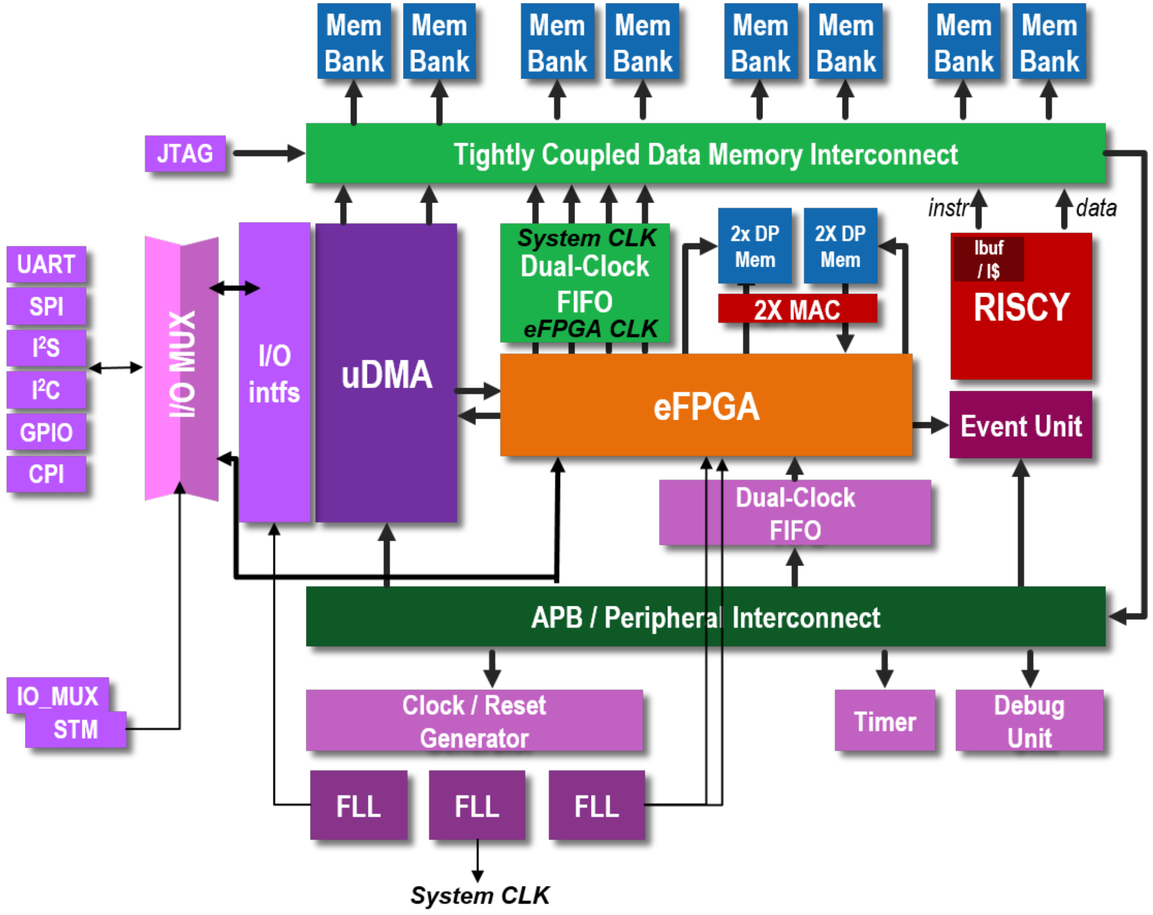
CORE-V™ APU & MCU SoCs

- Production Ready,
- Using CORE-V CVA6 & CVE4 Cores,
- Deep Sub-Micron SoCs,
- On eval boards,
- Running Linux / Zephyr
- Tapeout CORE-V MCU ~Q1’2021



Case Study: ETH Zurich “Arnold” Test Chip Platform

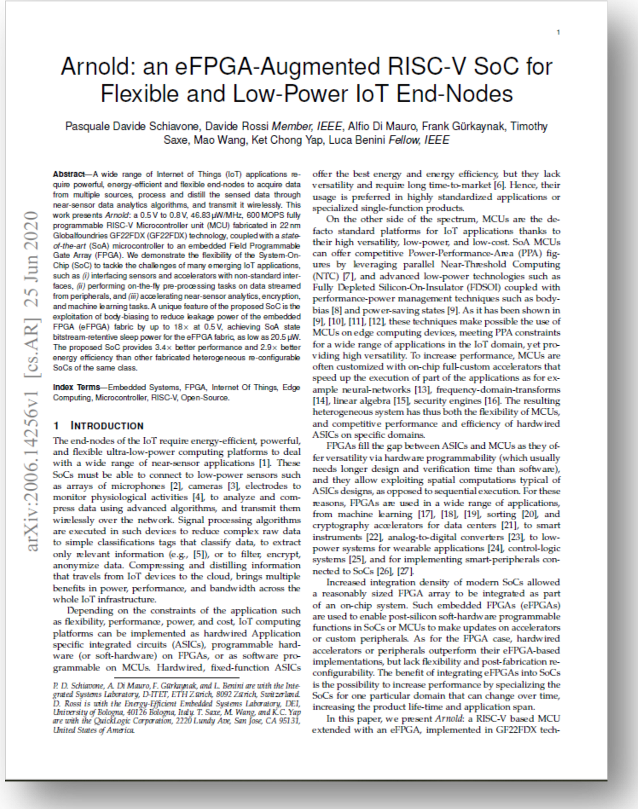




- 425MHz, 3mmx3mm die size on GF 22FDX
- RISC-V with QuickLogic ArcticPro 2 eFPGA

Arnold – Heterogenous, Energy-Efficient Architecture

- Features
  - RISC-V General Purpose Processor
  - 512 KB Onboard 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



1

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*

arXiv:2006.14256v1 [cs.AR] 25 Jun 2020

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 an 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 18× 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.

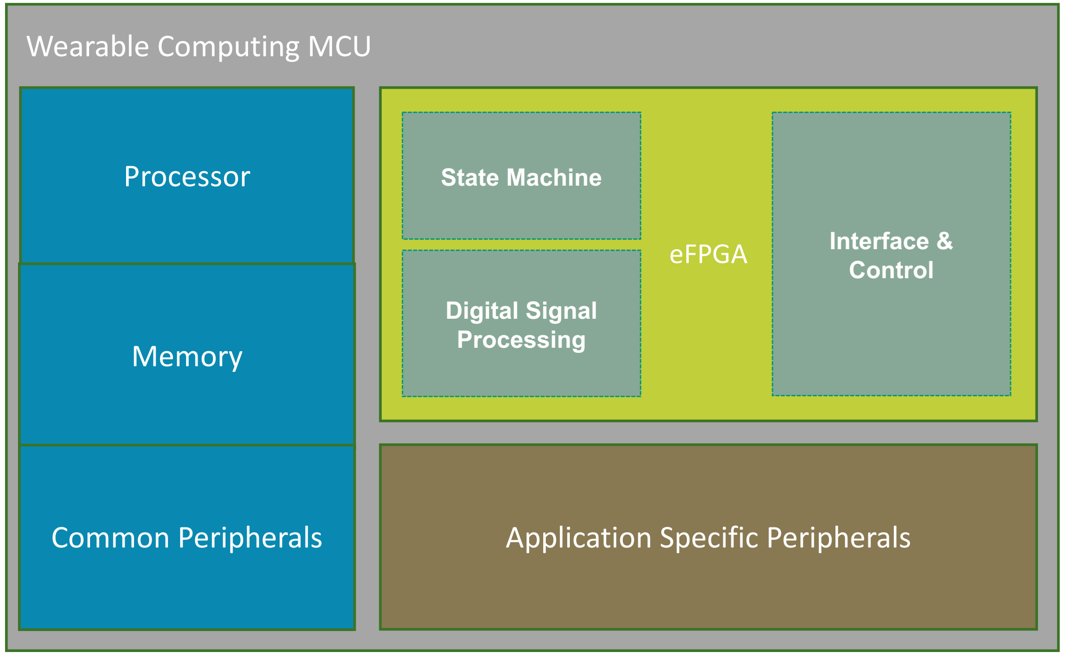
In this paper, we present *Arnold*: a RISC-V based MCU extended with an eFPGA, implemented in GF22FDX tech-

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, 40126 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

System Partitioning in Low Power Computing

- Software running on a processor is the most flexible method to implement any function, however it may:
  - Not fit in the processor's available compute capability
  - Not fit in the processor's available memory footprint
  - Not be able keep up with required AI inference rate within power budget
  - Not be able to implement strict I/O timing requirements
- In the above cases, onboard eFPGA is the ideal implementation vehicle



CORE-V™ MCU SoC Tapeout Early Q1 2021

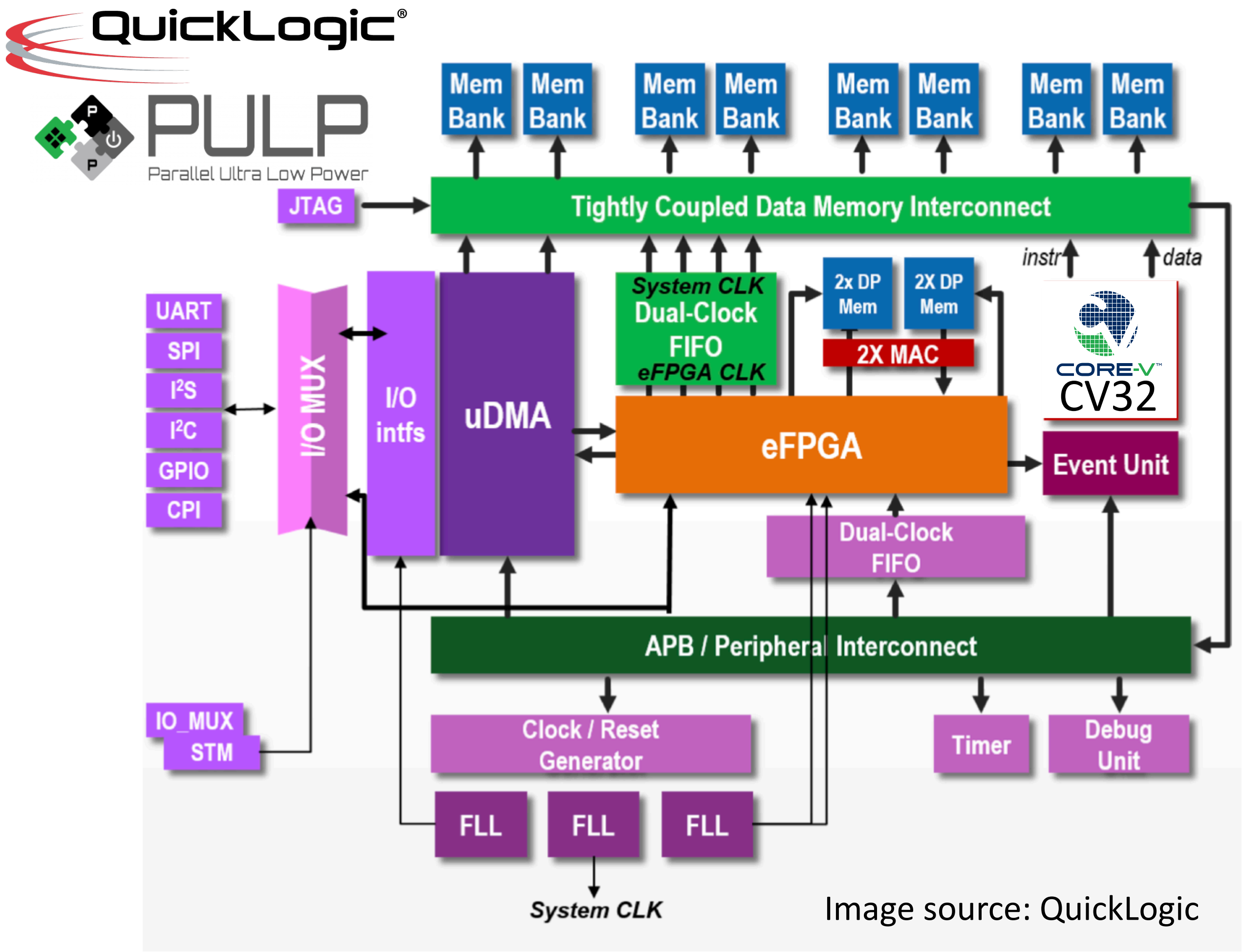


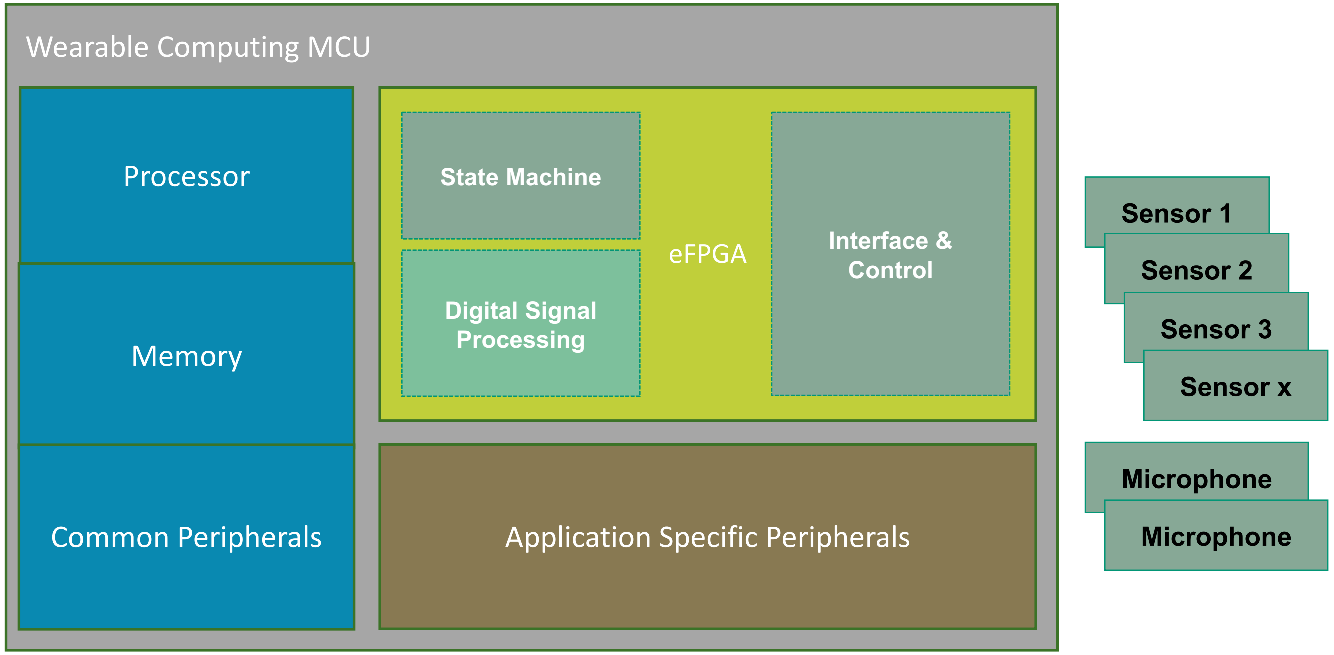
Image source: QuickLogic

- Project announced at Open Source Developer Forum Sept 2020
- Real Time Operating System (e.g. Zephyr) 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

Backup

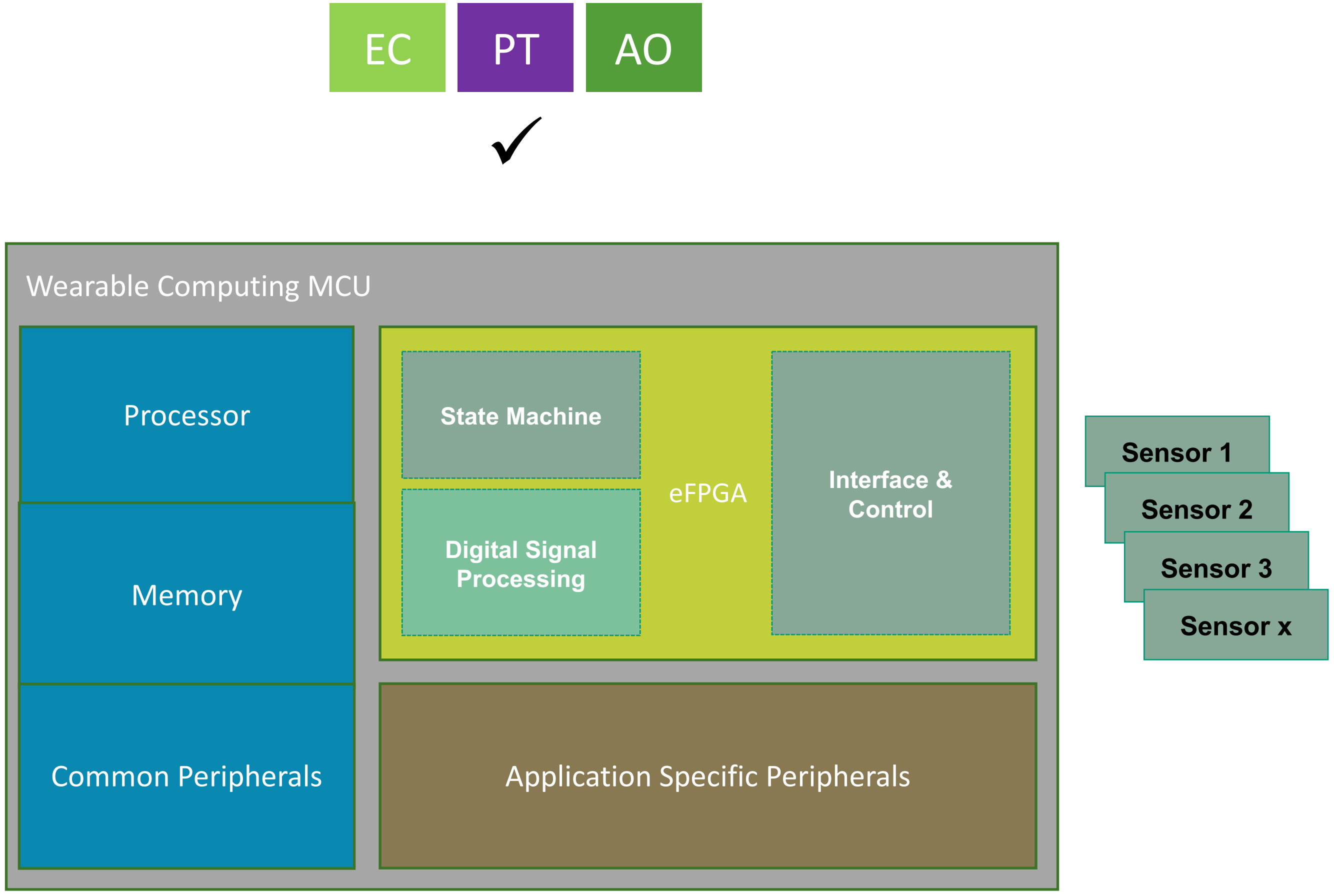
eFPGA Use Case – HW Offloaded/Accelerated DSP

- DSP functions (e.g. filters) are an integral part of many edge applications such as audio signal processing, and voice recognition
- In lieu of running all of those DSP functions as software on a general purpose processor, certain functions can be offloaded or accelerated to an eFPGA with tightly coupled DSP blocks
- The benefit is a more platform with more computational capability and flexibility to change implementation based on requirements



eFPGA Use Case – I/O Expansion

- Choosing the right specification and combination of peripheral I/O can challenging
- With integrated eFPGA, new & evolving I/O standards can be implemented post-tape out to extend the life of a mask set



EC PT AO

✓

Wearable Computing MCU

Processor

Memory

Common Peripherals

State Machine

Digital Signal Processing

eFPGA

Interface & Control

Application Specific Peripherals

Sensor 1

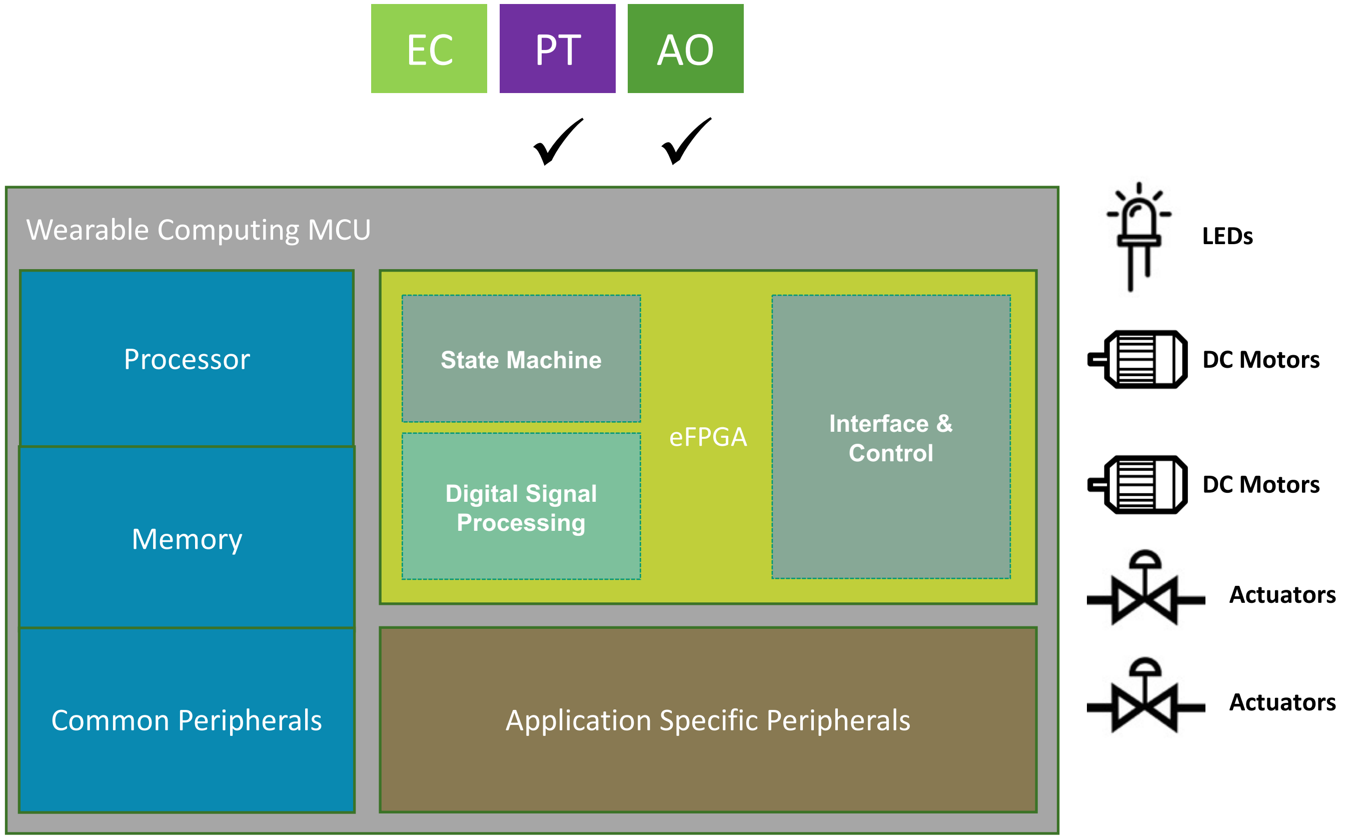
Sensor 2

Sensor 3

Sensor x

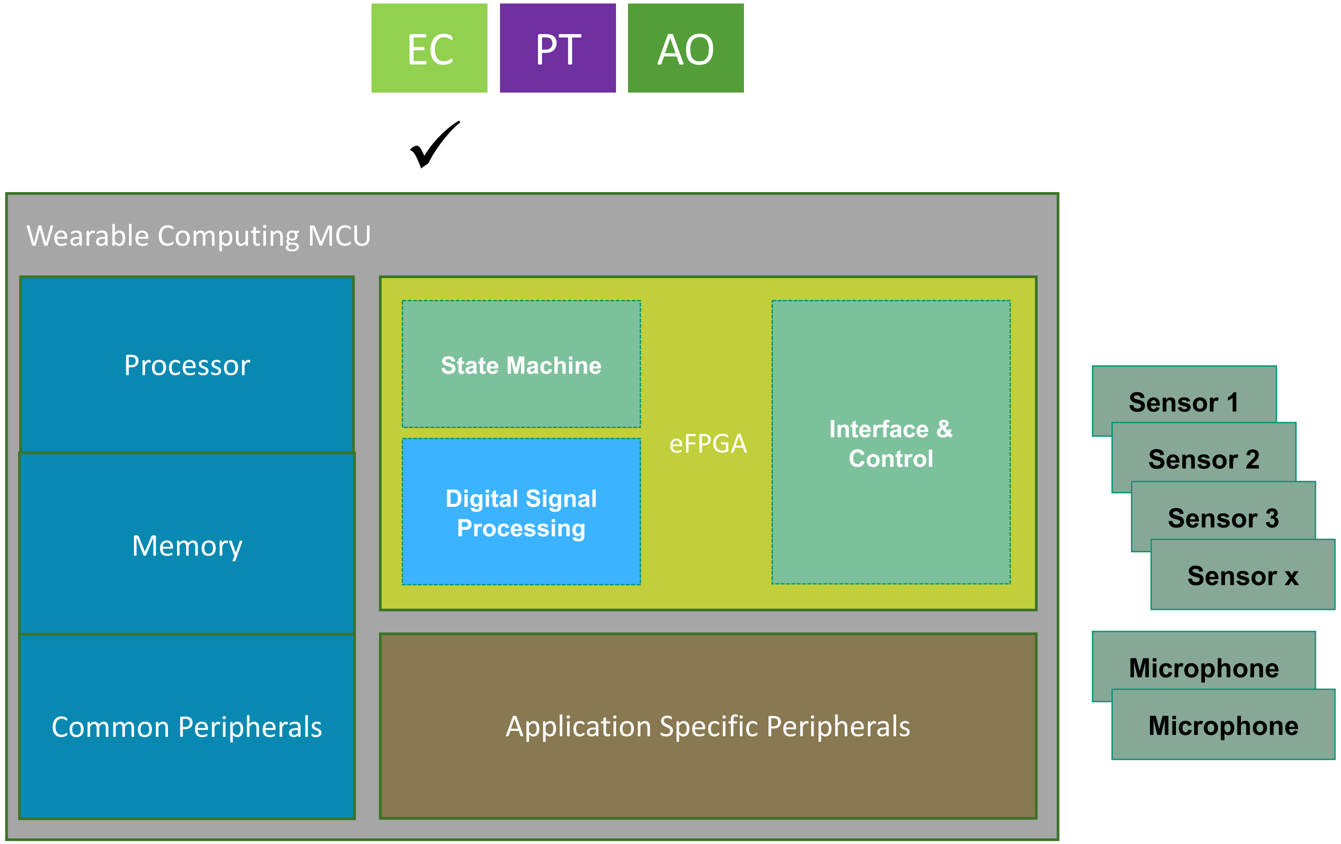
eFPGA Use Case – "Real Time" I/O Control

- Some system components need precisely controlled I/O timing to operate correctly
- Implementing this 'hard real time' with software can be challenging since processors are a shared resource
- With integrated eFPGA, precise I/O timing can be offloaded from the processor so that CPU loading is decoupled from I/O timing



eFPGA Use Case – Hardware offloaded/accelerated AI

- AI Inferencing that use neural networks tend to benefit from heterogeneous architectures that have parallel processing capability, particularly ones that can process millions of Multiply-Accumulate operations per second
- eFPGA with tightly coupled DSP blocks, and a direct path to an integrated general purpose processor are very efficient at implementing this architecture
- The benefit is a more platform with more computational capability (in terms of MACs) and flexibility to change the neural network implementation based on requirements



TensorFlow Lite

SensiML™