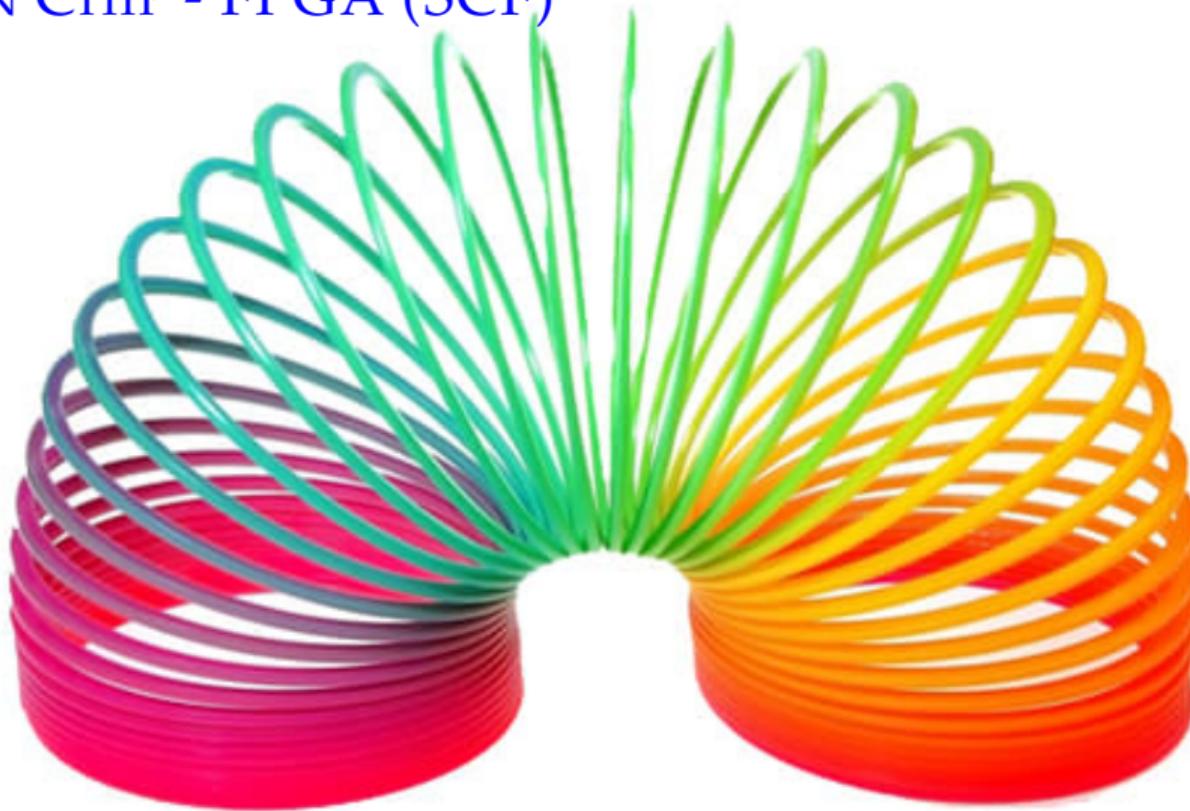


# SYSTEM ON CHIP - FPGA (SCF)



ALBERTO DASSATTI - YANN THOMA - 2024

# WHO

- ▶ Professor: Yann Thoma & Alberto Dassatti
- ▶ Office: A11 (Cheseaux)
- ▶ E-mail: [yann.thoma@heig-vd.ch](mailto:yann.thoma@heig-vd.ch) & [alberto.dassatti@heig-vd.ch](mailto:alberto.dassatti@heig-vd.ch)
- ▶ Telephone: +41 24 557 62 73 & +41 24 557 61 60

## YANN THOMA: PARCOURS

- ▶ 2001: Diplôme d'ingénieur en informatique de l'EPFL
- ▶ 2005: Doctorat EPFL (systèmes reconfigurables)
- ▶ 2005-2009: chargé d'enseignement à l'EIG (hepia) (systèmes numériques + systèmes logiques)
- ▶ 2006-2008: ingénieur pour le Groupe de Physique Appliquée de l'UniGe
- ▶ 2006-2008: ingénieur au REDS
- ▶ 2009- : Professeur au REDS
  - ▶ Cours BSL, CSF, PCO, PTR, VTF, VSN, VSE
  - ▶ Projets de recherche appliquée et développement, notamment:
    - ▶ FPGA: Math2mat, QCrypt, Bluetooth Low Energy Analyzer, analyseur USB3.0, génomique, switch PCIe3 déterministe
    - ▶ Informatique embarquée: ISyPeM, SpikeOnChip, OncoPok
    - ▶ Informatique: Tucuxi, TuberXpert
- ▶ 2015-2018 : Directeur du REDS
- ▶ 2018-2019 : Professeur visiteur à UNSW

2002: Master in Electronics Engineering, Politecnico di Torino

2003: Interim at Advanced System Technologies  
STMicroelectronics, Geneva

2004-2008 : Ph.D. in Telecommunications and Electronics  
(reconfigurable systems), Politecnico di Torino,

2008 : Visiting Ph.D. student UNSW Sydney, Australia

2008-2009: Post-Doc Researcher at VLSI Lab, Politecnico di Torino

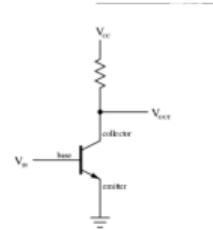
2003-2010: Entrepreneur fondateur of MicroC s.n.c., a Consultancy firm

2010-2012: ingénieur at NATO Undersea Research Centre

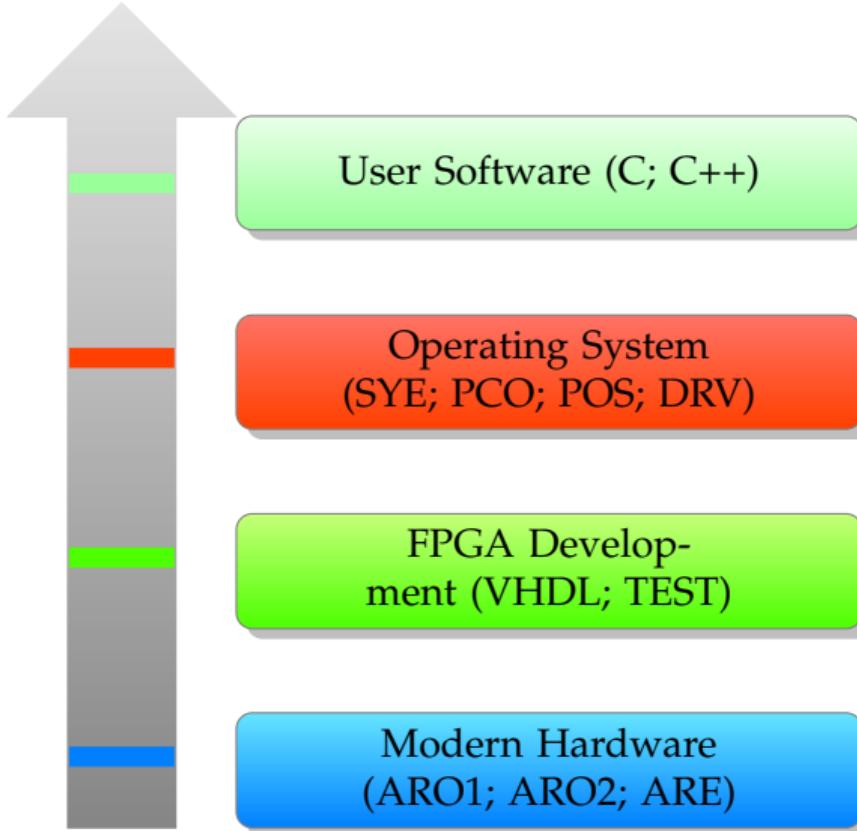
2012-2013: ingénieur au REDS

2013-: Professeur HES HEIG-VD

2018-: REDS' Director



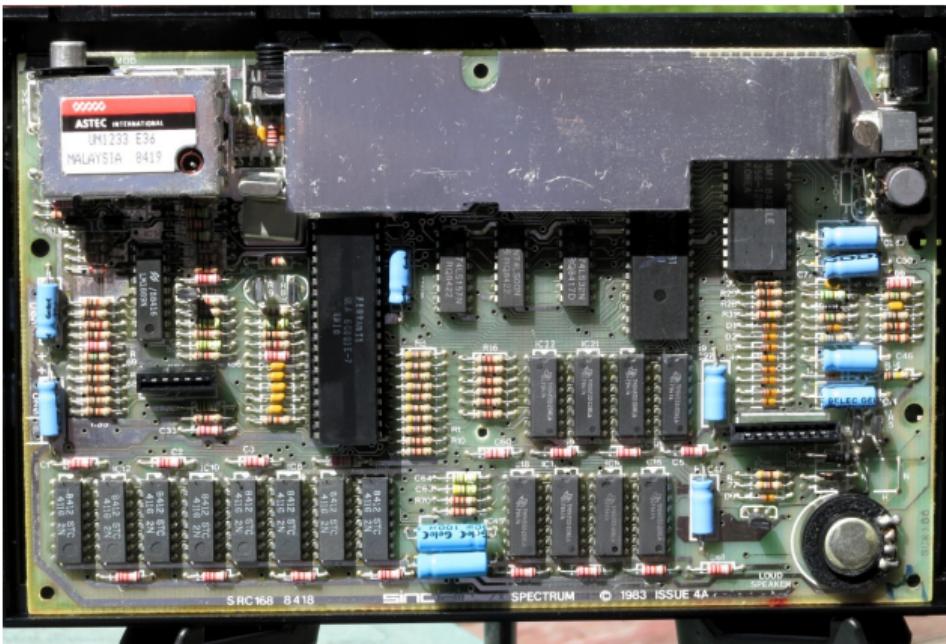
# WHAT YOU KNOW



## WHAT IS SCF GOAL?

Put it all together, really!

# ONCE UPON A TIME...



## TIMELINE<sup>1</sup>

- 1960: First MOSFETs (metal–oxide–semiconductor field-effect transistors) emerge. These basic elements, often referred to as gates in the FPGA world, act like switches. Depending on their input value (0 or 1), they allow or block the flow of current. MOSFETs play a crucial role in configuring FPGA logic;
- 1985: Xilinx, a pioneering FPGA company, releases the XC2064, the world's first FPGA. It features programmable logic blocks and interconnects, allowing users to configure custom digital circuits;

---

<sup>1</sup>All timelines are CoPilot Powered

## TIMELINE

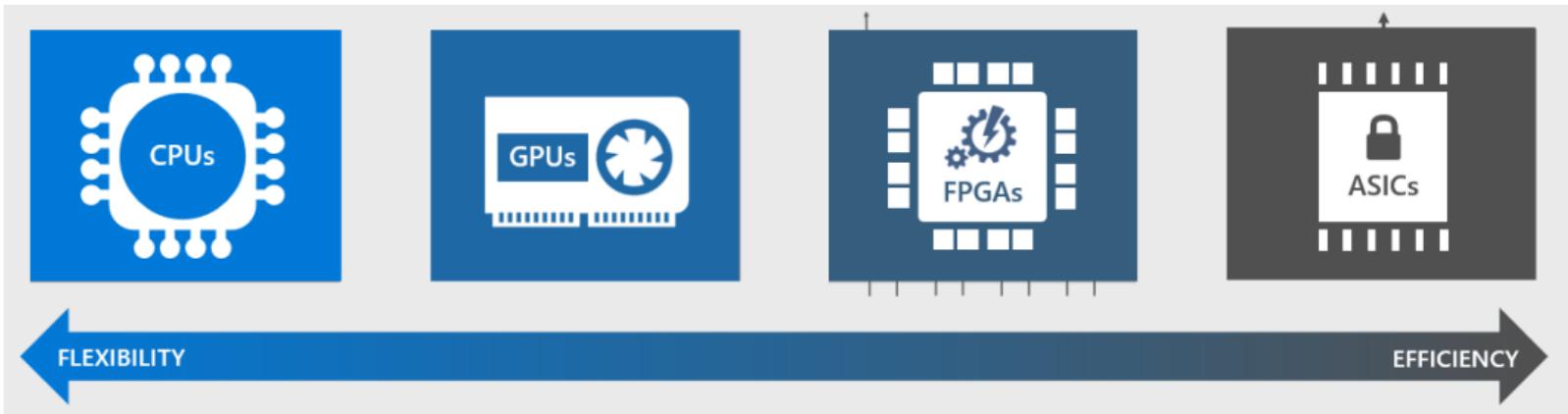
- 1990s: FPGAs gain popularity in various applications, including telecommunications, aerospace, and industrial automation;
- 2000s: System-on-Chip (SoC) FPGAs emerge, integrating FPGA fabric with processor cores, memory, and peripherals. This convergence enables complex embedded systems on a single chip;
- 2010s: FPGAs continue to evolve, with increased capacity, higher clock speeds, and improved power efficiency. Innovations like High-Level Synthesis (HLS) tools simplify FPGA programming by allowing developers to write code in higher-level languages;
- 2020s: ML!

## SOME MORE ACRONYMS

PLA: Programmable Array Logic. PAL is a subtype of PLA (Programmable Logic Array). It shares similarities with PLA but has a fixed OR plane, limiting the number of phrases that can be ORed together.

CPLD: Complex Programmable Logic Device. CPLDs resemble multiple PALs integrated onto a single chip, interconnected through a crosspoint switch.

# ALTERNATIVES



# WHERE CPUs AND FPGAs SHINE

CPUs

## WHERE CPUs AND FPGAs SHINE

### CPUs

- ▶ Fast
- ▶ Throughput oriented
- ▶ Fast
- ▶ Easy to program
- ▶ Used to be faster every day...

## WHERE CPUs AND FPGAs SHINE

### CPUs

### FPGAs

- ▶ Fast
- ▶ Throughput oriented
- ▶ Fast
- ▶ Easy to program
- ▶ Used to be faster every day...

## WHERE CPUs AND FPGAs SHINE

### CPUs

- ▶ Fast
- ▶ Throughput oriented
- ▶ Fast
- ▶ Easy to program
- ▶ Used to be faster every day...

### FPGAs

- ▶ Parallel
- ▶ Cycle accurate control
- ▶ Dataflow oriented
- ▶ Customizable
- ▶ Support data of any size (small is better)

## CPUs AND FPGAs LIMITS

CPUs

# CPUs AND FPGAs LIMITS

## CPUs

- ▶ Handling of short fast events
- ▶ Control
- ▶ Certifiable?
- ▶ Extremely Complex software stacks.
- ▶ Used to be faster every day...no more.

## CPU AND FPGAS LIMITS

### CPU

- ▶ Handling of short fast events
- ▶ Control
- ▶ Certifiable?
- ▶ Extremely Complex software stacks.
- ▶ Used to be faster every day...no more.

### FPGA

# CPU AND FPGAS LIMITS

## CPU

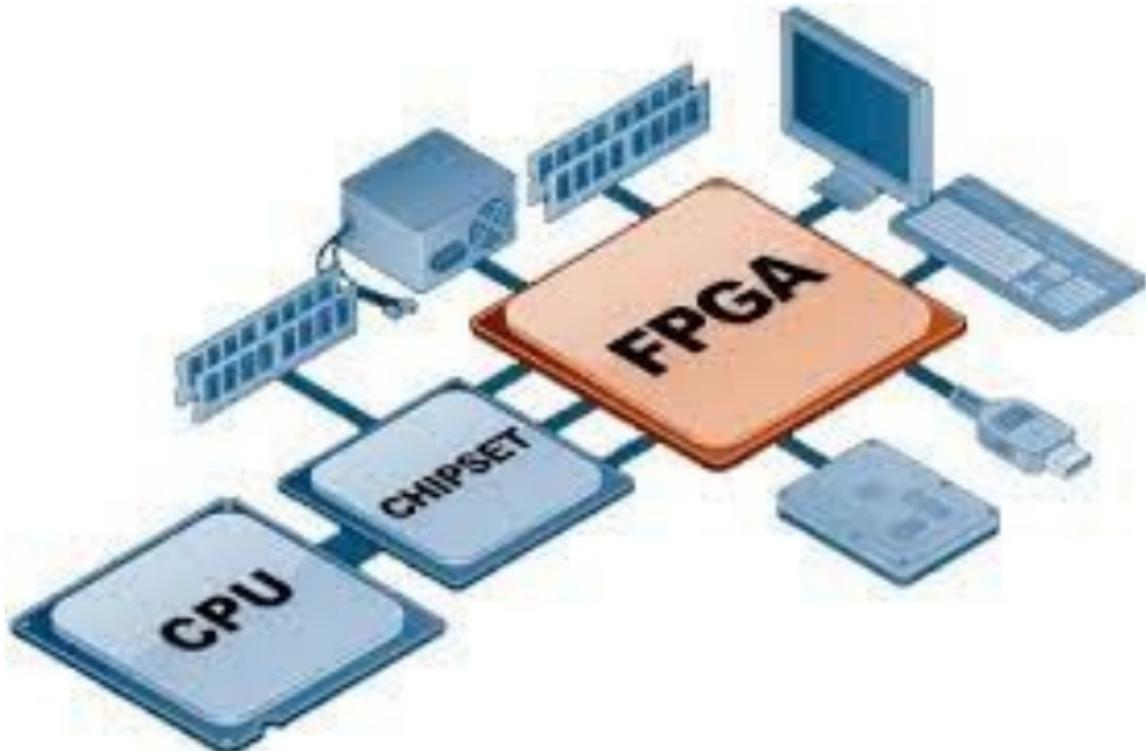
- ▶ Handling of short fast events
- ▶ Control
- ▶ Certifiable?
- ▶ Extremely Complex software stacks.
- ▶ Used to be faster every day...no more.

## FPGA

- ▶ Price
- ▶ Programming model and tools
- ▶ Custom boards
- ▶ Few programmers
- ▶ Floating point support is lacking...

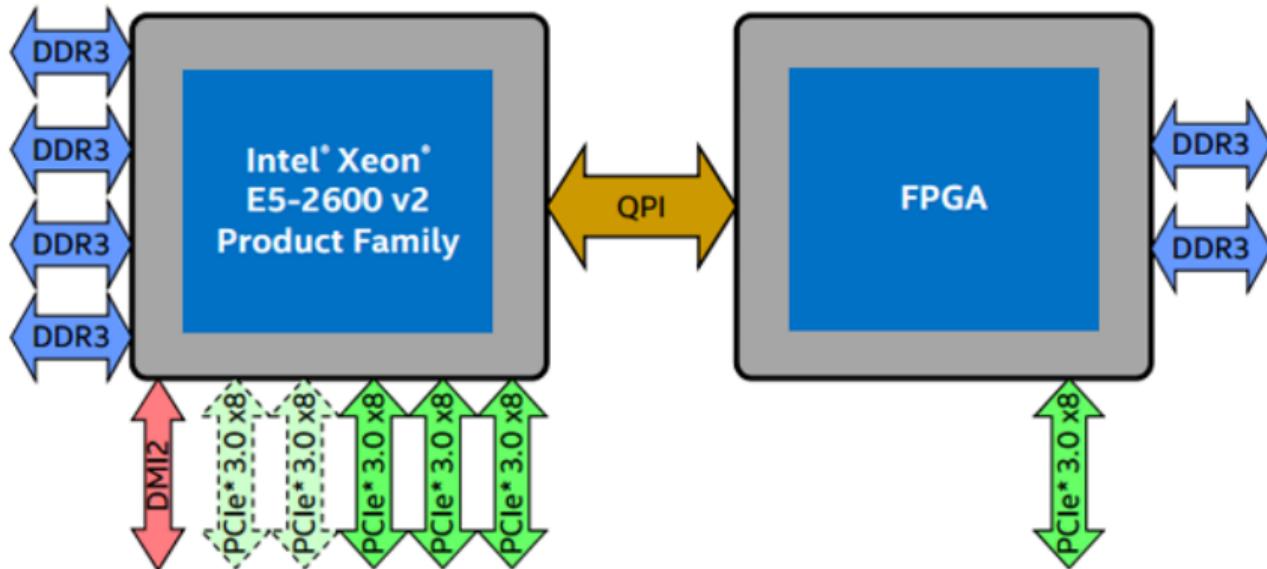
CIRCA 199X

In many cases, it makes a lot sense of combining the two.



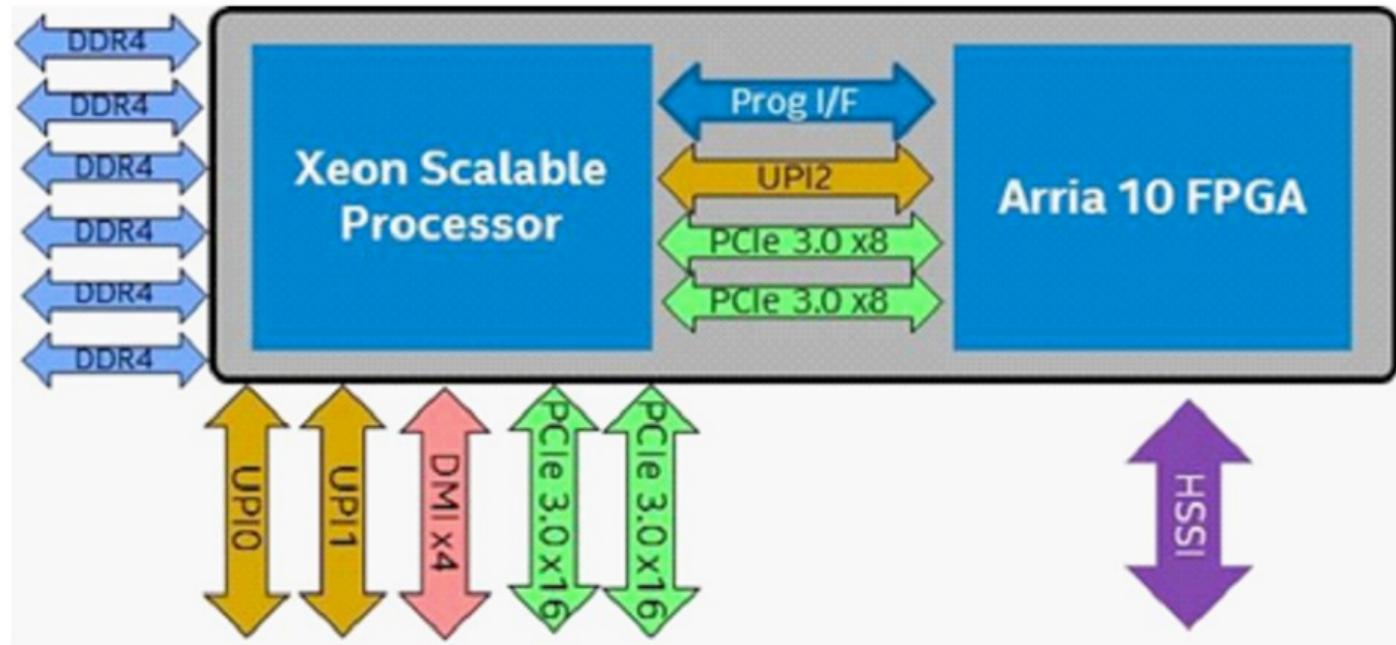
CIRCA 2010

First attempts to make a standard solutions



CIRCA 2016

And then to integrate them not only for the embedded world.



## SYSTEM-ON-CHIP (SoC) FPGAs

The best of the two worlds in a single chip!

## DIGRESSION: SOFT CORES

An FPGA is programmable, why not program in it a CPU<sup>2</sup>?

### Advantages

- ▶ Only one chip
- ▶ You can update CPU
- ▶ You can customize the CPU
- ▶ Many soft cores do exists: Nios II, uBlaze, Risc-V, Leon...
- ▶ you add the peripheral you need, no more.

### Drawbacks

- ▶ FPGA real estate is expensive
- ▶ Slow CPU clock
- ▶ Need DDR and code storage
- ▶ May need caches...
- ▶ Floating point support is lacking...

---

<sup>2</sup>ADI personal note: my master thesis in 2002

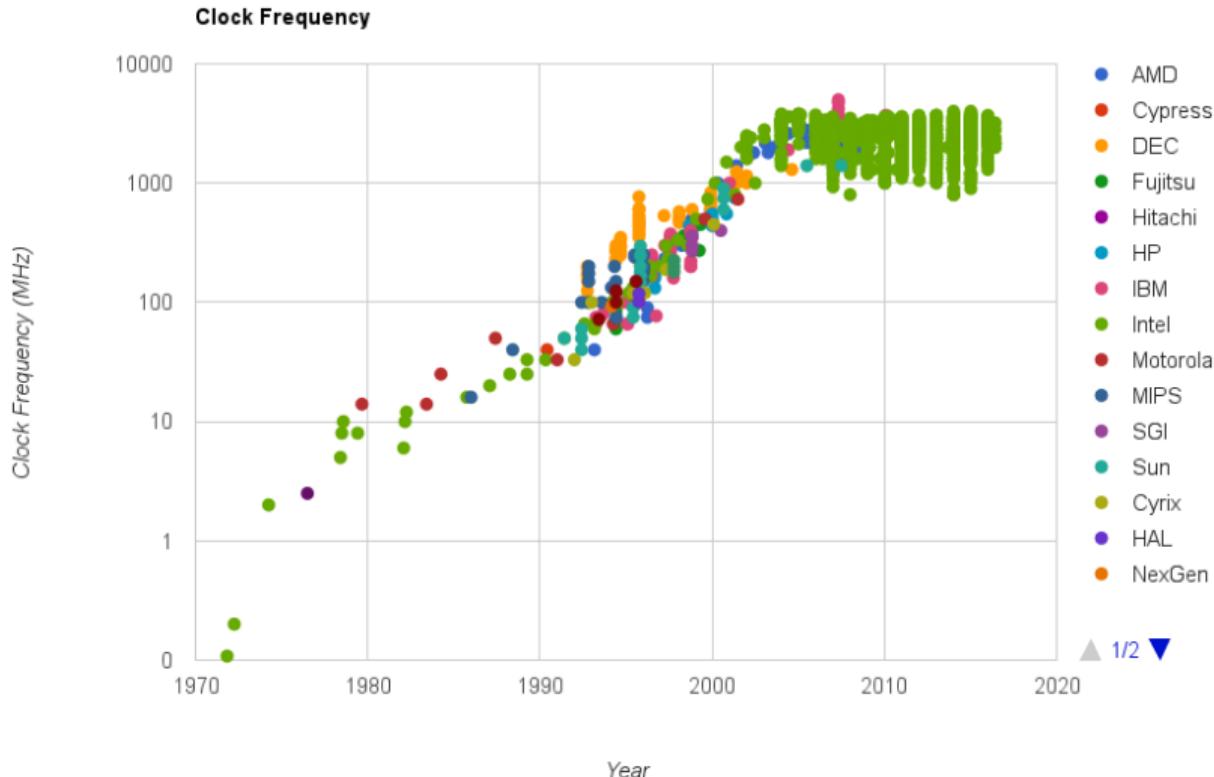
## TIMELINE

- 2000: Xilinx Virtex-II Pro: One of the earliest SoC FPGAs, integrating PowerPC processor cores with FPGA fabric. (ADI: a personal note: a nightmare to program) Altera (now Intel) Stratix: Introduced SoC variants, combining ARM cores with FPGA resources.
- 2005: Xilinx Zynq-7000: A SoC FPGA family featuring ARM Cortex-A9 cores alongside programmable logic. Widely adopted in embedded systems.
- 201x: Zynq UltraScale+ MPSoC: Xilinx's next-gen SoC FPGA with ARM Cortex-A53/A72 cores, GPU, and FPGA fabric. Used in automotive, 5G, and AI applications. Intel Cyclone V SoC: Dual-core ARM Cortex-A9 processor combined with FPGA logic on a single chip (your chip)

## TIMELINE

202x: Zynq UltraScale+ RFSoC: Integrates RF data converters, ARM cores, and FPGA fabric for wireless communication systems. Intel Agilex SoC FPGA: Combines Intel's 10nm FPGA fabric with ARM cores and HBM memory.

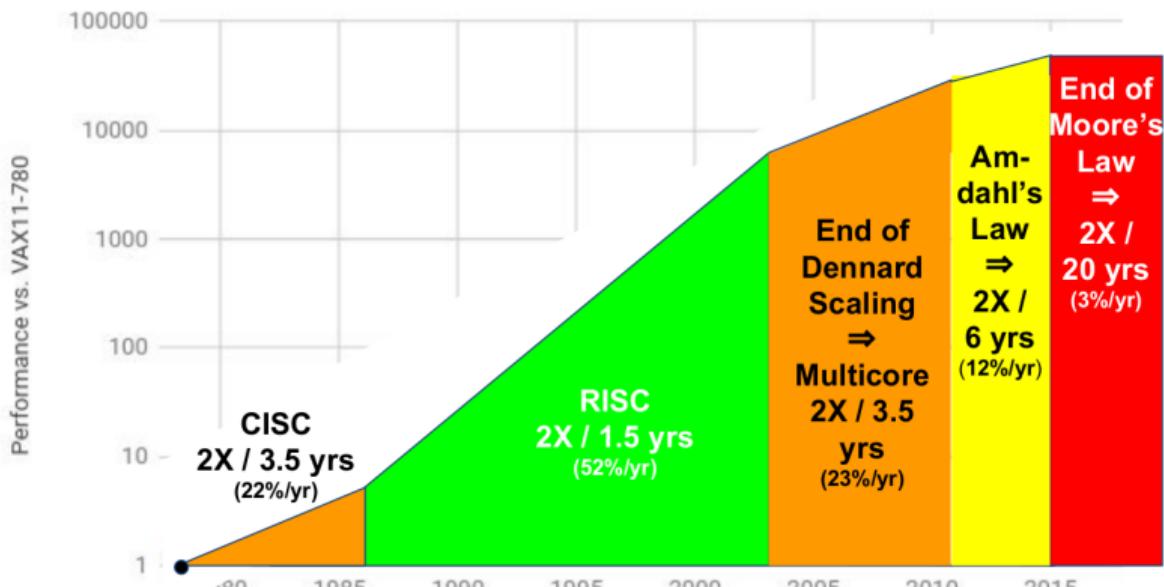
# DEEPER REASONING



[source]

# End of Growth of Performance?

## 40 years of Processor Performance



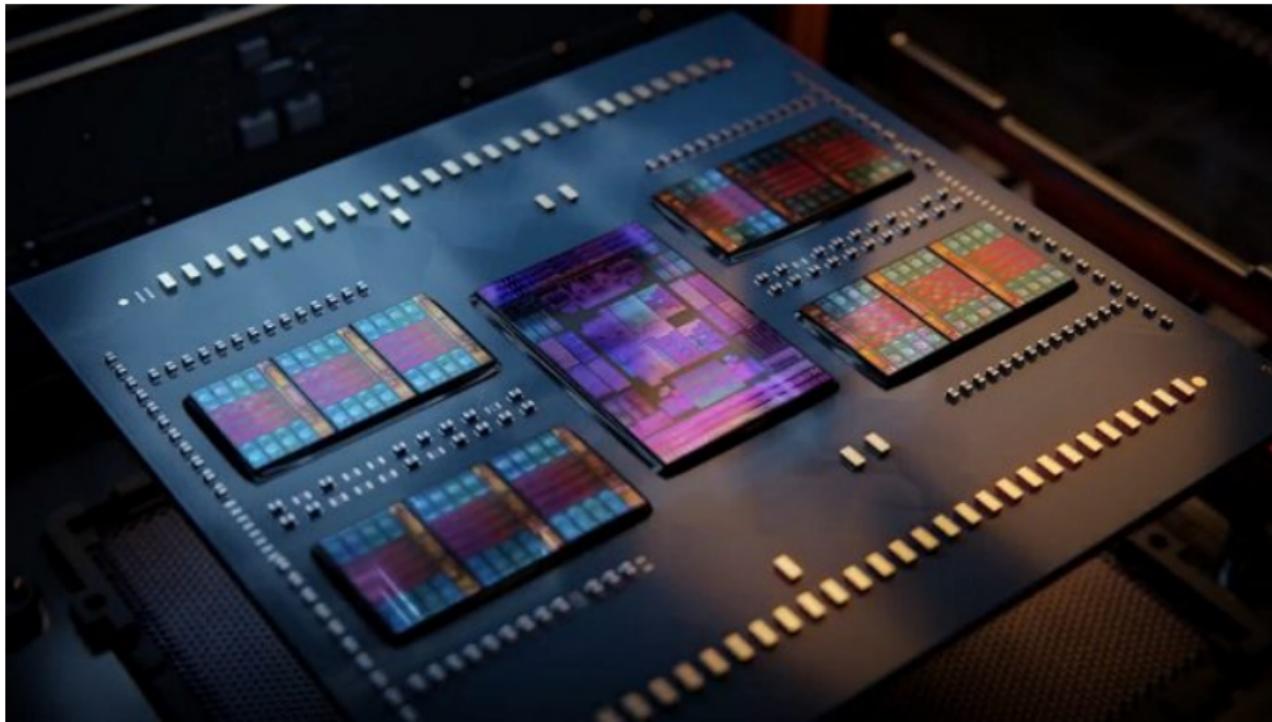
Based on SPECintCPU. Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e. 2018

[ Hennessy, Patterson 2018. ]

## WHAT'S NEXT

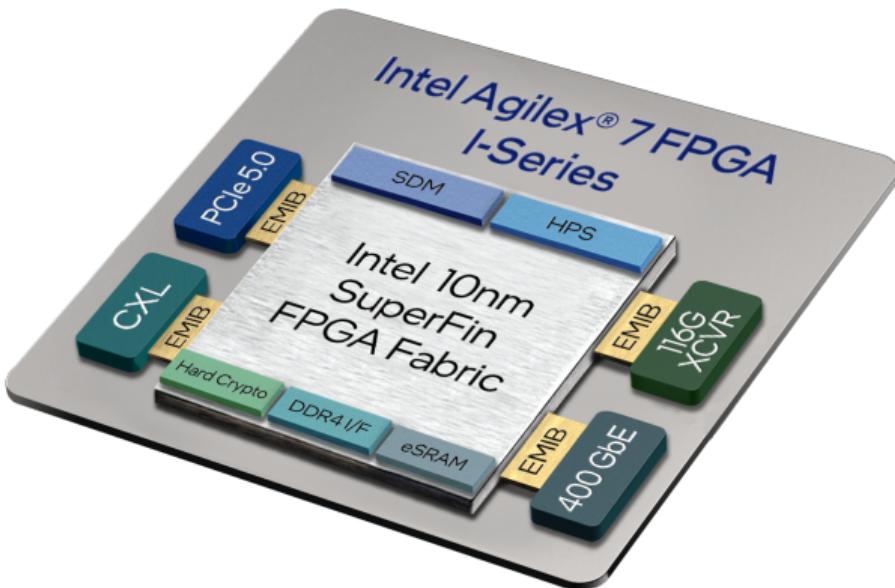
1. Chiplet
2. Specialization
3. Programming

# CHIPLET



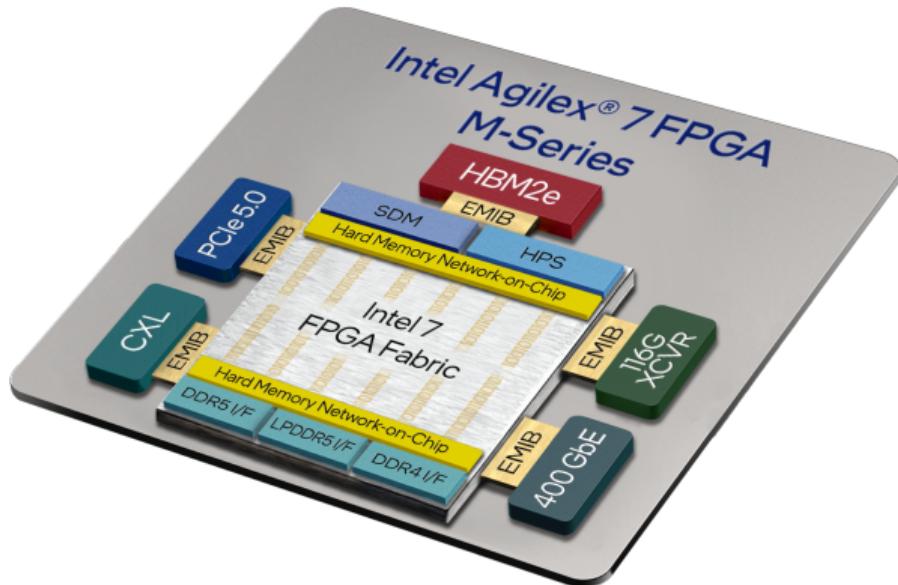
[source]

# CHIPLET



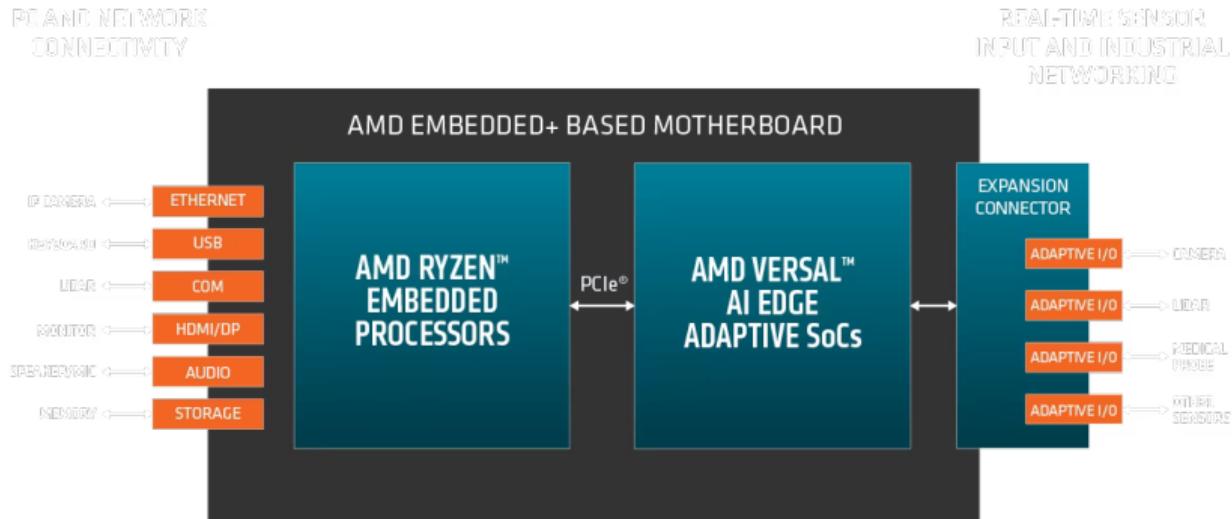
[source]

# CHIPLET



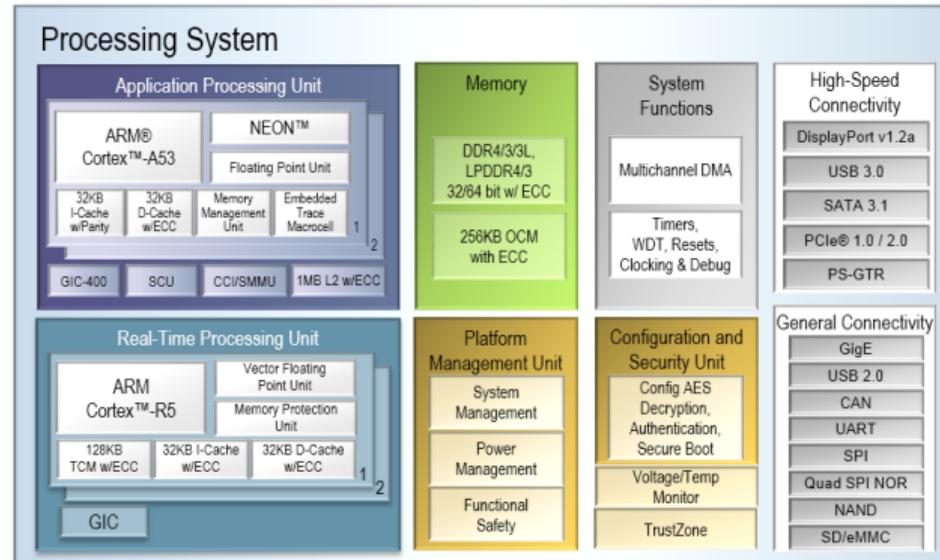
[source]

# CHIPLET FEBRUARY 2024

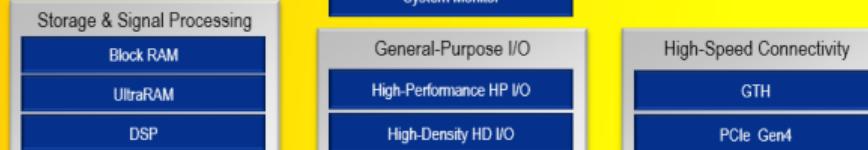


[source]

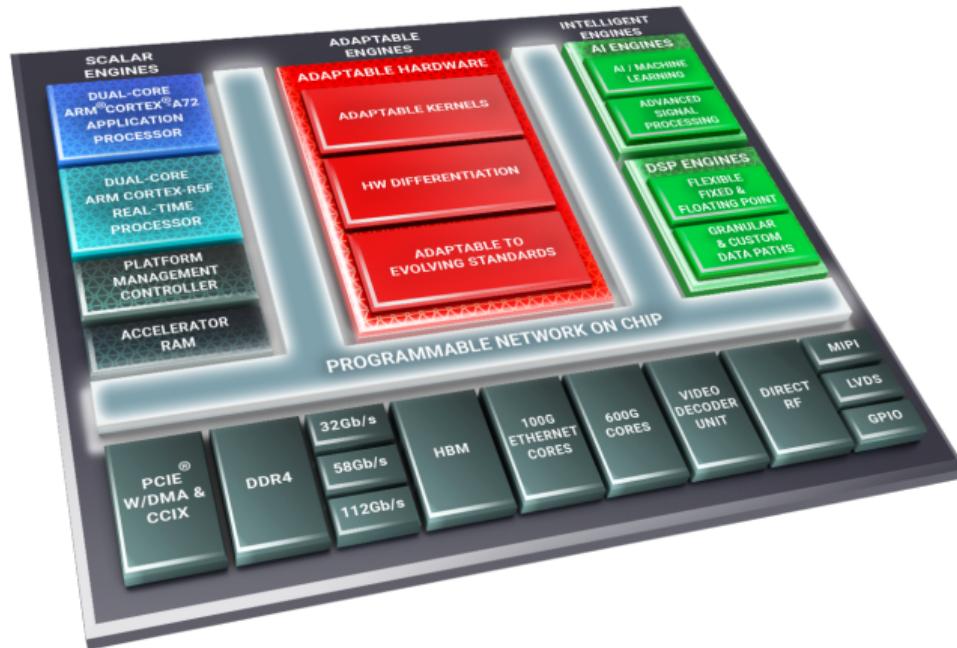
# SPECIALIZATION



## Programmable Logic



# SPECIALIZATION



[source]

# PROGRAMMING

Domain-Specific  
Development  
Environment

AI

Caffe

TensorFlow

Partner  
Development  
Environments

Vitis  
Accelerated  
Libraries



AI  
Models



Video  
Transcoding



Vision &  
Image



Data  
Analytics



Finance



Partner  
Libraries

Vitis  
Core  
Development Kit

Compilers

Analyzers

Debuggers

Xilinx Runtime library (XRT)

Vitis Target Platform

# WHERE IN THE SYSTEM?

- ▶ Interface
- ▶ Peripheral
- ▶ Central Unit
- ▶ DPU (Data Processing Unit):
  - ▶ Smart NIC
  - ▶ CSD
  - ▶ GPU helper

HOW



# HOW

- ▶ Theory presentations: 2h/w
- ▶ Lab work: 2h/w
- ▶ Your research and ideas
- ▶ Your personal effort (reading and watch video)
- ▶ Yann, Anthony and me are here to support and help (ask a meeting by email)
- ▶ It's all up to you

# WHAT YOU NEED

## to know

- ▶ System Architecture (the basics)
- ▶ Operating System (SYE)++
- ▶ Programming in C (very well)
- ▶ A little of assembly is useful
- ▶ Linux & Linux Drivers
- ▶ version control (git, ...)

## and...

- ▶ VHDL
- ▶ Adapt your coding style
- ▶ Investigate and relentless improving
- ▶ read, read, test, read, ask, test, read again

# IT IS HARD

- ▶ Very complex situations (each system is different)
- ▶ Hard to work in isolation, reproduce
- ▶ Custom tools
- ▶ Concurrency (SW and HW/SW)
- ▶ Few methodologies

# QUESTIONS

