The Abdus Salam International Centre for Theoretical Physics
www.ictp.it
Trieste, Italy

IAEA
International Atomic Energy Agency

Joint ICTP-IAEA School on Systems-on-Chip Based on FPGA for Scientific Instrumentation and Reconfigurable Computing

The Open Standard RISC-V Architecture

Fernando Rincón
University of Castilla-La Mancha
fernando.rincon@uclm.es

Agenda

- RISC-V short history
- What’s really RISC-V?
- RISC-V ISA Specification
- Open Source Hardware
- Licensing
- Roadmap & needs

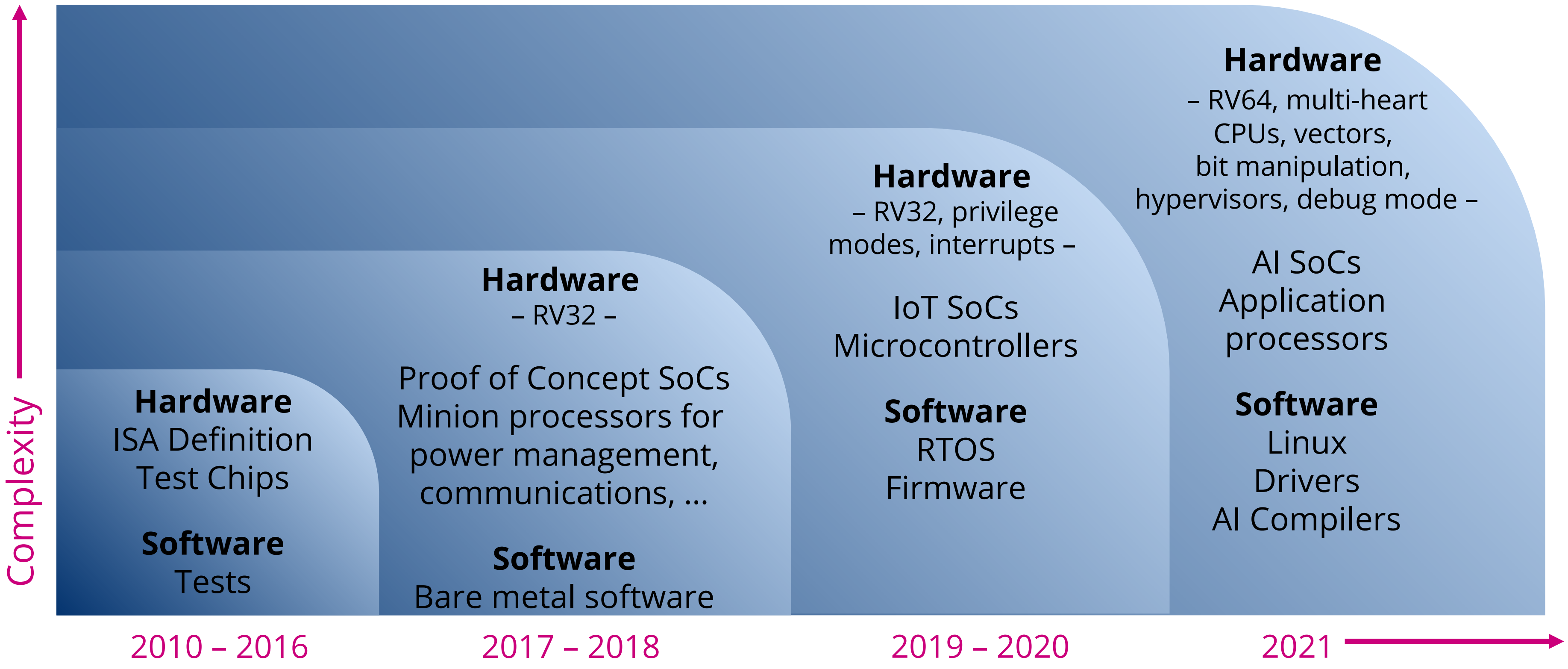
Little History

- Started as a 3-month project in 2010 I UC Berkerley
  - Purpose: design a simple ISA to be later extended
  - Why?
    - Commercial ISAs too complex
    - Royalties
- Now:

RISC-V foundation now > 230 members

Free, open, extensible ISA for all computing devices

Industry Innovation on RISC-V



Complexity

Hardware
ISA Definition
Test Chips

Software
Tests

2010 – 2016

Hardware
– RV32 –

Proof of Concept SoCs
Minion processors for
power management,
communications, …

Software
Bare metal software

2017 – 2018

Hardware
– RV32, privilege
modes, interrupts –

IoT SoCs
Microcontrollers

Software
RTOS
Firmware

2019 – 2020

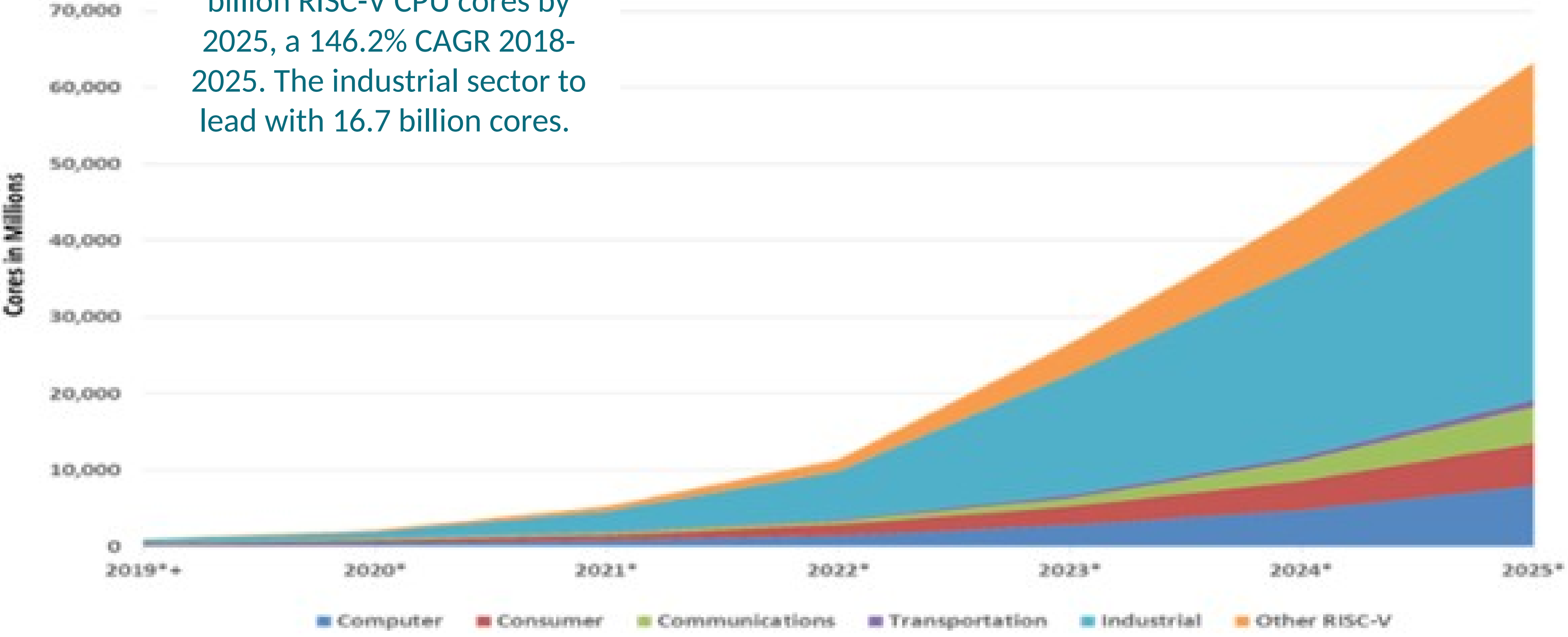
Hardware
– RV64, multi-heart
CPUs, vectors,
bit manipulation,
hypervisors, debug mode –

AI SoCs
Application
processors

Software
Linux
Drivers
AI Compilers

2021

Expected Growth



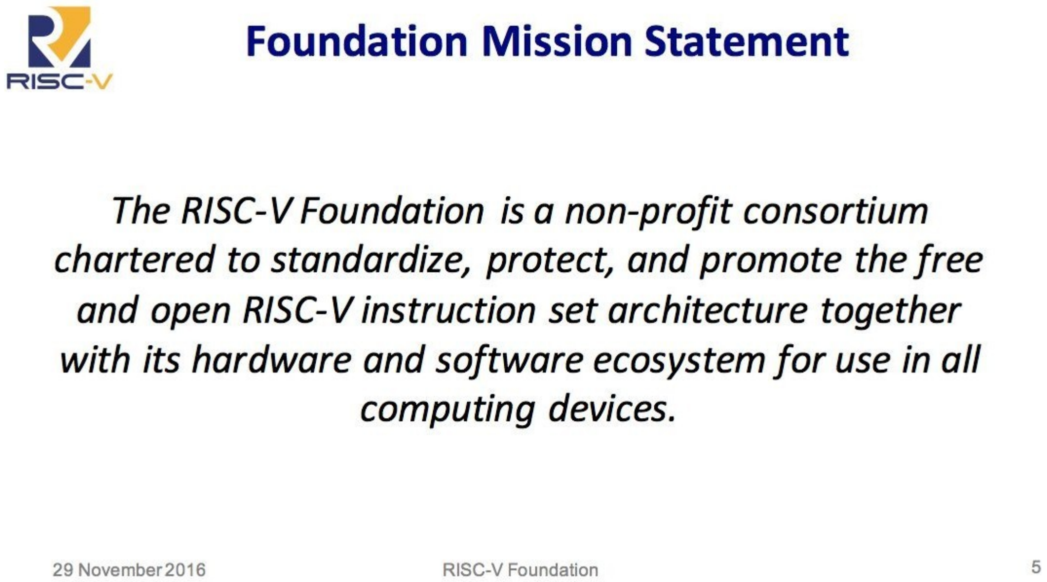
Semico Research predicts the market will consume 62.4 billion RISC-V CPU cores by 2025, a 146.2% CAGR 2018-2025. The industrial sector to lead with 16.7 billion cores.

What’s really RISC-V

- Royalties-free high quality RISC ISA
- Standard mantained by the RISC-V Foundation
- General-purpose wide-spectrum architecture
  - From microcontroller to supercomputers
- Available under a permissive license
- It’s not:
  - A company
  - A CPU implementation

RISC-V Foundation

- Non-profit organization founded in 2015
  - Responsible for the standard
  - Drives future evolution
  - Architecture verification
  - Promotion



RISC-V

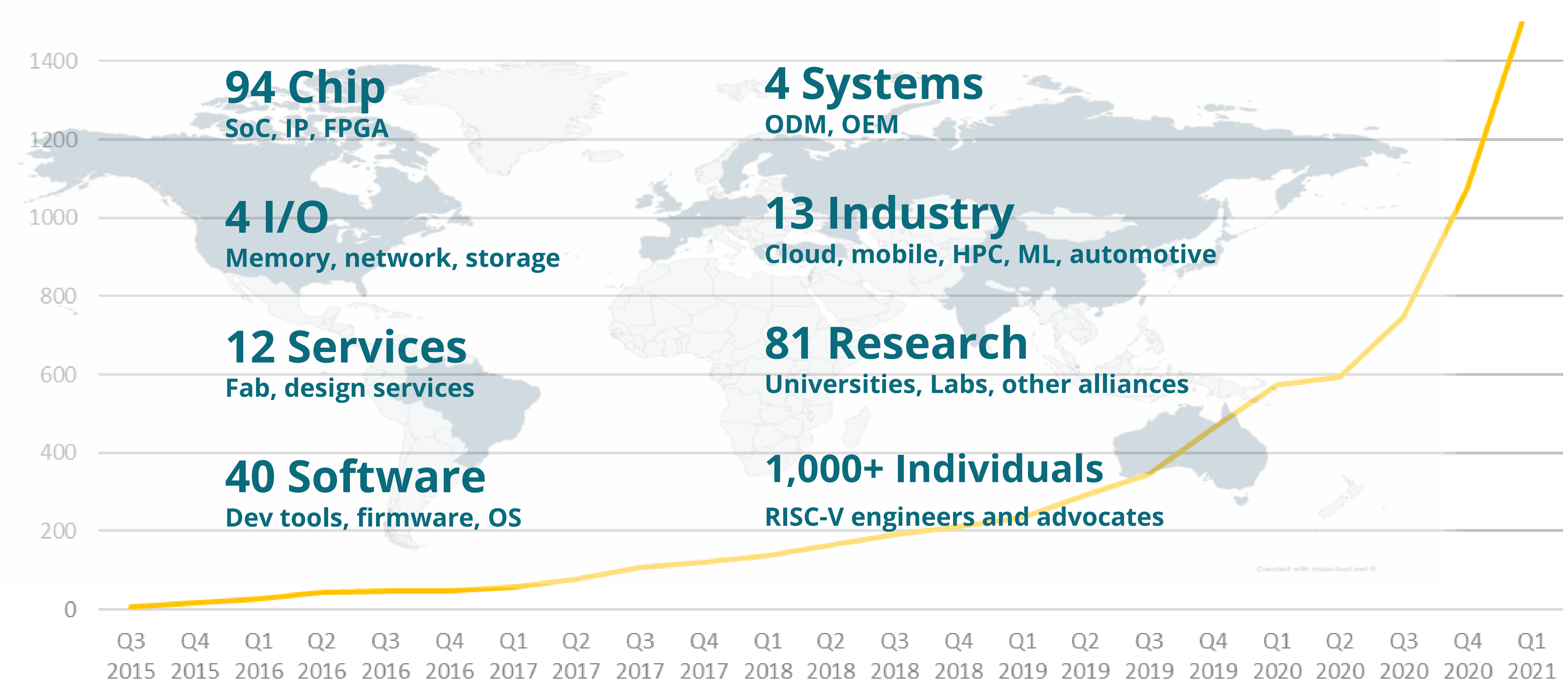
Foundation Mission Statement

The RISC-V Foundation is a non-profit consortium chartered to standardize, protect, and promote the free and open RISC-V instruction set architecture together with its hardware and software ecosystem for use in all computing devices.

29 November 2016 RISC-V Foundation 5

Members

Across 70 countries



94 Chip
SoC, IP, FPGA

4 I/O
Memory, network, storage

12 Services
Fab, design services

40 Software
Dev tools, firmware, OS

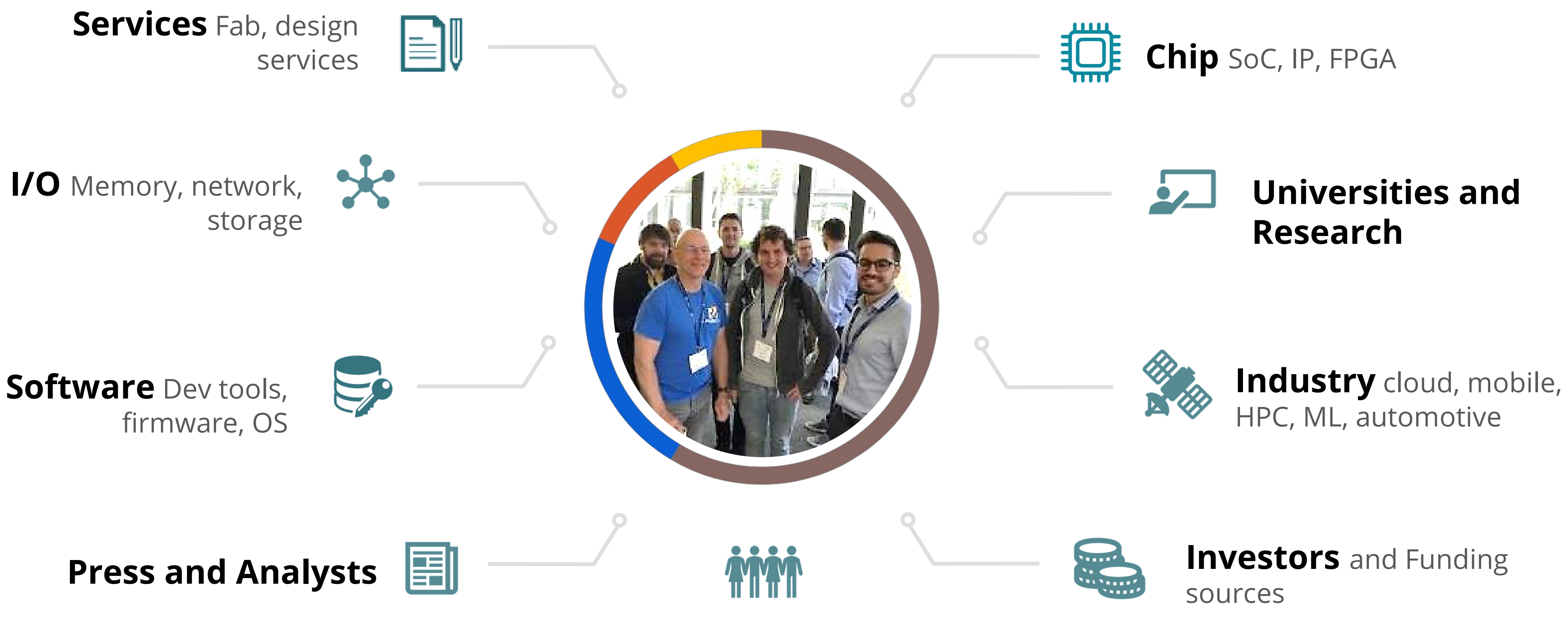
4 Systems
ODM, OEM

13 Industry
Cloud, mobile, HPC, ML, automotive

81 Research
Universities, Labs, other alliances

1,000+ Individuals
RISC-V engineers and advocates

Dedicated Community



Services Fab, design services

I/O Memory, network, storage

Software Dev tools, firmware, OS

Press and Analysts

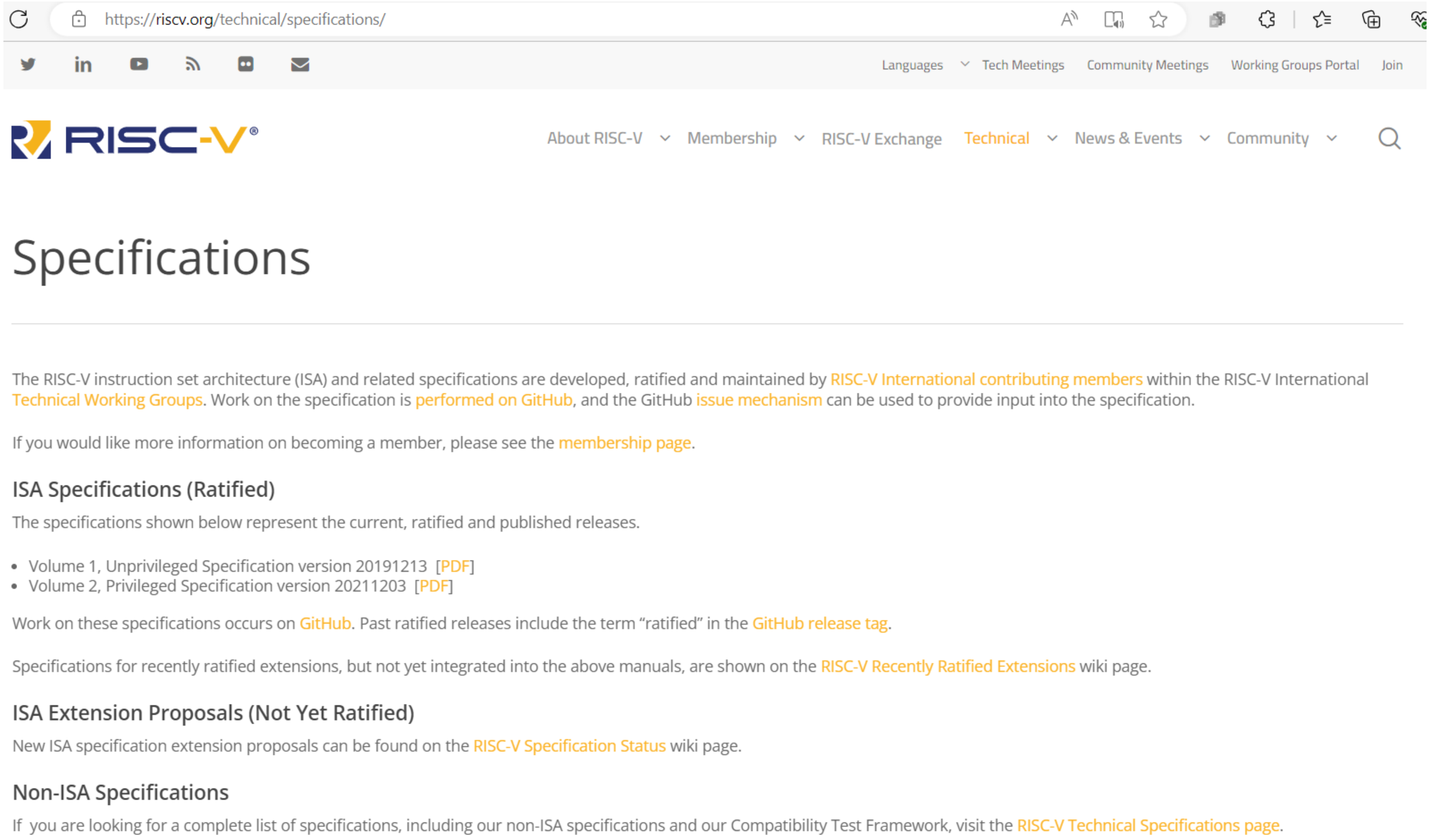
Chip SoC, IP, FPGA

Universities and Research

Industry cloud, mobile, HPC, ML, automotive

Investors and Funding sources

RISC-V ISA



Specifications

The RISC-V instruction set architecture (ISA) and related specifications are developed, ratified and maintained by RISC-V International contributing members within the RISC-V International Technical Working Groups. Work on the specification is performed on GitHub, and the GitHub issue mechanism can be used to provide input into the specification.

If you would like more information on becoming a member, please see the membership page.

ISA Specifications (Ratified)

The specifications shown below represent the current, ratified and published releases.

- Volume 1, Unprivileged Specification version 20191213 [PDF]
- Volume 2, Privileged Specification version 20211203 [PDF]

Work on these specifications occurs on GitHub. Past ratified releases include the term "ratified" in the GitHub release tag.

Specifications for recently ratified extensions, but not yet integrated into the above manuals, are shown on the RISC-V Recently Ratified Extensions wiki page.

ISA Extension Proposals (Not Yet Ratified)

New ISA specification extension proposals can be found on the RISC-V Specification Status wiki page.

Non-ISA Specifications

If you are looking for a complete list of specifications, including our non-ISA specifications and our Compatibility Test Framework, visit the RISC-V Technical Specifications page.

RISC-V ISA

- Follows a convention to describe different “flavours” for a specific implementation
- ISA naming format:
  - RV[###][abc.....xyz] RV: Implies RISC-V architecture
  - [###]: {32, 64, 128}
    - width of integer register file
    - with of the user address space
  - [abc...xyz]: Describes de set of extensions included

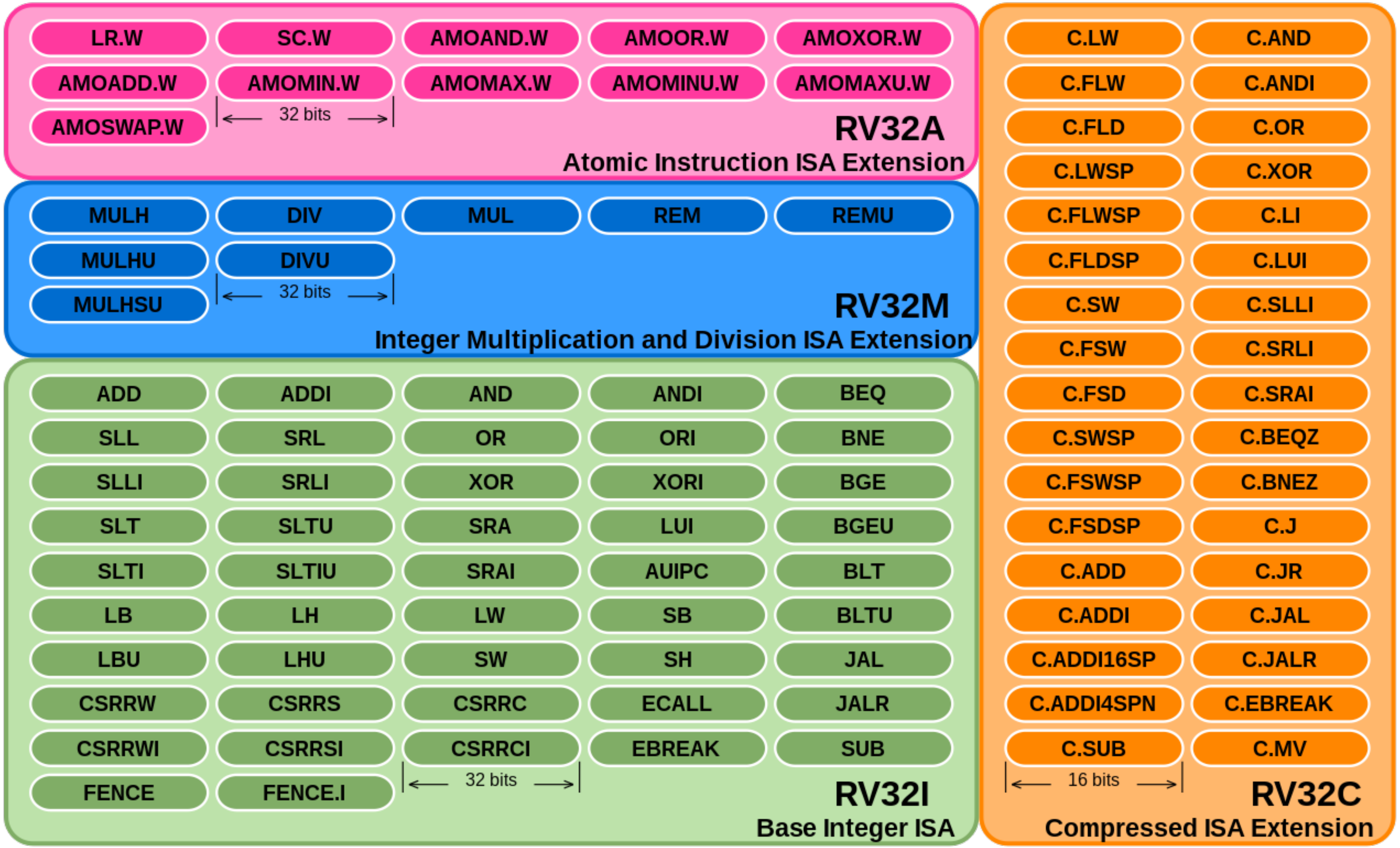
RISC-V Extensions

- Extensions define instructions: as extensiones definen las instrucciones.
  - "I" (Integer) only required extension. Defines 47 different instructions
  - RISC-V spec. defines several optional “standard extensions”
  - RISC-V allow user-specific non-standard extensions
- Examples:
  - RV32I: Most basic RISC-V
  - RV32IMAC:
    - Integer + Multiply + Atomic + Compressed
  - RV64GC: 64 bits IMAFDC de

Code	Extension
I	Integer
M	Integer Multiplication and Division
A	Atomics
F	Single-Precision Floating Point
D	Double-Precision Floating Point
G	General Purpose = IMAFD
…	…
X	Non-standard extension

RISC-V Extensions

RV32IMAC



LR.W	SC.W	AMOAND.W	AMOOR.W	AMOXOR.W
AMOADD.W	AMOMIN.W	AMOMAX.W	AMOMINU.W	AMOMAXU.W
AMOSWAP.W	← 32 bits →			

RV32A
Atomic Instruction ISA Extension

MULH	DIV	MUL	REM	REMU
MULHU	DIVU			
MULHSU	← 32 bits →			

RV32M
Integer Multiplication and Division ISA Extension

ADD	ADDI	AND	ANDI	BEQ
SLL	SRL	OR	ORI	BNE
SLLI	SRLI	XOR	XORI	BGE
SLT	SLTU	SRA	LUI	BGEU
SLTI	SLTIU	SRAI	AUIPC	BLT
LB	LH	LW	SB	BLTU
LBU	LHU	SW	SH	JAL
CSRRW	CSRRS	CSRRC	ECALL	JALR
CSRRWI	CSRRSI	CSRRCI	EBREAK	SUB
FENCE	FENCE.I	← 32 bits →		

RV32I
Base Integer ISA

C.LW	C.AND
C.FLW	C.ANDI
C.FLD	C.OR
C.LWSP	C.XOR
C.FLWSP	C.LI
C.FLDSP	C.LUI
C.SW	C.SLLI
C.FSW	C.SRLI
C.FSD	C.SRAI
C.SWSP	C.BEQZ
C.FSWSP	C.BNEZ
C.FSDSP	C.J
C.ADD	C.JR
C.ADDI	C.JAL
C.ADDI16SP	C.JALR
C.ADDI4SPN	C.EBREAK
C.SUB	C.MV
← 16 bits →	

RV32C
Compressed ISA Extension

RISC-V Register File

Registro	ABI Name	Description	Saver
x0	Zero	Hard-wired zero	-
x1	ra	Return address	Caller
x2	sp	Stack Pointer	Callee
x3	gp	Global Pointer	-
x4	tp	Thread pointer	-
x5-7	t0-2	Temporaries	Caller
x8	s0/fp	Saved register/Frame pointer	Callee
x9	s1	Saved register	Callee
x10-11	a0-1	Function Arguments/return values	Caller
x12-17	s2-7	Function arguments	Caller
x18-27	s2-11	Saved registers	Callee
x28-31	t3-6	Temporaries	Caller

- RV32I/64I have 32 integer registers
- 32 optional floating point registers
- RV32E is a reduced version for embedded with just 16 registers
- Width determined by ISA.
- Application Binary Interface (ABI) de RISC-V defines standard functionality
- Enables Sw interoperability

RISC-V Addressing Modes

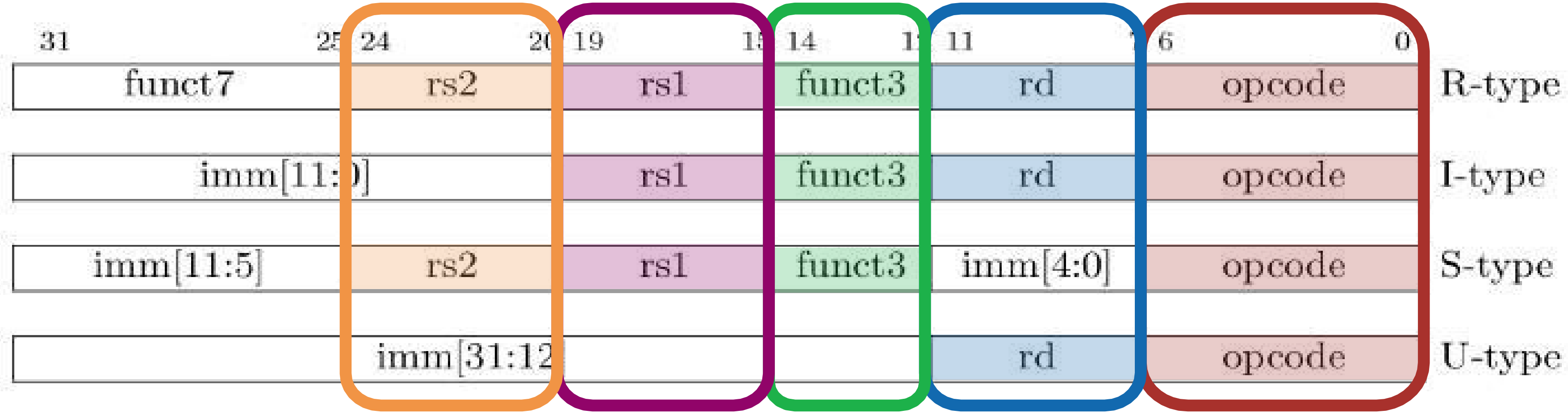
- Privileges spec. in RISC-V defines 3 levels (modes)
- Machine Mode is the highest and the only required
- Very flexible. Adaptable to the range from microcontrollers to high-performance processors.
- Machine, Hypervisor and Supervisor have own Control and State Registers (CSRs).

RISC-V Modes		
Level	Name	Abbr.
0	User/Application	U
1	Supervisor	S
2	Hypervisor	HS
3	Machine	M

Supported Combinations of Modes	
Supported Levels	Modes
1	M
2	M, U
3	M, S, U
4	M, HS, S, U

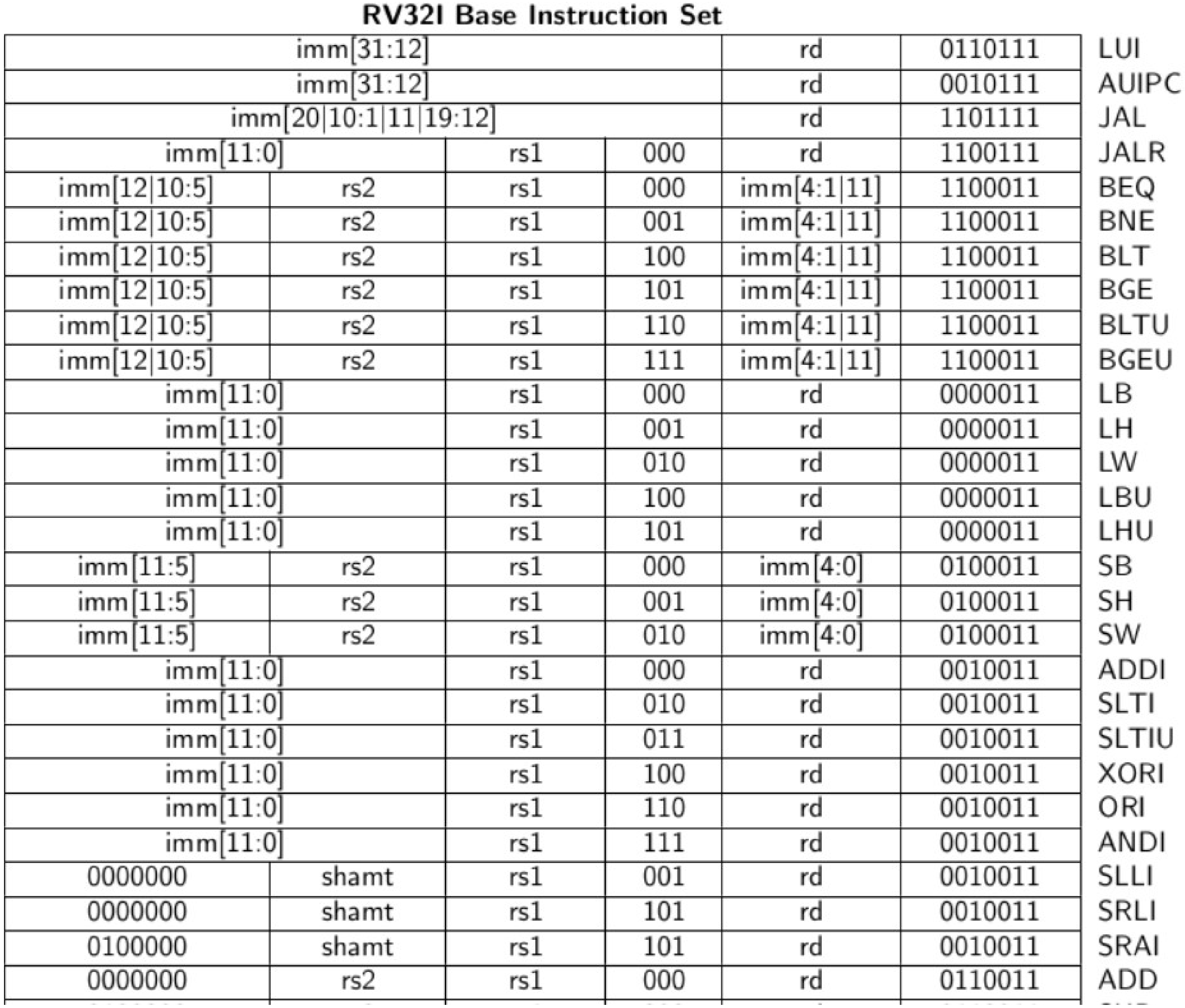
Basic Instructions format

- R
  - Register to register instructions
- I
  - Operations with constant and/or immediate values
- S/SB
  - Operations with 2 source registers
- U/UJ
  - Operations with large constant and/or immedites



31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type
imm[11:0]		rs1	funct3	rd	opcode	I-type
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-type
imm[31:12				rd	opcode	U-type

Basic Instructions Coding

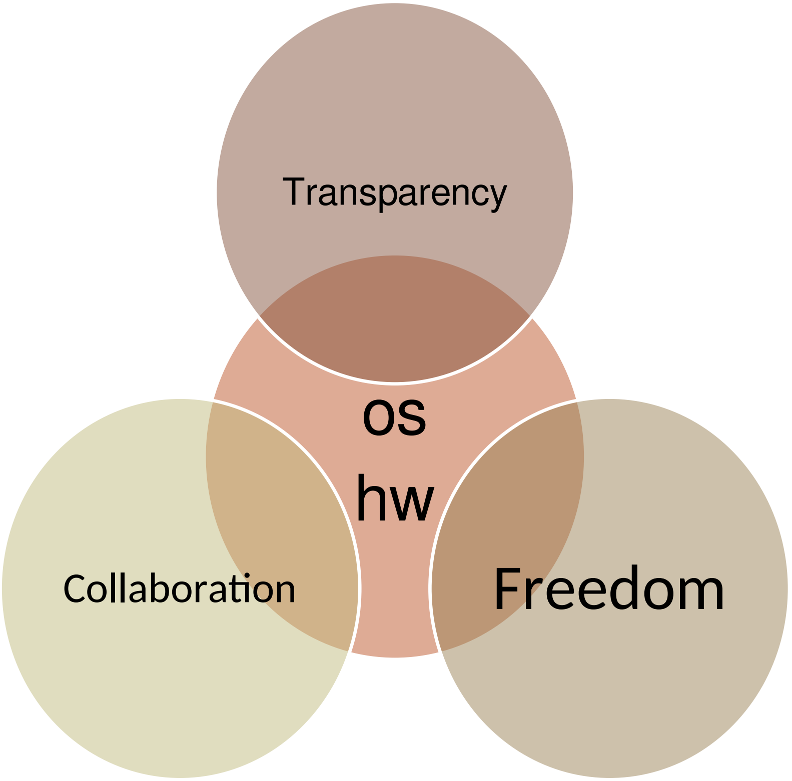


RV32I Base Instruction Set

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD

Open Hardware

- Design of physical object, such as electronic circuits, development boards and/or devices
- Layouts, specifications, schematics and interfaces are public
- They can be freely studied, modified, used and distributed



open source hardware

osHW Benefits

- Accelerated innovation:
  - Open collaboration and feedback promote fast improvement and novel solutions
- Accesibility:
  - Easier access to technology
  - Lowers the barrier to feasible product design
- Transparency & trust:
  - User have access to the code
  - Code can be understood
  - Code can be audited
  - This generates trust

osHW is not just ...

- public sharing. Need for:
  - Maintenaince & support
  - Documentation
  - Avoid effort fragmentation
  - Governance:
    - Balance control between different stateholders

Courtesy of rawpixel.com in Freepik

Main osHw initiatives

RISC-V®

CHIPS ALLIANCE

skywater

GROUP OPENHW®
PROVEN PROCESSOR IP

FOSSi Foundation

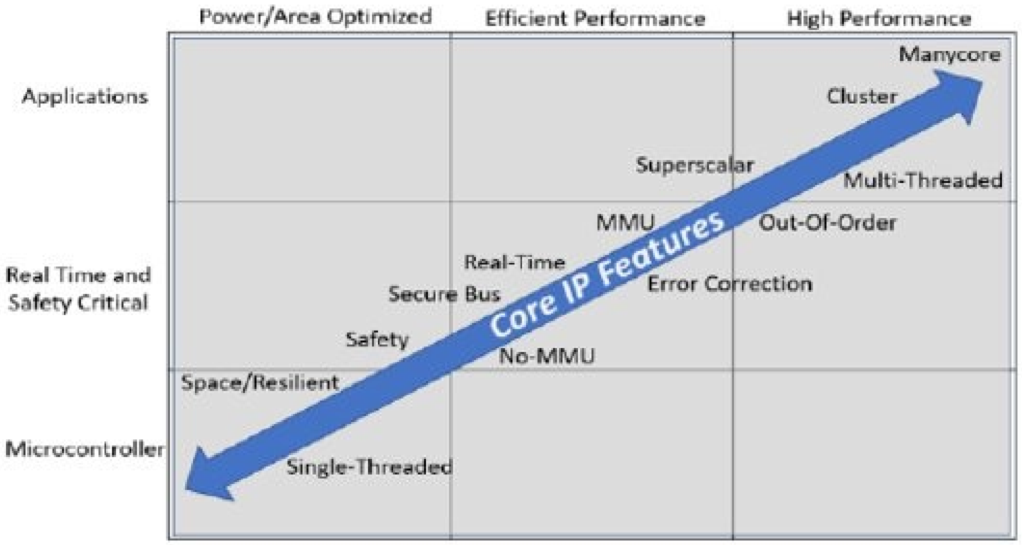
EUROPRACTICE

Most common licensing mechanisms

- **Creative Commons Zero (CC0)**: Allows unrestricted use and waives all copyright rights. The hardware is placed in the public domain.
- **Creative Commons Attribution-ShareAlike (CC-BY-SA)**: Allows the use, modification, and distribution of the hardware, provided that credit is given to the author and modified versions are shared under the same license.
- **GNU General Public License (GPL)**: De-facto standard. Requires that any improvements or modifications to the hardware be shared with the same GPL license. LGPL is the weakly reciprocal version.
- **CERN Open Hardware License (CERN OHL)**: Based on the copyleft principle, it requires that any redistribution of the hardware, along with its modifications, remains open source. There are three variants: permissive, weakly reciprocal/strongly reciprocal.
- **Apache License 2.0**: Can be used in permissive contexts for open hardware projects.

RISC-V Roadmap

- Each need has different requirements:
  - Size & cost
  - Energy efficiency
  - Performance
- Objetive:
  - Provide different types of processors
  - RISC-V, beyond RISC-V, ultra low-power, high-end
  - Different types of profiles
  - Solutions highly adaptable & configurable
- Need for the development of a large number of IP Cores

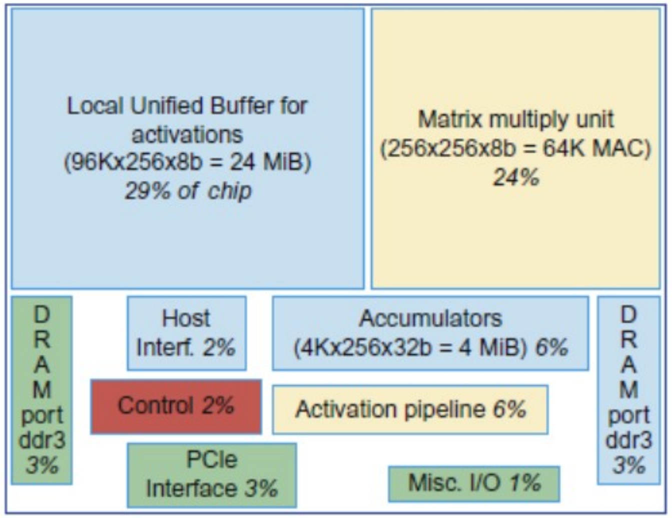


RISC-V Core IP Roadmap

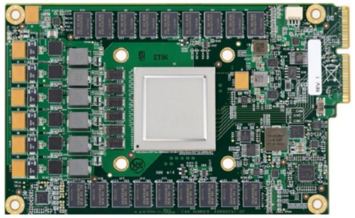
OpenSource Hw & Sw Working Group

Domain Specific Architectures

- Evenmore, CPUs are not the only processors
- CPUs are convenient if:
  - Memory access patterns are predictable (cache)
  - Reasonable balance between control vs computation
  - Execution of different types of tasks
- End of Moore’s law -> need for optimization
  - Specific processors/accelerators
  - More convenient of other types of data access patterns
  - Optimized datapaths for specific computations
    - Ej: matrix/vectorial operations for deep learning


Local Unified Buffer for activations (96Kx256x8b = 24 MiB) *29% of chip*
Matrix multiply unit (256x256x8b = 64K MAC) *24%*
DRAM port ddr3 3%
Host Interf. 2%
Accumulators (4Kx256x32b = 4 MiB) 6%
DRAM port ddr3 3%
Control 2%
Activation pipeline 6%
PCIe Interface 3%
Misc. I/O 1%

Google TPU



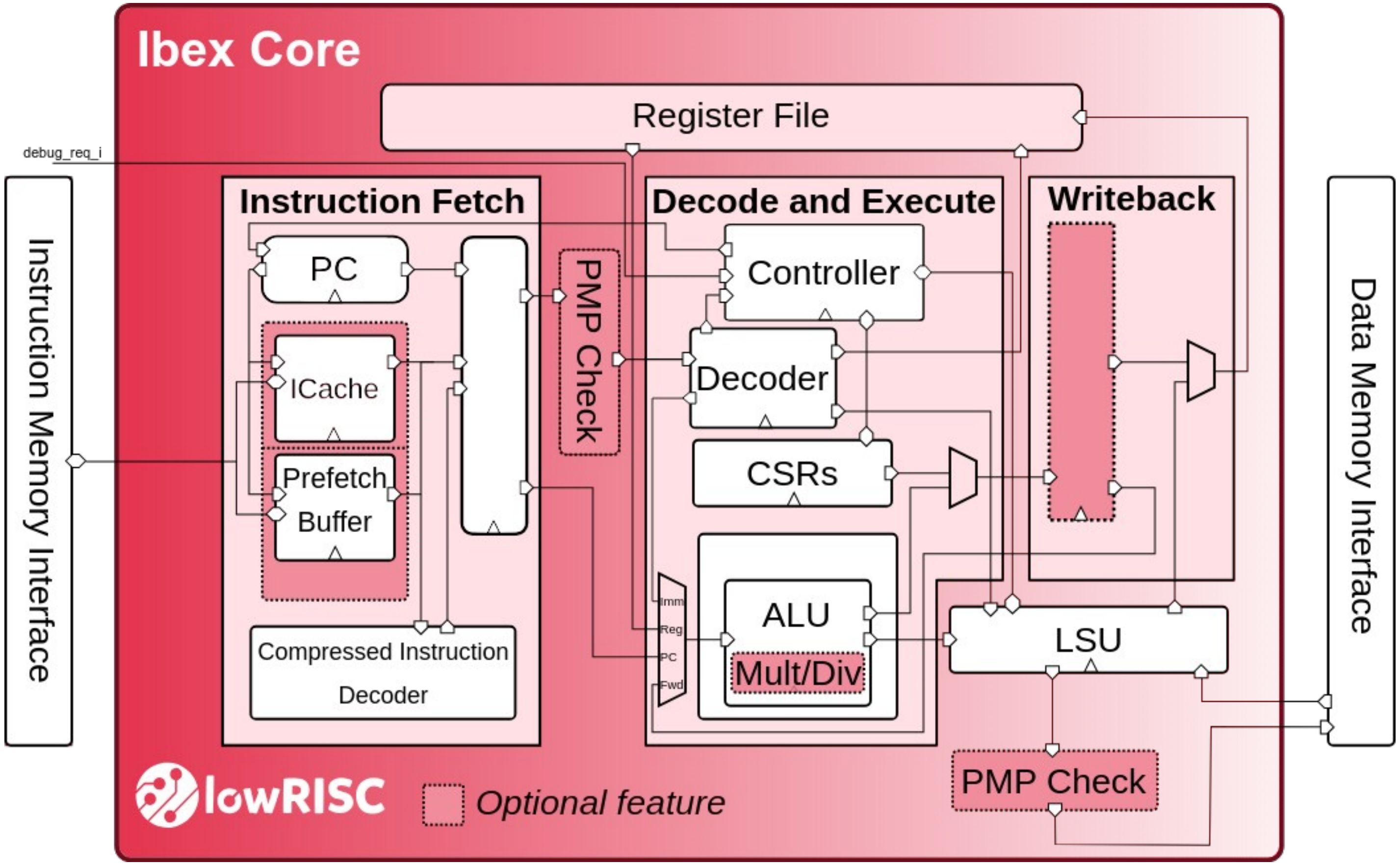
Custom Accelerators

- Domain specific architectures are complemetary to the CPU
- They provide acceleration for specific tasks
  - The rest handled by the CPU
- Many types depending on the purpose
  - Digital Signal Processing
  - Deep learning
  - 3D Vision for self-driving cars
  - Post-quantum cryptografy
  - ...
- Implemented as ASICs or using FPGAs

- Integration needs
  - Definition of CPU→accel interface
    - Reusable solutions
    - Specific but standarized buses
    - Communication IP Cores
  - Memory:
    - High speed / high bandwidth buses
  - CPU
    - Selection of the right processor
    - Specific extensions for the ISA
  - SW:
    - O.S. drivers
    - Libraries for user applications

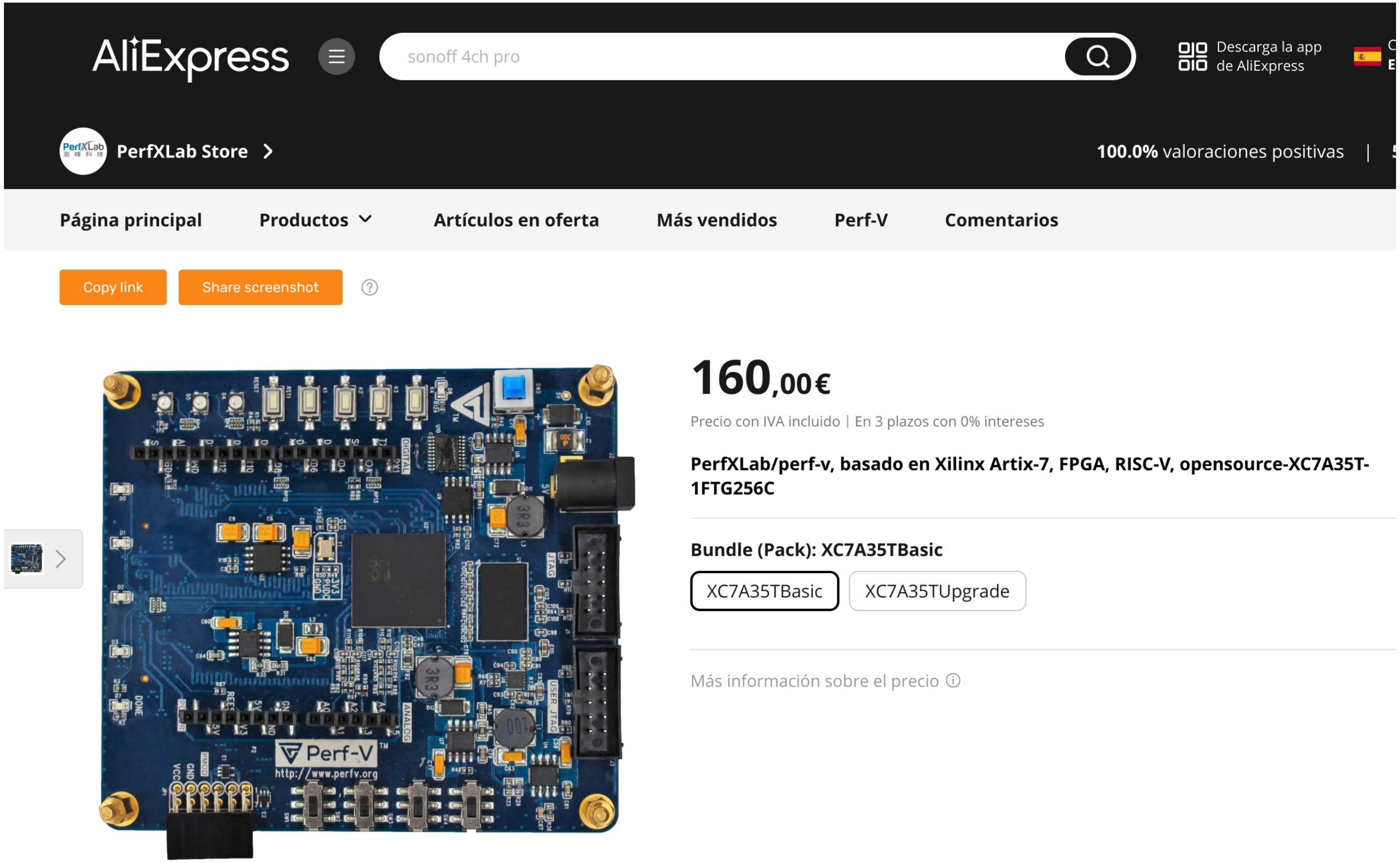
A starting point

- The IBEX project: https://github.com/lowRISC/ibex/tree/master
  -



A starting point

- PerfXLab boards



sonoff 4ch pro

PerfXLab Store

100.0% valoraciones positivas

160,00€

Precio con IVA incluido | En 3 plazos con 0% intereses

PerfXLab/perf-v, basado en Xilinx Artix-7, FPGA, RISC-V, opensource-XC7A35T-1FTG256C

Bundle (Pack): XC7A35TBasic

Más información sobre el precio

A starting point

- Beagle Board: https://www.beagleboard.org