# LOCOD

## AN OPEN-SOURCE HW/SW CO-DESIGN TOOL FOR SOC/FPGA

**Clément COGGIOLA**, Mickaël BRUNO, **Florent MANNI**
*CNES On-Board Data Handling Office*

**Hugo KIEFFER**, Julien ARMENGAUD, Sébastien TARRIS
*Viveris Technologies*

FPGA DEVELOPERS FORUM - CERN
**11-13 OF JUNE 2024**

# TABLE OF CONTENTS

01 INTRO

02 HARDWARE/SOFTWARE CO-DESIGN

03 WHAT IS LOCOD?

04 HOW LOCOD WORKS?

05 EXPERIMENTAL RESULTS

06 FUTURE WORK AND OPEN-SOURCE RELEASE

# 01

# INTRO

A QUICK OVERVIEW

# CNES

## CNES STRENGTH : 3 ROLES

### Space Agency

- Define and apply **France space policy**
- Represent France **at ESA and European Union**
- **Cooperate** and set partnerships
- French Space Operations Act (LOS)
- Education and outreach

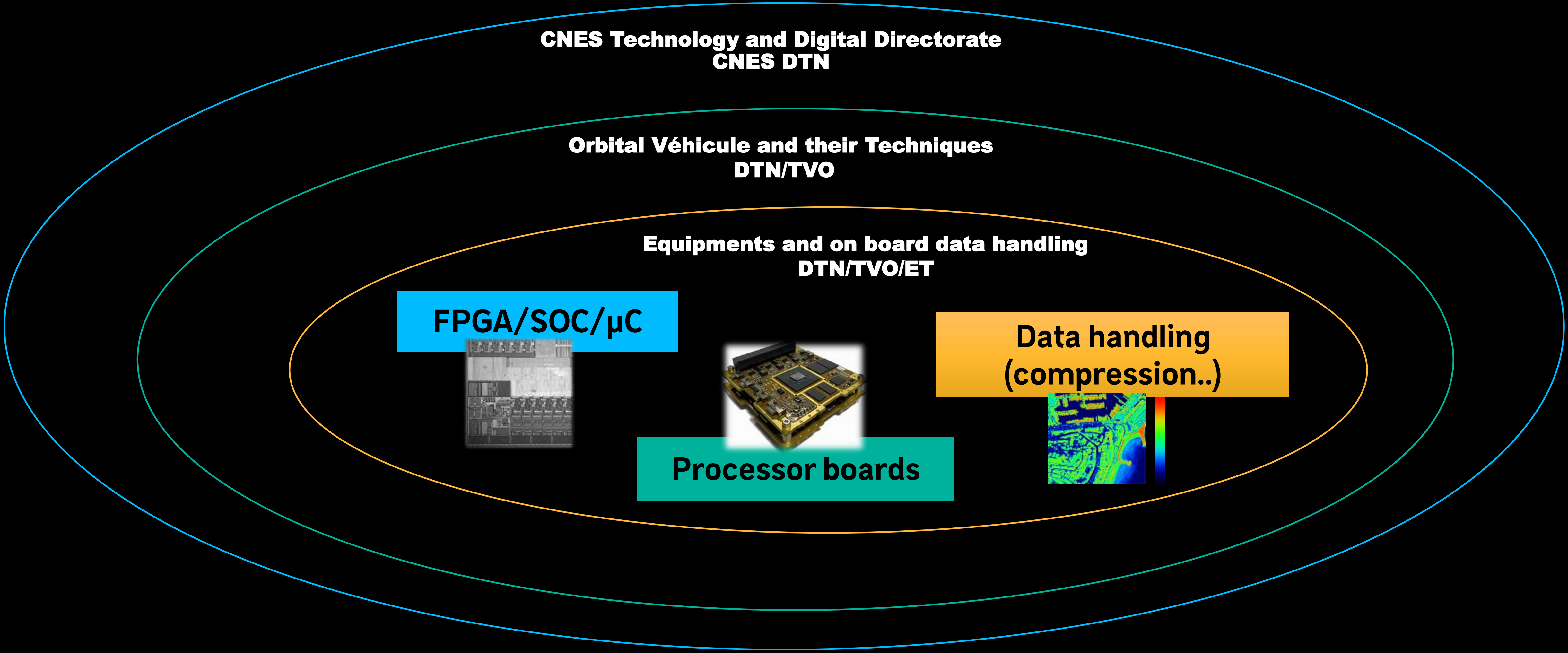
### Technical Centre

- Develop **technical skills and platforms**
- Propose, develop and exploit **space programs**
- Imagine and specify **future space systems**

### Operational Centre

- **Operational means** for **launches**, and in-orbit **operations**

# CNES



CNES Technology and Digital Directorate
CNES DTN

Orbital Véhicule and their Techniques
DTN/TVO

Equipments and on board data handling
DTN/TVO/ET

FPGA/SOC/µC

Processor boards

Data handling (compression..)

# THE VIVERIS GROUP

## Our values

### INNOVATE

We capitalise on high-level expertise to bring innovative approaches to projects.

### SIMPLIFY

The complexity of projects is constantly increasing: simplifying is a cultural issue for us.

### SHARE

We share our knowledge with our customers, our expertise with our employees and our passion for our professions with future graduates.

37

years

84,7

M€

4

Information Systems

Scientific Computing

Infrastructures

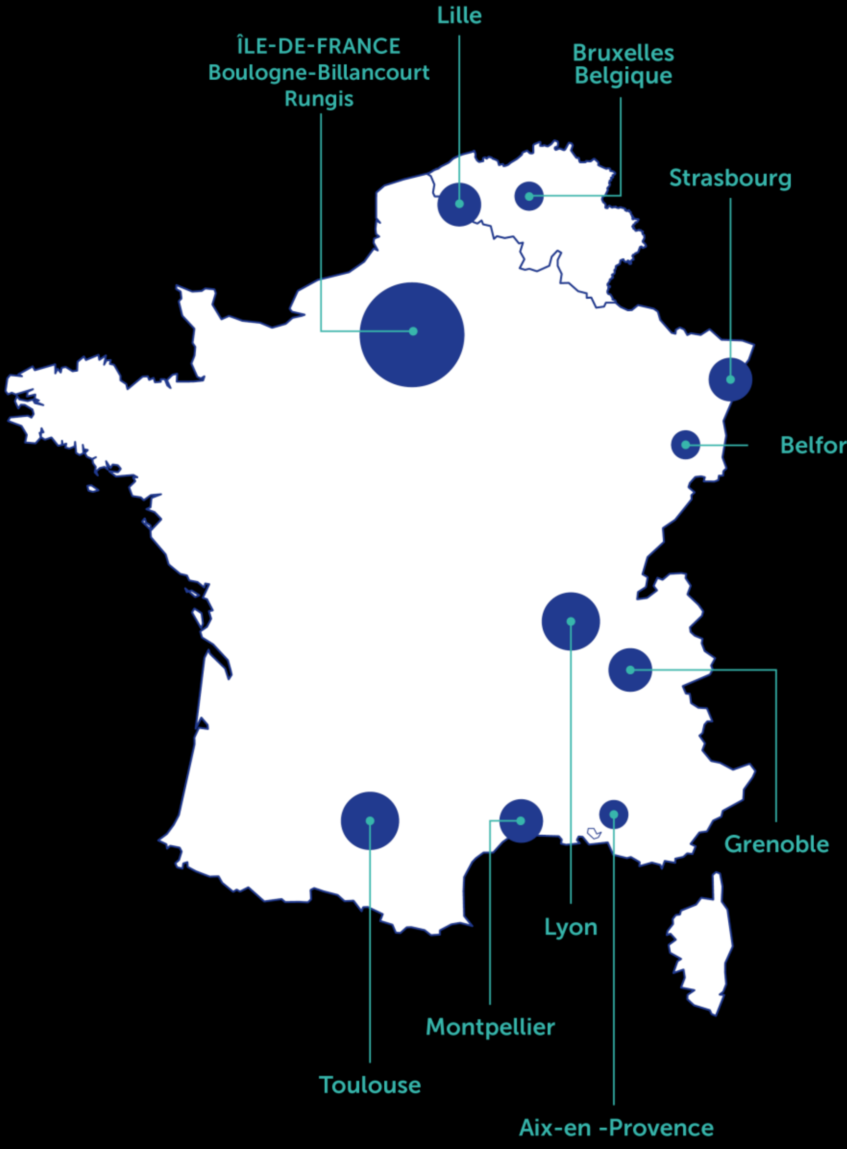
Embedded systems

940

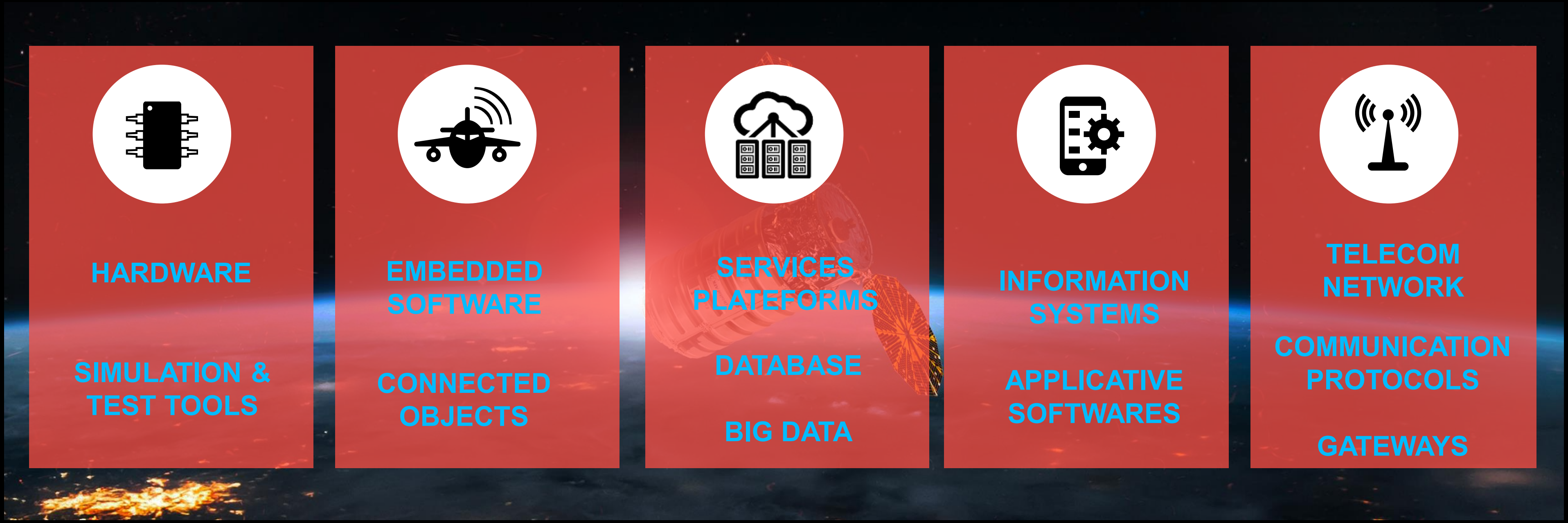
people

65%

Activities in Firm Fixed



# VIVERIS – TECHNICAL SKILLS (INDUSTRIAL)



HARDWARE

SIMULATION & TEST TOOLS

EMBEDDED SOFTWARE

CONNECTED OBJECTS

SERVICES PLATEFORMS

DATABASE

BIG DATA

INFORMATION SYSTEMS

APPLICATIVE SOFTWARES

TELECOM NETWORK

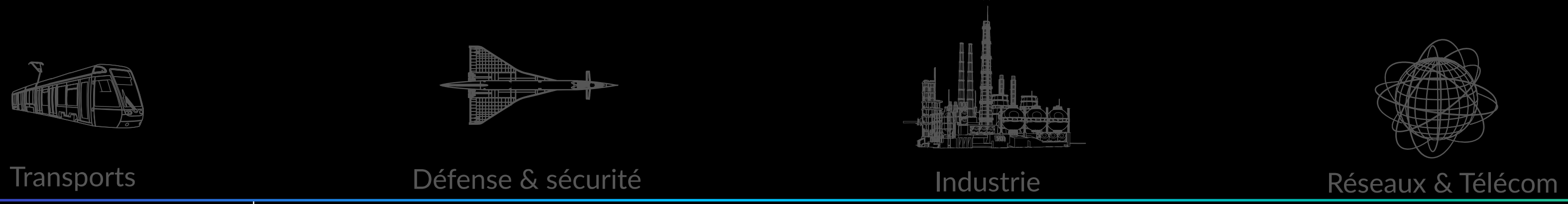
COMMUNICATION PROTOCOLS

GATEWAYS

(CYBER) SECURITY – INTEROPERABILITY - NORMS

Consulting, Projects, Audits, Training.

In the rail, automotive, energy and aerospace industries, the Embedded and Technical Computing Division provides its customers with expertise in the most challenging sectors:



Transports

Défense & sécurité

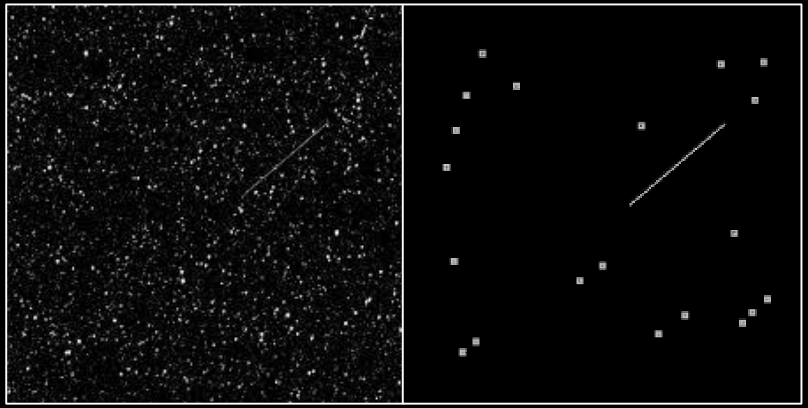
Industrie

Réseaux & Télécom

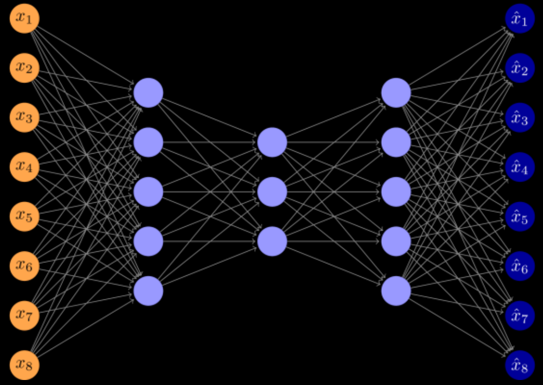
# TOWARDS NEW CHALLENGING ON-BOARD DATA PROCESSING

On-board computer capabilities

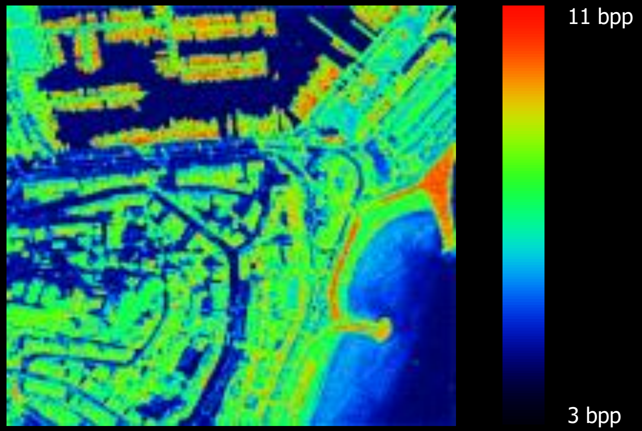
On-board algorithms complexity



Space Based Space Surveillance

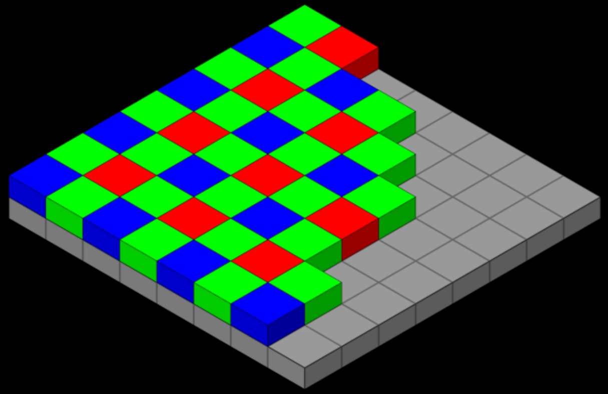


Edge AI



Quality-driven image compression

*Study of low complexity matrix sensors image compression for spacecrafts*



Color matrix sensors processing

Images sources:
SBSS: Sharma et al., Improving Space Surveillance with Space-Based Visible Sensor, MIT Lincoln Laboratory
Autoencoder https://tikz.net/autoencoder/
Bayer pattern: Colin M.L. Burnett

# HETEROGENEOUS COTS CHIPS

## FPGA

Xilinx KU060

NanoXplore NG-Medium 🇪🇺

MicroChip RTG4

...

## MANY-CORES CPU

Kalray MPPA Coolidge 🇪🇺

Ramon Space RC64

...

## SYSTEM-ON-CHIP CPU/FPGA

Xilinx Zynq 7000

NanoXplore NG-Ultra 🇪🇺

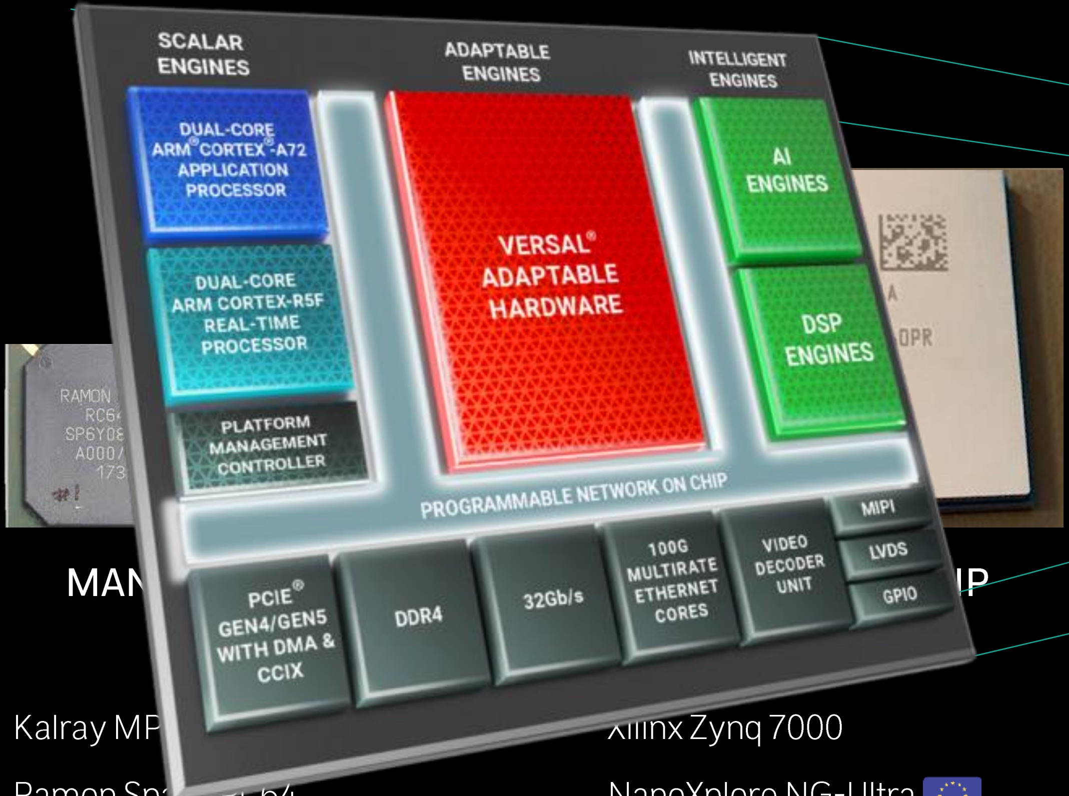
...

## NETWORK-ON-CHIP

Xilinx ACAP Versal

Images sources: NanoXplore, Kalray, Ramon Space, AMD Xilinx

# HETEROGENEOUS COTS CHIPS


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL® ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® GEN4/GEN5 WITH DMA & CCIX
DDR4
32Gb/s
100G MULTIRATE ETHERNET CORES
VIDEO DECODER UNIT
MIPI
LVDS
GPIO
RAMON
RC64

## FPGA

Xilinx KU060

NanoXplore NG-Medium

MicroChip RTG4

...

## MAN

Kalray MP

Ramon Space RC64

...

## IP

Xilinx Zynq 7000

NanoXplore NG-Ultra

...

## NETWORK-ON-CHIP

Xilinx ACAP Versal

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

cnes

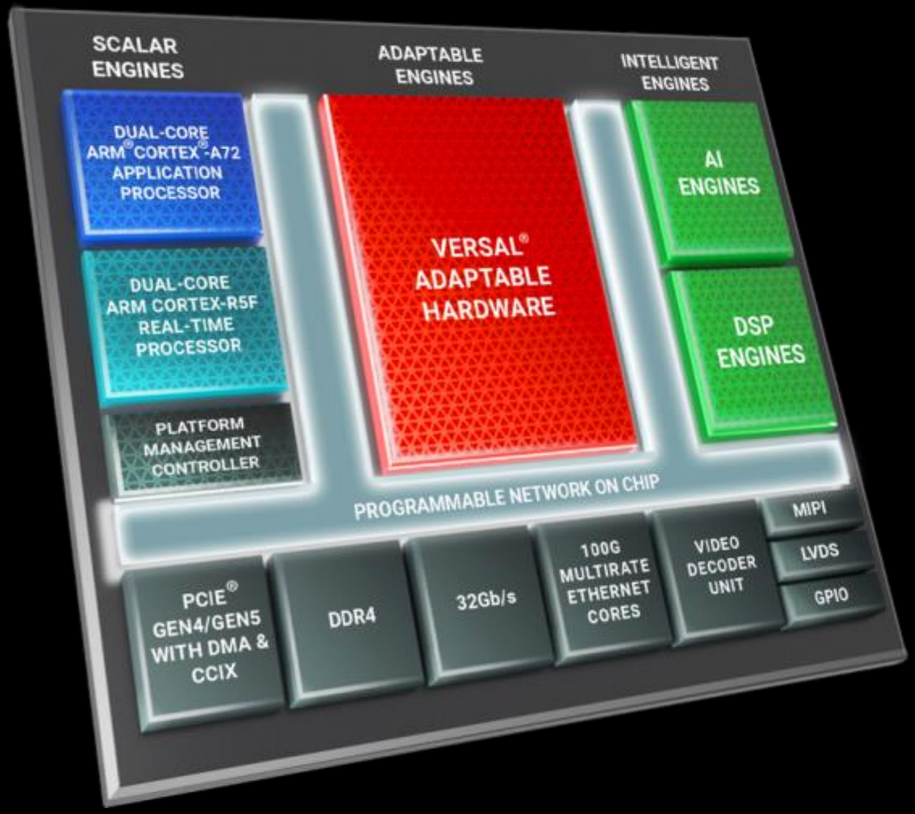
ViVERiS
Innover. Simplifier. Partager.

Images sources: NanoXplore, Kalray, Ramon Space, AMD Xilinx

# 02

# HW/SW CO-DESIGN

# WHY HW/SW CO-DESIGN?


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL® ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® GEN4/GEN5 WITH DMA & CCIX
DDR4
32Gb/s
100G MULTIRATE ETHERNET CORES
VIDEO DECODER UNIT
MIPI
LVDS
GPIO

**Processing optimization**

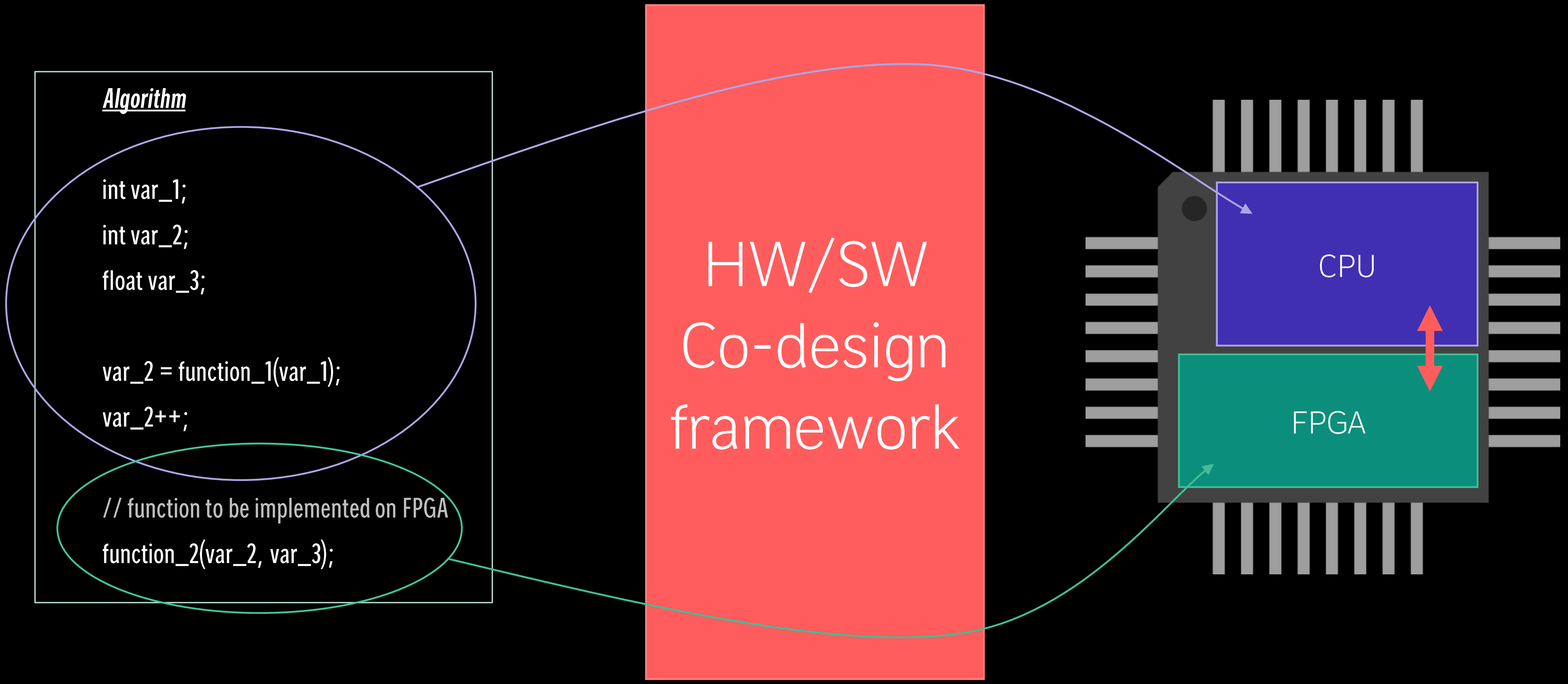
- Hardware-wise
- The earlier the better

**HW/SW co-design frameworks**

- Ease algorithms implementation
- Get the best from HW & SW
- Architecture exploration

Image source: AMD Xilinx

# HW/SW CO-DESIGN TOOLS



***Algorithm***

```
int var_1;
int var_2;
float var_3;

var_2 = function_1(var_1);
var_2++;

// function to be implemented on FPGA
function_2(var_2, var_3);
```

# 03

# WHAT IS LOCOD?

WHY LOCOD?

# LOCOD IS A HW/SW CO-DESIGN FRAMEWORK

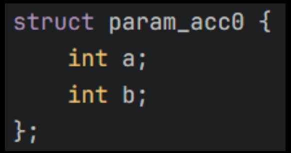
Fully open-source

Also targets NanoXplore NG-Ultra SoC

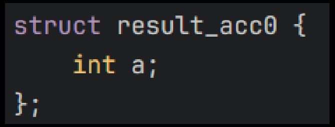
# WHAT DOES LOCOD PROVIDE?

## Standard C API

- Data structures to store parameters and results

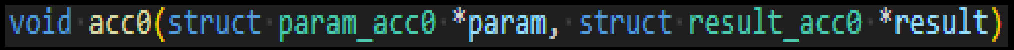


```
struct param_acc0 {
    int a;
    int b;
};
```



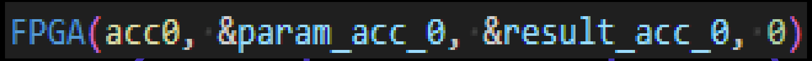
```
struct result_acc0 {
    int a;
};
```

- Generic function prototype



```
void acc0(struct param_acc0 *param, struct result_acc0 *result)
```

- Macro to launch functions in FPGA



```
FPGA(acc0, &param_acc_0, &result_acc_0, 0)
```

Function

Input data

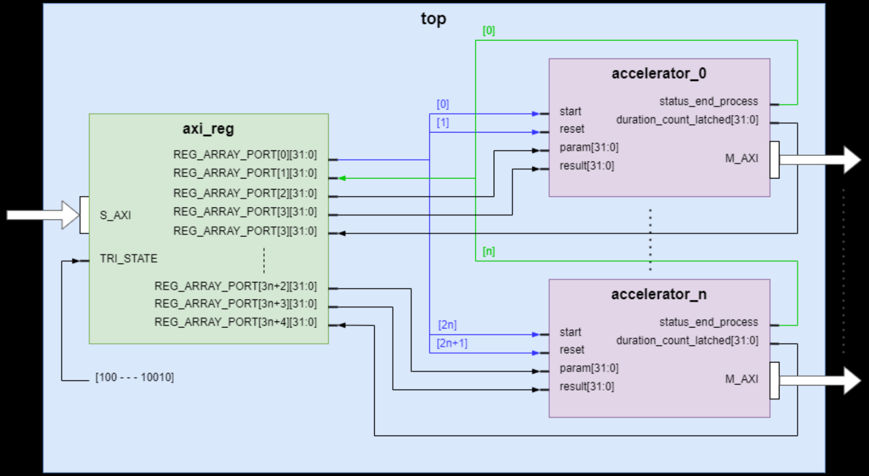
Output data

Accelerator index

- Methods to control accelerators execution

## Adaptable Hardware Architecture

- Top level FPGA component
  - Include accelerated function logic
  - Generic AXI interfaces

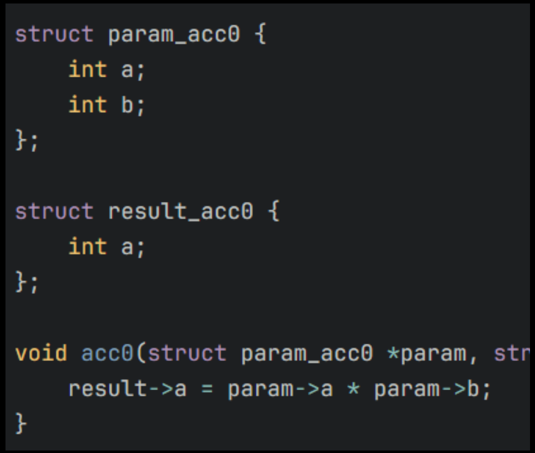


- Accelerators as many as necessary
  - Not limited by design
  - Only by the FPGA size

# HOW TO USE LOCOD?

1

Develop C code using the LoCod API



```
struct param_acc0 {
    int a;
    int b;
};

struct result_acc0 {
    int a;
};

void acc0(struct param_acc0 *param, str
    result->a = param->a * param->b;
}
```

2

Launch LoCod tool through CLI

```
./locod.sh -t <target> -f <source file>
```

3

Flash executable and bitstream on target board

# HOW TO USE LOCOD?

## Everything is developed in standard C

### Parameters shall be defined as structures.

```
Struct param_f1 {
    int a;
};
```

```
Struct result_f1 {
    int b;
};
```

F1
parameters

```
Struct param_f2 {
    int c;
    float d;
};
```

```
Struct result_f2 {
    float e;
};
```

F2
parameters

### Functions are standard C functions

```
void f1(struct param_f1 *param, struct result_f1 *result) {
    result->b = param->a + 10;
}

void f2(struct param_f2 *param, struct result_f2 *result) {
    result->e = param->c * param->d;
}
```

### Functions can be executed by CPU or FPGA

```
struct param_f1 p1 = { .a = 3 };
struct result_f1 r1;

struct param_f2 p2 = { .c = 0, .d = 3.14 };
struct result_f2 r2;

/* Call function f1 in CPU */
CPU(f1, p1, r1);

p1.a ++;
P2.c = p1.a;

/* Call function f2 in FPGA accelerator */
FPGA(f2, p2, r2, 0);
```

Accelerator
ID

### Launch LoCod

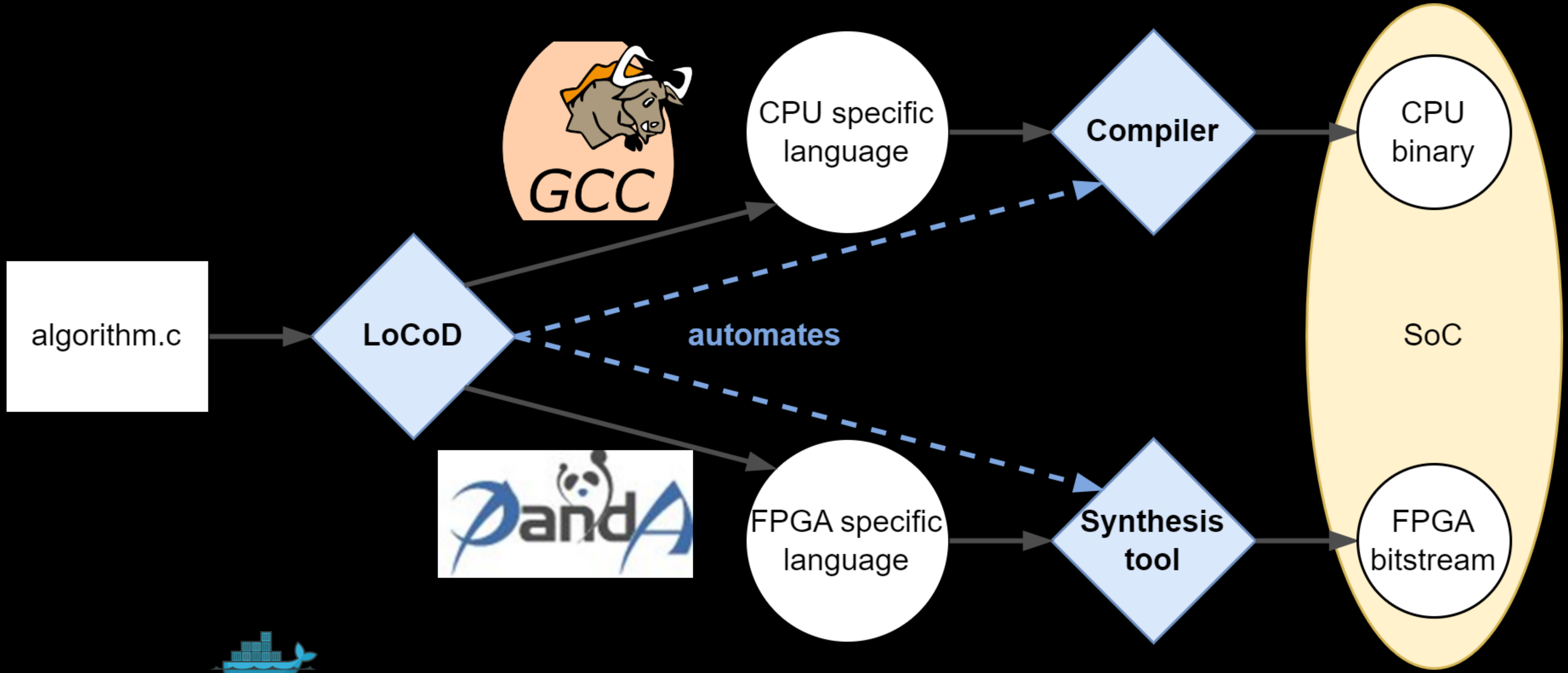
```
./locod.sh -t <target> -f <source file>
```

# 04

# HOW LOCOD WORKS?

CPU/FPGA INTERFACES AND HARDWARE ACCELERATORS

# TOP-LEVEL ARCHITECTURE



Each step is done in docker environments, allowing :

- **Modularity:** for future evolutions (new targets, others HLS…),
- **Testability:** all steps can be tested unitary.

Ferrandi F., et al. Invited: Bambu: An Open-Source Research Framework for the High-Level Synthesis of Complex Applications, 2021 58th ACM/IEEE Design Automation Conference (DAC), IEEE, dec. 2021. DOI: 10.1109/DAC18074.2021.9586110

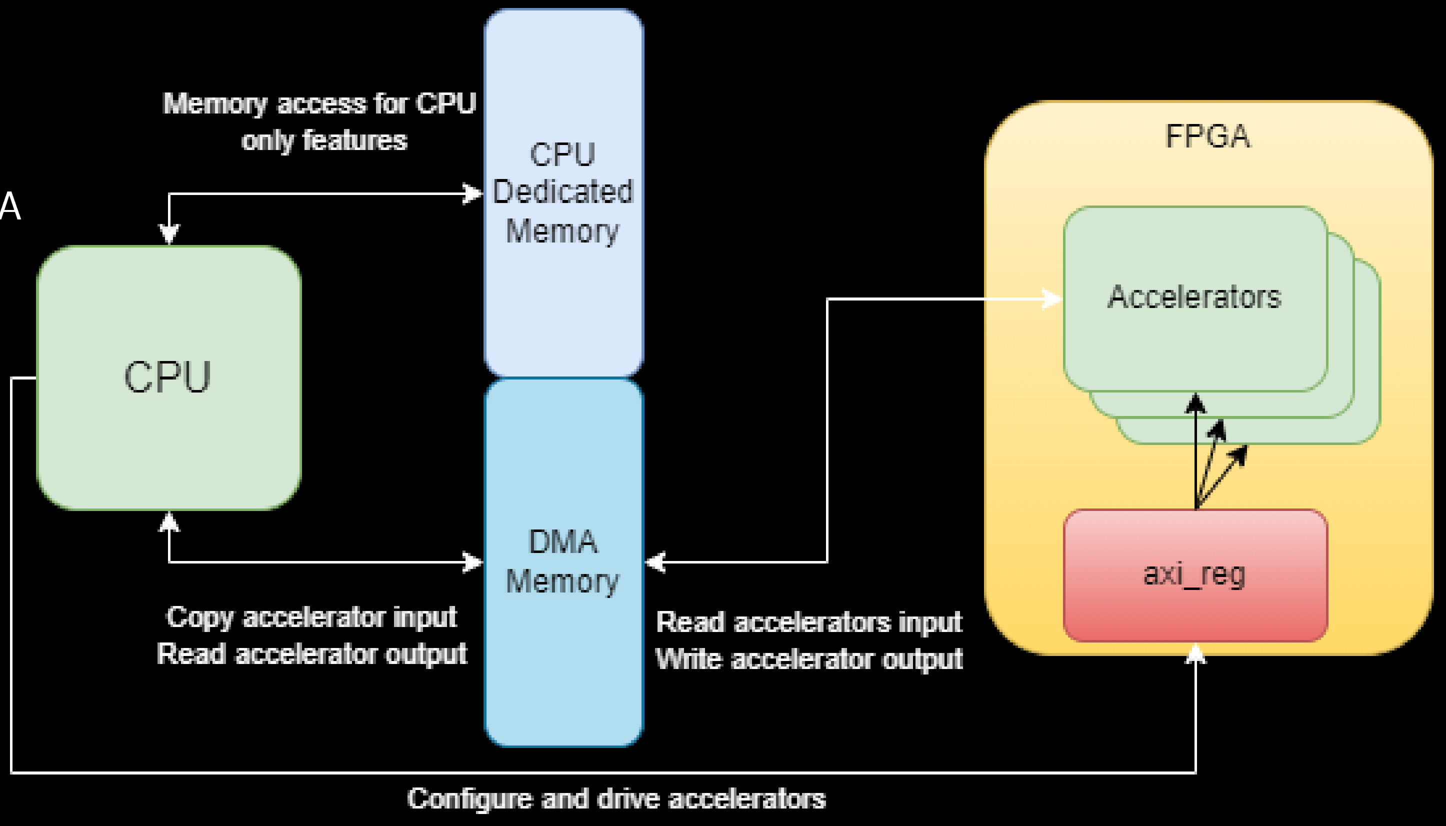
# CPU/FPGA INTERFACES

Memory:

- Splitted in CPU and DMA memory
  - CPU: for CPU usage.
  - DMA: for data exchange between CPU ⇔ FPGA

Axi register (axi_reg):

- Accelerators parameters
  - Input  data memory address
  - Output data memory address
- Drive Accelerator
  - Start / reset processing
  - Inform end of processing
- Measure processing time

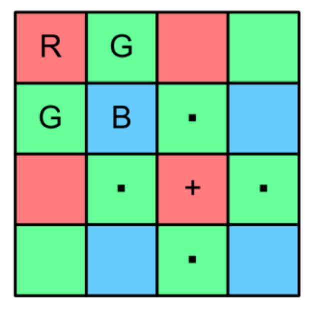


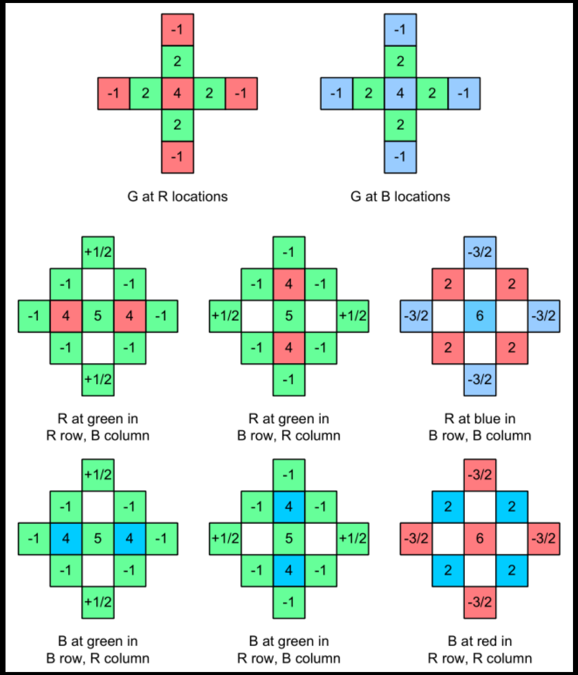
# 05

# EXPERIMENTAL RESULTS

# IMAGE PROCESSING USE CASE IMPLEMENTATION

- Algorithm for « Bayer pattern images demosaicing »
- Images from a matrix sensor
- Colour channels interpolation (Malvar)





G at R locations

G at B locations

R at green in R row, B column

R at green in B row, R column

R at blue in B row, B column

B at green in B row, R column

B at green in R row, B column

B at red in R row, R column

MATLAB (for reference)

Zynq Ultrascale+ (using LoCod)

Goals:

- Test LoCod with generic on-board image processing functions
- Identify limitations
- Identify evolutions

# 06

# FUTURE WORK

OPEN-SOURCE RELEASE AND IMPROVEMENTS

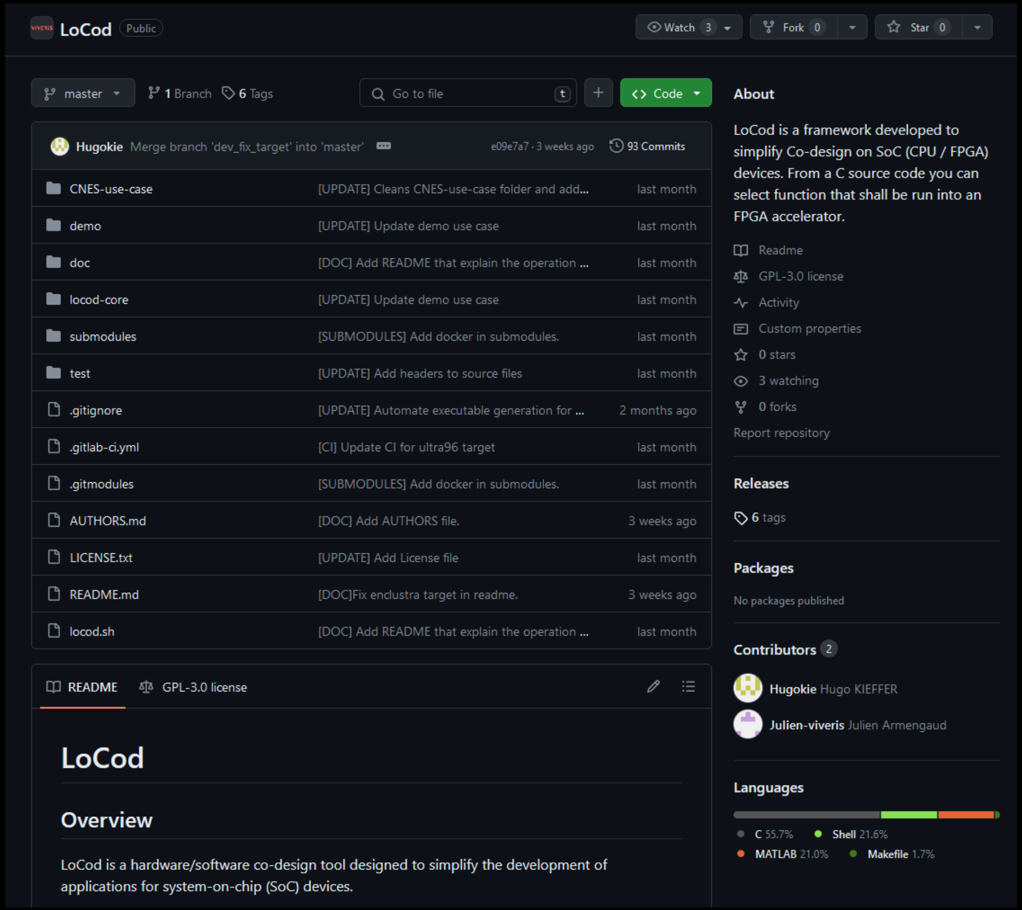
# STATUS

## Open-source release!

https://github.com/viveris/LoCod/

## Targets :

- Xilinx Zynq UltraScale+
- NanoXplore NG-Ultra
  - *Almost done*
  - *Debug in progress with NanoXplore*



Hugokie Merge branch 'dev_fix_target' into 'master' e09e7a7 · 3 weeks ago 93 Commits

| | | |
|---|---|---|
| CNES-use-case | [UPDATE] Cleans CNES-use-case folder and add... | last month |
| demo | [UPDATE] Update demo use case | last month |
| doc | [DOC] Add README that explain the operation ... | last month |
| locod-core | [UPDATE] Update demo use case | last month |
| submodules | [SUBMODULES] Add docker in submodules. | last month |
| test | [UPDATE] Add headers to source files | last month |
| .gitignore | [UPDATE] Automate executable generation for ... | 2 months ago |
| .gitlab-ci.yml | [CI] Update CI for ultra96 target | last month |
| .gitmodules | [SUBMODULES] Add docker in submodules. | last month |
| AUTHORS.md | [DOC] Add AUTHORS file. | 3 weeks ago |
| LICENSE.txt | [UPDATE] Add License file | last month |
| README.md | [DOC]Fix enclustra target in readme. | 3 weeks ago |
| locod.sh | [DOC] Add README that explain the operation ... | last month |

LoCod

Overview

LoCod is a hardware/software co-design tool designed to simplify the development of applications for system-on-chip (SoC) devices.

About

LoCod is a framework developed to simplify Co-design on SoC (CPU / FPGA) devices. From a C source code you can select function that shall be run into an FPGA accelerator.

Readme
GPL-3.0 license
Activity
Custom properties
0 stars
3 watching
0 forks
Report repository

Releases

6 tags

Packages

No packages published

Contributors 2

Hugokie Hugo KIEFFER
Julien-viveris Julien Armengaud

Languages

● C 55.7% ● Shell 21.6%
● MATLAB 21.0% ● Makefile 1.7%

# TO DO

- NanoXplore NG-Ultra support debug
- Benchmark multiple use cases
  - To highlight bottlenecks (further improvements)
  - To assess LoCod performances
- Wait for community feedback

https://github.com/viveris/LoCod/

# THANK YOU

ANY QUESTIONS?