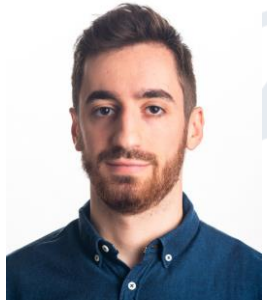
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

FPG-AI: a Technology Independent Framework for Edge AI Deployment Onboard Satellite, and its Characterisation on NanoXplore FPGAs

ESA OSIP Idea - Early Technology Development

ESA TEC-ED Final Presentation Days



Presenter: Tommaso Pacini

4 June 2024

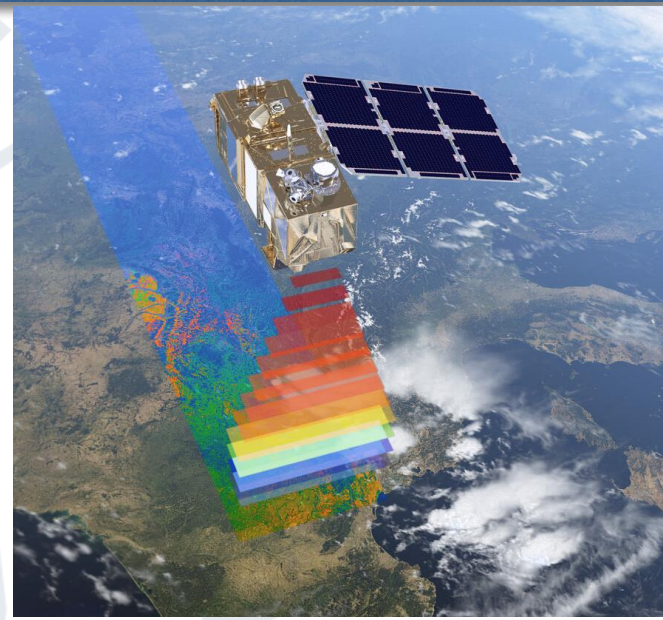
Presentation Outline

- Activity Context and Background
- Proposal Objectives
- FPG-AI Extension to RNNs
- Implementation on NX FPGAs
- Hardware Prototyping
- Conclusions

Presentation Outline

- Activity Context and Background
- Proposal Objectives
- FPG-AI Extension to RNNs
- Implementation on NX FPGAs
- Hardware Prototyping
- Conclusions

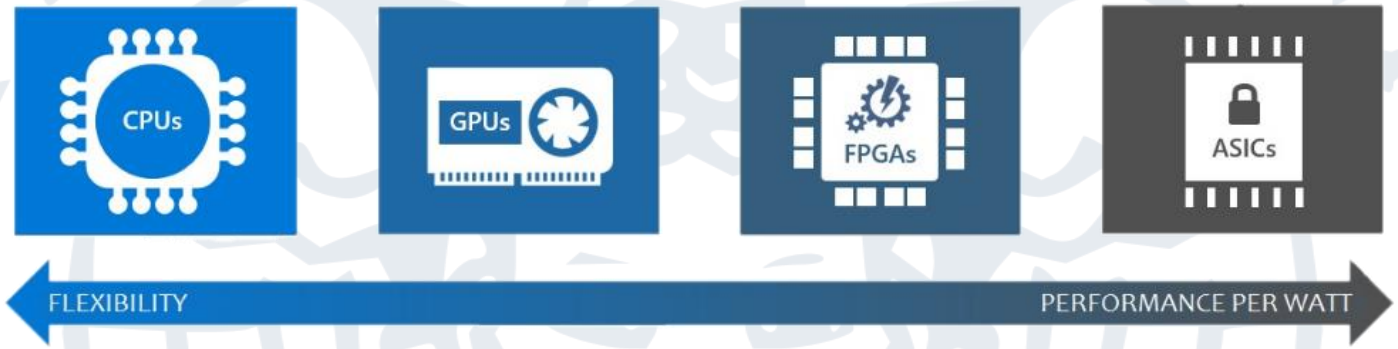
Activity Context



- Growing interest in **AI for space applications**:
  - Weather and Atmospheric Monitoring
  - Object Detection and Tracking
  - Ground Classification
  - Fault Detection, Isolation, and Recovery for Reliability
  - Autonomous Spacecraft Navigation
- AI deployment onboard the satellite constitutes an **open challenge**
- **Satellites** are **resource-** and **power- constrained** devices operating in a **harsh environment**
- The **complexity of AI models** collides with the limitations of satellite platforms

Activity Context

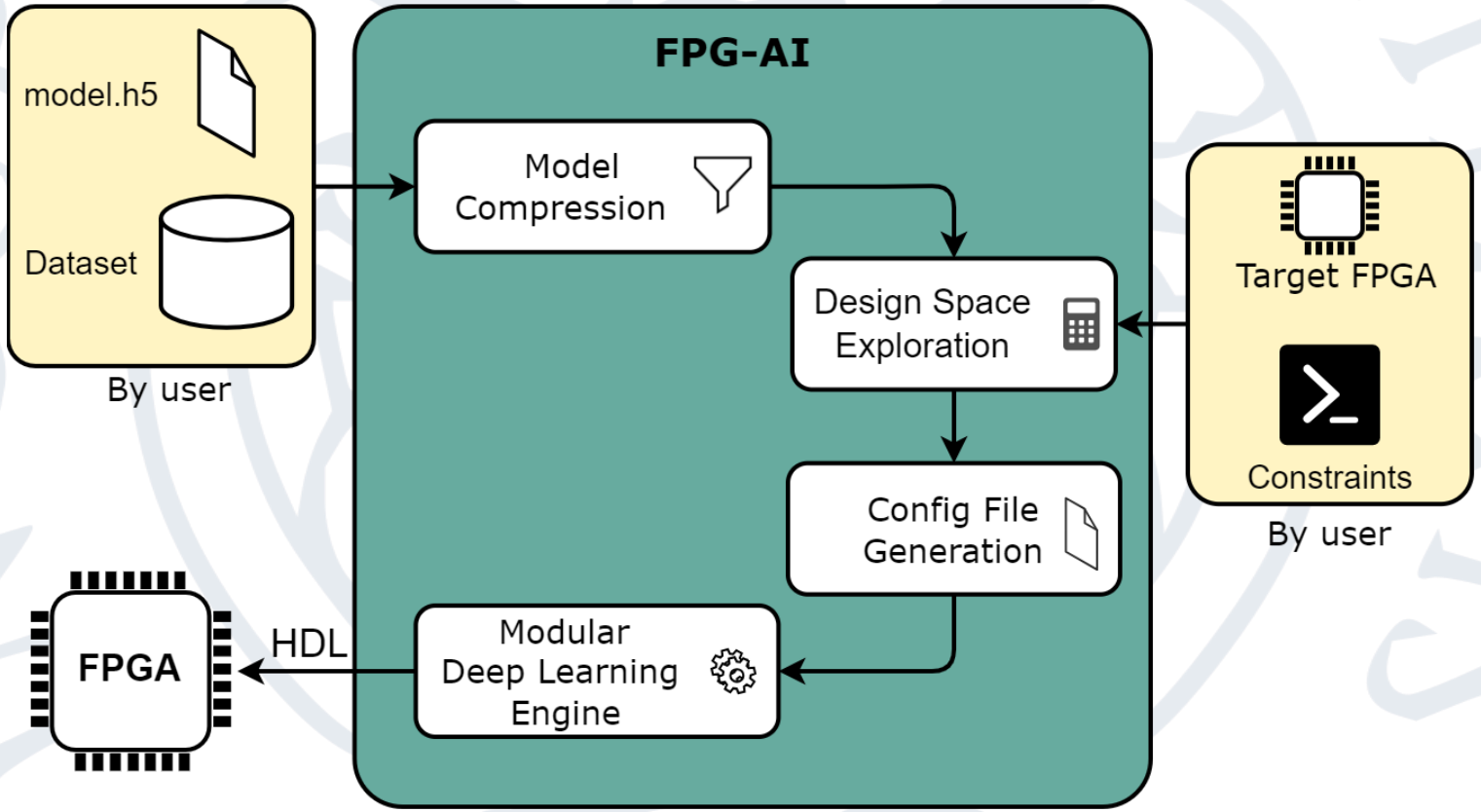
- Multiple hardware technologies are being investigated for AI acceleration onboard the satellite:



- FPGAs are a promising technology for AI acceleration for their energy efficiency and radiation tolerance
- The design of FPGA-based accelerators for AI typically requires high design expertise and long time-to-market
- Growing interest in **DNN-to-FPGA automation toolflows** for rapid AI deployment onboard the satellite

Background: FPG-AI Toolflow for CNNs

- Automation toolflow for efficient deployment of pre-trained CNN models on FPGA technology [1], [2]



FPG-AI Key Features

- **Easy integration in user-defined SoCs:**
  - Providing as output the accelerator HDL sources and not the final bitstream
  - Possibility to tune the resource consumption according to the requirements of other IPs
  - No workload sharing with the Host-CPU
- **Unmatched device portability of the Modular Deep Learning Engine (MDE) thanks to:**
  - Absence of third-party IPs
  - High scalability in terms of DSP/On-chip memory usage
  - Fine-grain configurable through a .vhd file

➡ Enabling the implementation on FPGAs from different vendors and heterogeneous resource budgets!

XILINX MICROCHIP intel

Presentation Outline

- Activity Context and Background
- Proposal Objectives
- FPG-AI Extension to RNNs
- Implementation on NX FPGAs
- Hardware Prototyping
- Conclusions

FPG-AI: Extension to RNNs and NanoXplore FPGAs

→ THE EUROPEAN SPACE AGENCY

ACTIVITY

FPG-AI: A TECHNOLOGY INDEPENDENT FRAMEWORK FOR EDGE AI DEPLOYMENT ONBOARD SATELLITE, AND ITS CHARACTERISATION ON NANOXPLORE FPGAS

Overview Events

RUNNING

Prime contractor
UNIVERSITA DI PISA

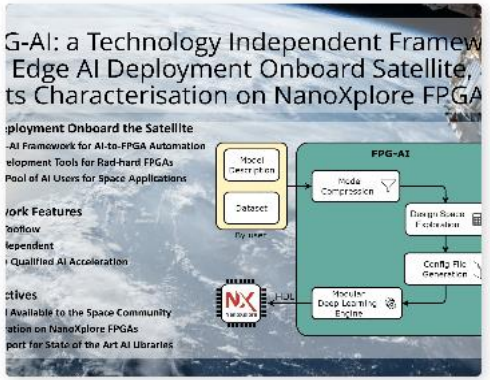
Organisational Unit
TEC-SF

START
31 March 2023

ESTIMATED END
01 October 2024

DURATION: 18 MONTHS

Activity Type
Early technology development



https://activities.esa.int/index.php/4000141108

Proposal Objectives

- “Extending and consolidating the framework to a wider set of supported AI algorithms, e.g. Recurrent Neural Networks (RNNs)”
- “Ensuring that all state-of-the-art devices are supported by the tool, especially focusing NanoXplore (NX) FPGAs, enabling the use of these devices for AI applications and pursuing European sovereignty”
- “Evaluating the tool capability with a prototype hardware demonstrator”

Team Composition

- **Full Prof. Luca Fanucci**
  - Responsible for Management tasks
  - Time allocated to the project: 2M

- **ESA Staff Silvia Moranti**
  - Technical Officer
  - ESA/ESTEC Microelectronics Section

- **Assistant Prof. Pietro Nannipieri**
  - Responsible for Hardware Prototyping and Dissemination Activity
  - Time allocated to the project: 4M

- **Dr. Tommaso Pacini**
  - Research Fellow
  - Responsible for Extension to RNNs, Extension to NX FPGAs, Benchmark Activity
  - Time allocated to the project: 12M

- **Eng. Matteo Dada**
  - Scholarship holder
  - Working on Extension to RNNs
  - Time allocated to the project: 5M

- **Eng. Tommaso Bocchi**
  - Working on Extension to NX FPGAs, Hardware Prototyping, Benchmark Activity
  - Time allocated to the project: 6M

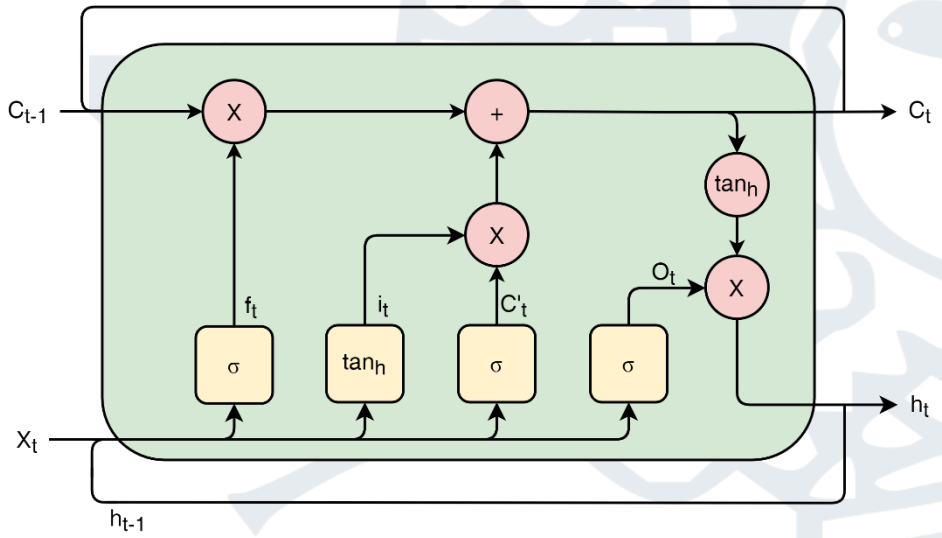
Presentation Outline

- Activity Context and Background
- Proposal Objectives
- FPG-AI Extension to RNNs
- Implementation on NX FPGAs
- Hardware Prototype
- Conclusions

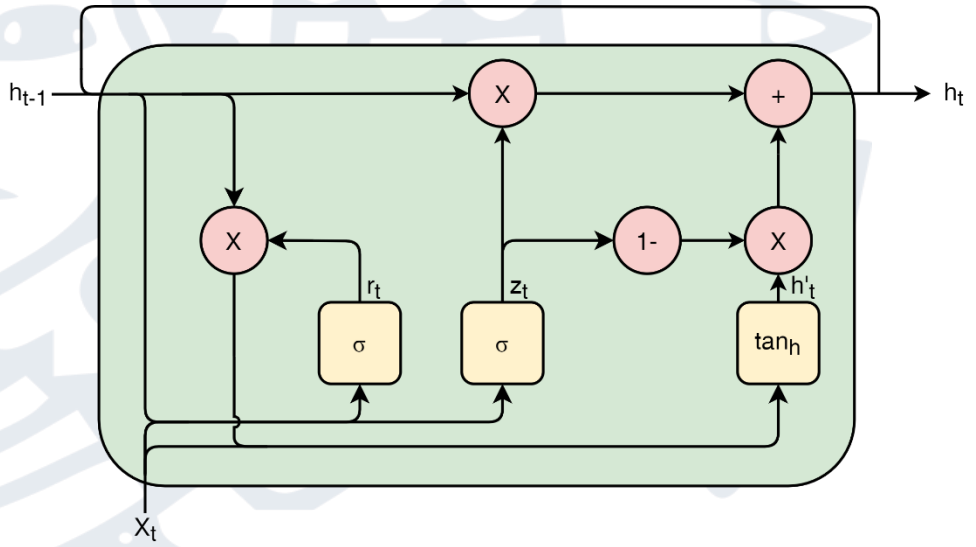
Recurrent Neural Networks (RNNs)

- Commonly used for sequence classification or time series forecasting tasks (e.g. FDIR onboard satellite)
- Exploiting feedback loops to deal with temporal sequences of data
- Most popular architectures:

Long Short-Term Memory (LSTM)

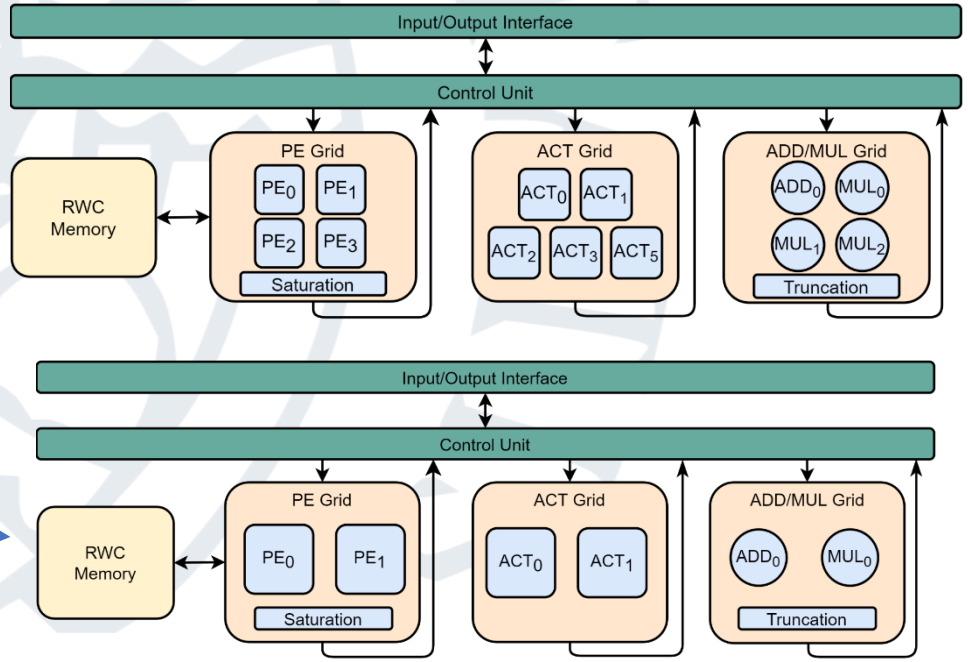


Gated Recurrent Unit (GRU)



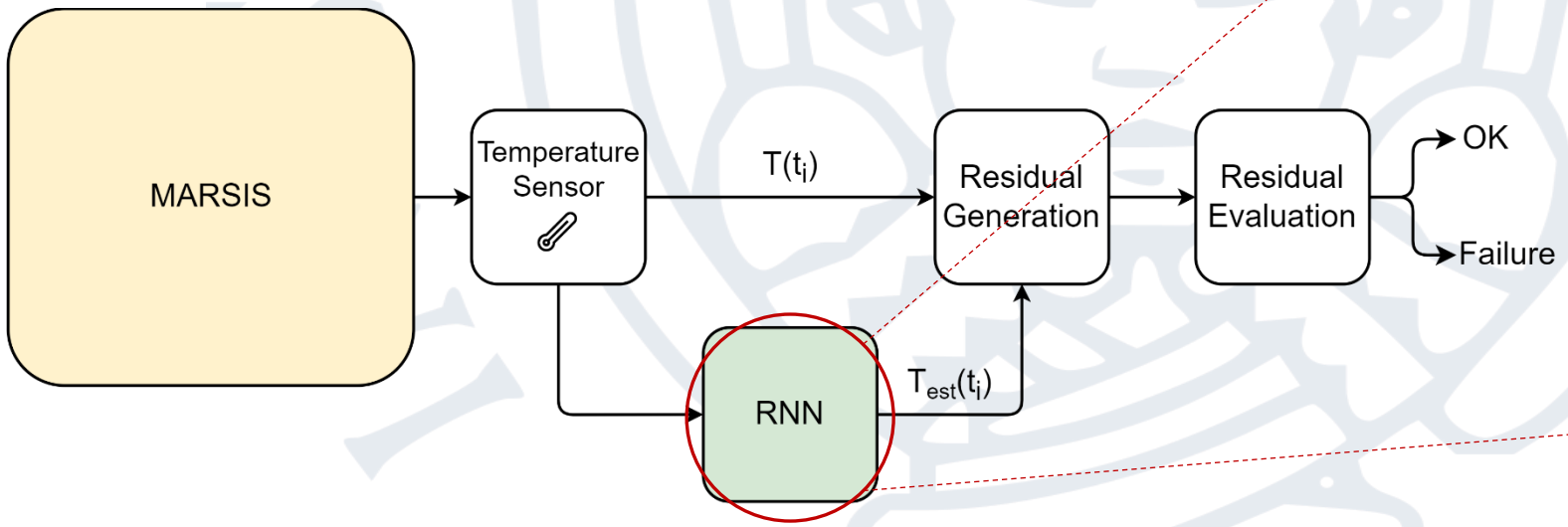
Extension to RNNs

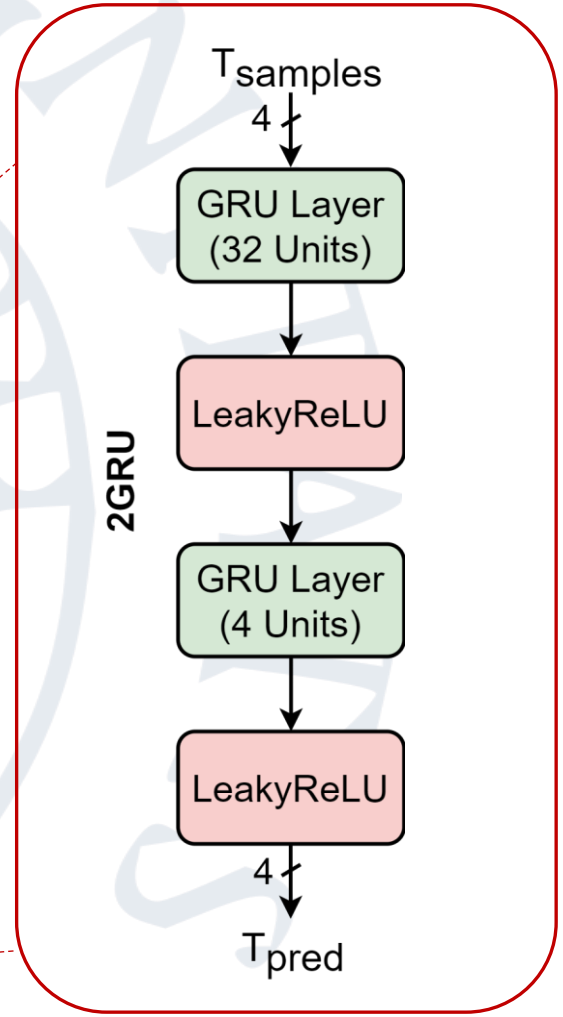
- **Model compression strategy**:
  - Global: Recurrent layers shares the same quantization configuration
  - Uniform: the quantization levels are equally spaced
  - Symmetric: each quantized distribution is centered on zero
  - Post-training: to be applied on pre-trained models
- **Design Space Exploration**:
  - Iterative algorithm which explores one configuration at a time
  - Exploiting a detailed analytical model of the hardware
  - Sensitive to user's constraints
- **Hardware architecture**:
  - Streamline architecture
  - HDL-based custom blocks for LSTM and GRU [3]
- **Test and validation**:
  - Bit-true comparison between quantized model and hardware outputs



Reported Case Study

- Fault Detection Isolation and Recovery (**FDIR**) system based on **MARSIS** (Mars Advanced Radar for Subsurface and Ionosphere Sounding) dataset [4]
- **RNN-based models** used to forecast temperature values collected by the sensors
- Maximum Absolute Error (MAE) used as quality metric





Implementation Results – 1 PE

- Implementation on AMD, Intel, and Microsemi FPGAs to prove the technology-independence:

	XCZU7EV	XC7Z045	XQRKU060	RTG4	10AX048
Vendor	AMD Xilinx			Microchip	Intel
LUT	23033 (10.0%)	23304 (10.7%)	23092 (7.0%)	37009 (24.4%)	12698 (6.9%)
FF	12874 (2.8%)	12846 (2.9%)	12867 (1.9%)	15379 (10.1%)	13523 (1.8%)
BRAM/LSRAM/M20K	7 (2.2%)	6 (1.1%)	7 (0.6%)	8 (3.8%)	3 (0.2%)
DSP	14 (0.8%)	14 (1.6%)	14 (0.5%)	63 (13.6%)	9 (0.7%)
MDE Frequency [MHz]	200.0	122.0	120.5	57.5	111.1
AXI Frequency [MHz]	333.3	212.8	222.2	95.2	212.8

Presentation Outline

- Activity Context and Background
- Proposal Objective and Organization
- FPG-AI Extension to RNNs
- Implementation on NX FPGAs
- Hardware Prototype
- Conclusions

Implementation on NX FPGAs 1/3

- **Study of NX Technology and Design Suite**:
  - Ramp up on NX Design Suite (Impulse 23.3.0.2)
  - Study of NanoXplore on-chip memory and DSP resources
  - **20/07/2023**: One-day visit to NanoXplore headquarters in Paris for early feedbacks on Impulse flow and for identifying the hardware platform

    ➔ **NG-Ultra FPGA** selected as the target device

- **Selected Case Studies**:
  - **LeNet-5**:
    - Digits recognition on MNIST dataset
    - Layers: 2x[Conv + AvgPool] + 3 Dense
    - 44K parameters (~1.36 Mbit)
  - **Network in Network (NiN)**:
    - Image classification on CIFAR10 dataset
    - Layers: 9 Conv + 1 GlobalAvgPool
    - 969K parameters (~29.68 Mbit)

Implementation on NX FPGAs 2/3

- **Upgrade of FPG-AI hardware architecture for NX technology**:
  - Main issue: low implementation frequency
  - Pin-point changes to the architectural stage for CNNs to reduce the critical path
  - Frequency increased from **28.6 MHz up to 43.0 MHz for LeNet-5, 15 MHz from up to 25.6 MHz for NiN** (for the MDE only)

- **Exploitation of FPG-AI and NX development tools to obtain implementation results:**
  - Summary of the collected results on **NG-Ultra**:

	LeNet-5	NiN	MobileNet	VGG16
1 PE	✓	✓	On-going (actively working with NX support team)	
16 PE	✓	✓		

Implementation on NX FPGAs 3/3

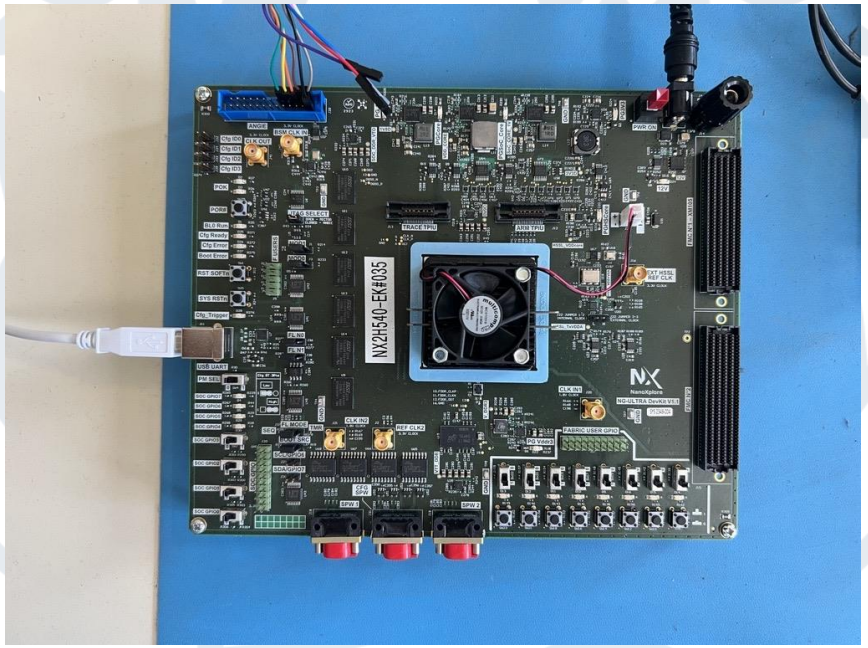
Model	#PE	LUT	FF	Register File Block	DPRAM	DSP	MDE Frequency [MHz]	AXI Frequency [MHz]
LeNet	1	9197 (2%)	4631 (1%)	89 (4%)	29 (5%)	51 (4%)	30.82	33.7
	16	13252 (3%)	5415 (2%)	38 (2%)	149 (23%)	426 (32%)	24.16	34.17
NiN	1	29503 (5.5%)	11433 (2.3%)	0 (0%)	340 (50.6%)	41 (3.1%)	20.8	24.4
	16	38247 (8%)	12450 (3%)	0 (0%)	297 (45%)	415 (31%)	19.5	33.8

Presentation Outline

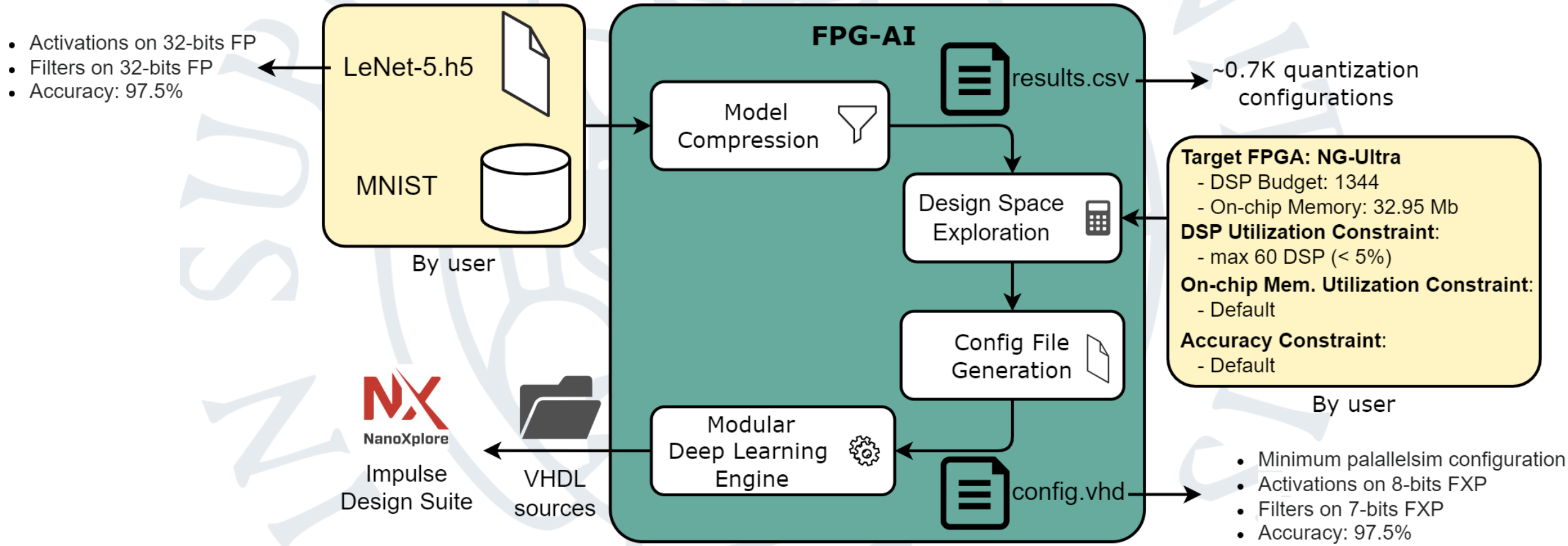
- Activity Context and Background
- Proposal Objectives
- FPG-AI Extension to RNNs
- Implementation on NX FPGAs
- Hardware Prototype
- Conclusions

Model and Platform Selection

- **Selection of a development platform hosting a NX FPGA**:
  - NG-Ultra Devkit Board v1.1, suggested by NanoXplore and kindly received on loan from ESA Microelectronic Section
- **Identification of the DNN model to be characterized in hardware**:
  - LeNet-5 selected as the target model (light and commonly used model)

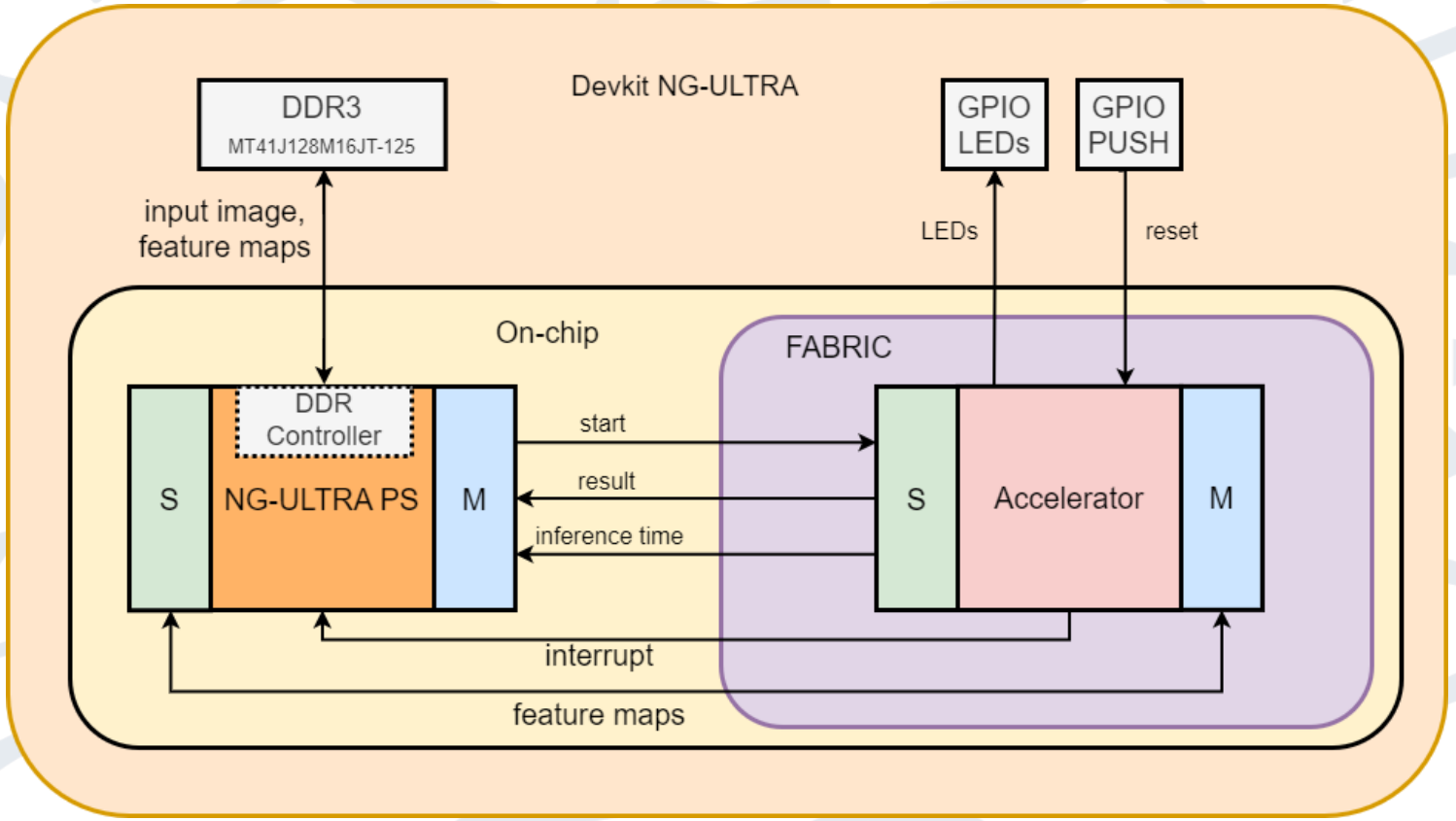


Accelerator Generation with FPG-AI



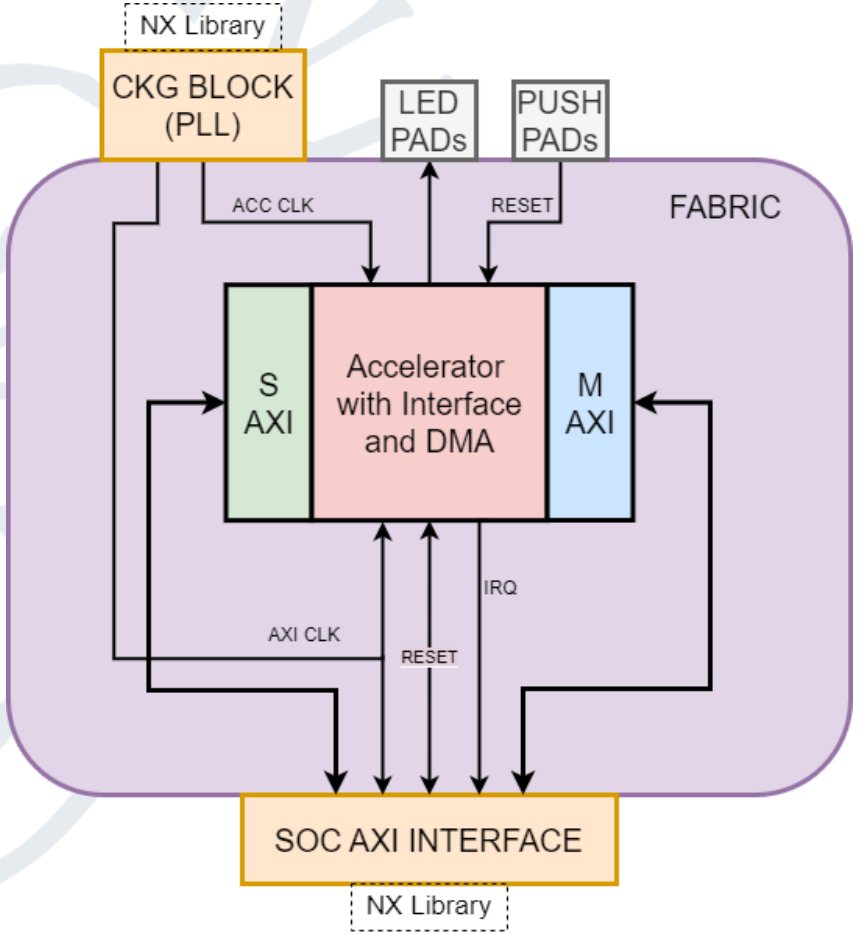
FPG-AI Integration on NG-Ultra Devkit SoC

- **Hardware prototype concept:**



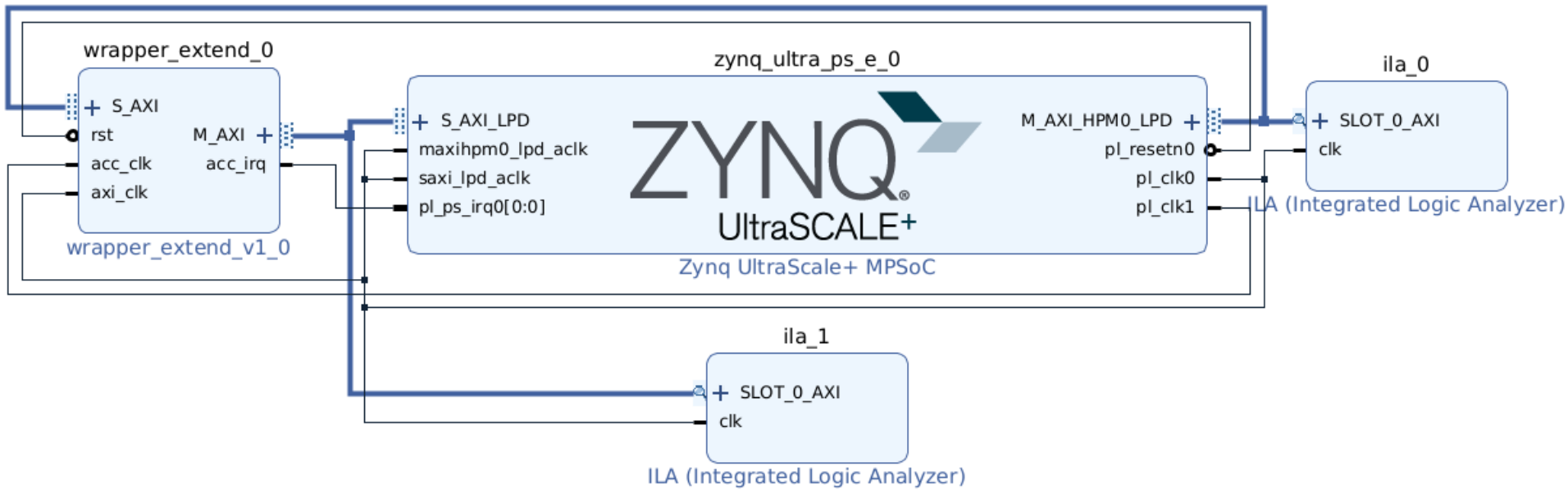
PL/PS Interface

- Extension of the accelerator AXI interface to 128-bit to ensure compliance with NG-Ultra SoC – Successful ✓
- Instantiation of the AXI SoC interface component (NX Library) in the VHDL top level to connect the PS and the PL – Successful ✓
- Development of C code to initialize the DDR memory and to control the accelerator – Successful ✓
- Clock generation with PLL – Successful ✓
- Performed test to validate communication between PS and accelerator's register file (AXI Slave interface) – Work in progress
  - Problem discussed with NX
  - **Outcome: NX recently released NX-scope logic analyzer support for NG-Ultra Dev-kit**
  - **Currently investigating the problem with the tool**



FPG-AI Integration on ZCU106

- Deployed LeNet-5 accelerator on an AMD Xilinx ZCU106 Board to validate FPG-AI interface and logic
- Featuring the ARM Cortex A53 on the PS side
- The accelerator IP is the same as that instantiated on the NG-ULTRA SoC



- **Test successful ✓**

Comparison with Bambu 1/2

- Bambu: open-source framework aimed at assisting the designer during the high-level synthesis of complex applications developed by Politecnico di Milano
- Target model:

Type	CNN
Dataset	MNIST
Input image size	28 x 28 x 1
# Convolutional layers	1
# Convolutional filters for each layer	12
Convolutional filters size	3 x 3
# Pooling layers	1
Pooling shape	2 x 2
Pooling type	Max Pooling
# FC layers	1
# Neurons for each FC layer	10

Comparison with Bambu 2/2

- Implementation results on NG-Ultra:

	Bambu [6]	FPG-AI – 1 PE	FPG-AI – 4 PE
LUT	4627 (0.9%)	4648 (0.9%)	5023 (0.9%)
FF	5714 (1.1%)	2428 (0.5%)	2533 (0.5%)
DPRAM	34 (2.2%)	21 (1.4%)	31 (2.0%)
DSP	54 (4.0%)	15 (1.1%)	42 (3.1%)
Frequency [MHz]	45.7	41.3	35.1
Cycles	169649	12300	5748
Inference time [ms]	3.71	0.30	0.16
Accuracy [%]	N/A	93.55	93.55

- The specificity of FPG-AI for AI workloads leads to higher efficiency metrics than general-purpose frameworks that exploit HLS code generation

Presentation Outline

- Activity Context and Background
- Proposal Objective and Organization
- FPG-AI Extension to RNNs
- Implementation on NX FPGAs
- Hardware Prototype
- Conclusions

Project Outcome

- **FPG-AI: end-to-end toolflow for the acceleration of DNNs on FPGAs**
  - Technology-independent flow: possibility to target FPGAs from Xilinx, Intel, Microsemi, and NanoXplore
  - Easy integration in user-defined SoCs and high degree of customization

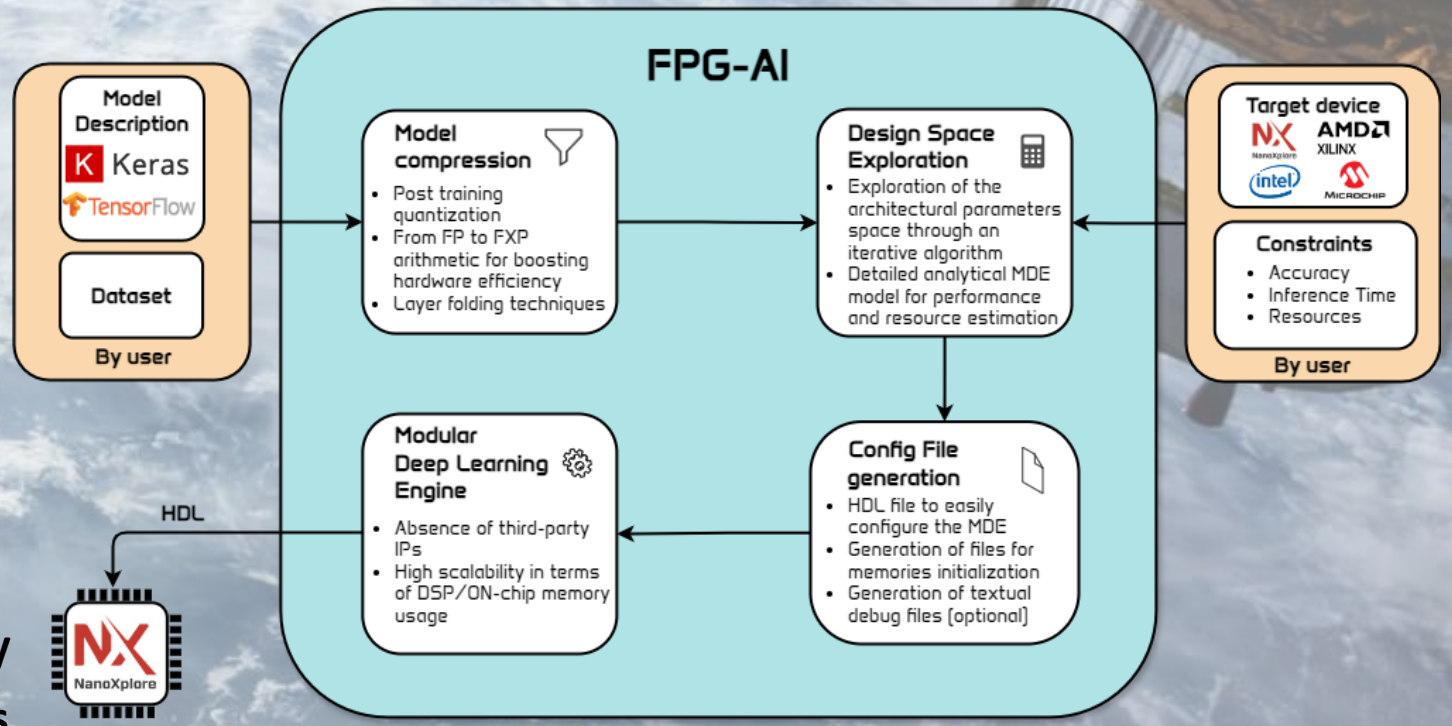
- **Extension to Recurrent Neural Networks (RNNs):**
  - Achieved implementation results on multiple RNN-FPGA pairs
  - Toolflow characterized for Fault Detection and Sequence Classification tasks

- **Extension to NanoXplore technology:**
  - Achieved implementation results for two CNN models targeting the NG-Ultra device
  - Deployed FPG-AI's accelerator on a Zynq ZCU106 Development Board to evaluate the flow
  - Built a solid expertise on NX flow that will be used to finalize the hardware prototype on NG-ULTRA

Thanks for the attention!

FPG-AI Framework Features

- Ready-to-use Tooflow
- Supporting for both CNN and RNN models
- Technology Independent HDL
- Extremely portable solution
- Enabling Space Qualified AI Acceleration



Project Technical Outcomes

- ✓ Made FPG-AI Available to the Space Community
- ✓ Designed support for LSTM and GRU RNN layers
- ✓ First AI Implementation on NanoXplore NG-ULTRA FPGA

Contacts:

- Prof. Luca Fanucci, luca.fanucci@unipi.it
- Assistant Prof. Pietro Nannipieri, pietro.nannipieri@unipi.it
- Research Fellow Tommaso Pacini, tommaso.pacini@phd.unipi.it