Inteligencia Artificial en Sistemas Embebidos

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Perú - Online - 2025 -

Inteligencia Artificial en Sistemas Embebidos

Objetivos del Curso:

- Comprender el rol de FPGAs y SoC en el despliegue eficiente de modelos de machine learning (ML).
- Aprender técnicas de compresión de modelos basados en ML (tales como pruning, quantization y knowledge distillation).
- Comprender el uso de hls4ml.
- Sintetizar modelos ML en hardware.
- Conocer flujos reales de inferencia embebida y sus aplicaciones en Edge AI.

Inteligencia Artificial en Sistemas Embebidos

Tres bloques principales:

- **Bloque 1:** Fundamentos y Compresión de Modelos.
- **Bloque 2:** Diseño Hardware y Síntesis de Modelos basados en ML.
- **Bloque 3:** Workflow Completo y Aplicaciones Reales.

The Abdus Salam International Centre for Theoretical Physics

Machine Learning and FPGA: Evolution, Current State of These Technologies, and Edge AI

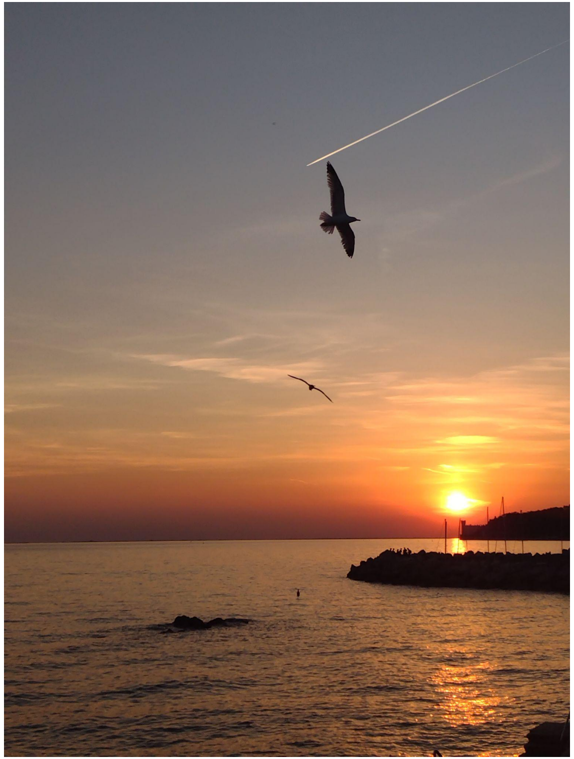
Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Perú - Online - 2025 -

Universidad Tecnológica del Perú

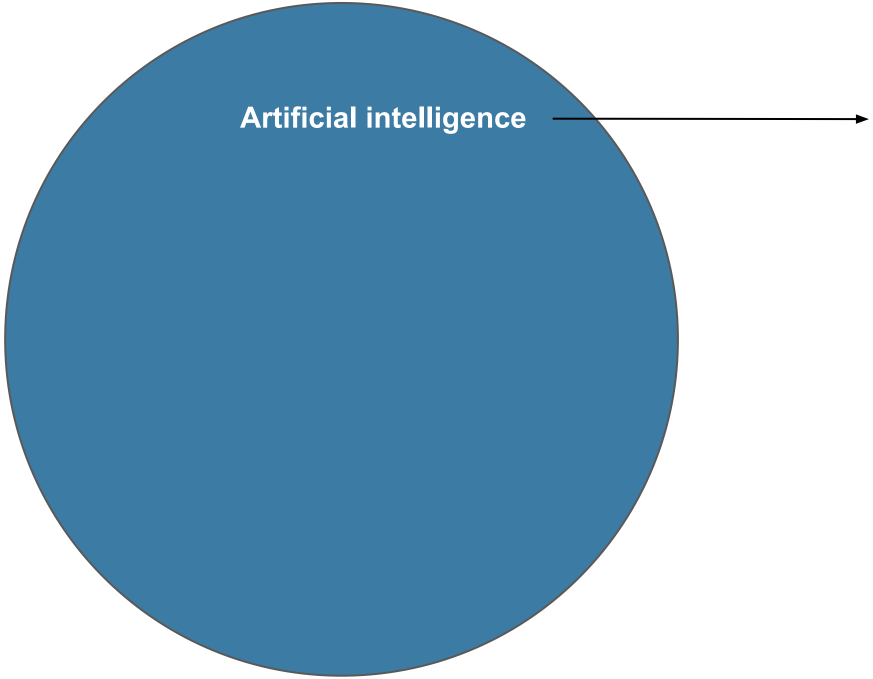
Outline

- Introduction.
- Edge AI.
- Remarks from the State-Of-The-Art.
- Optimizing every phase of the design and implementation process.
- MNIST-based binary classification.



Introduction

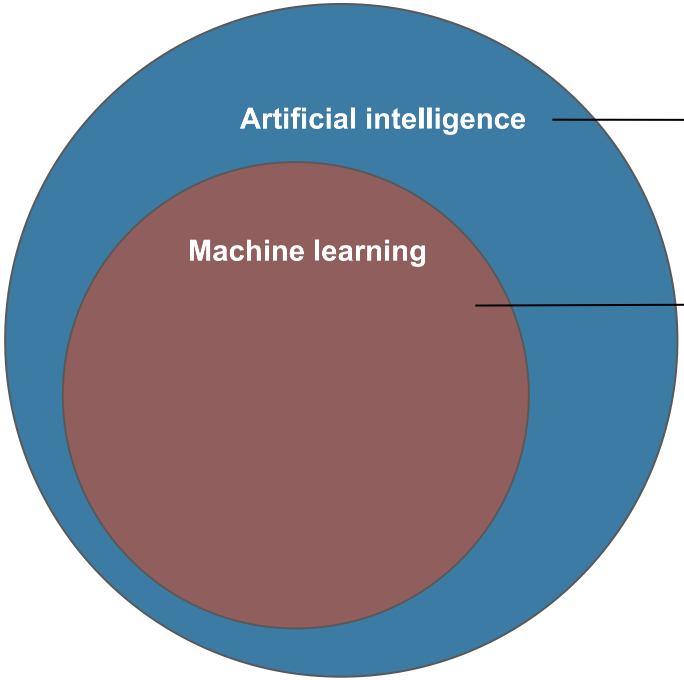
Introduction



A machine/system capable of imitating human behavior

The Abdus Salam International Centre for Theoretical Physics

Introduction

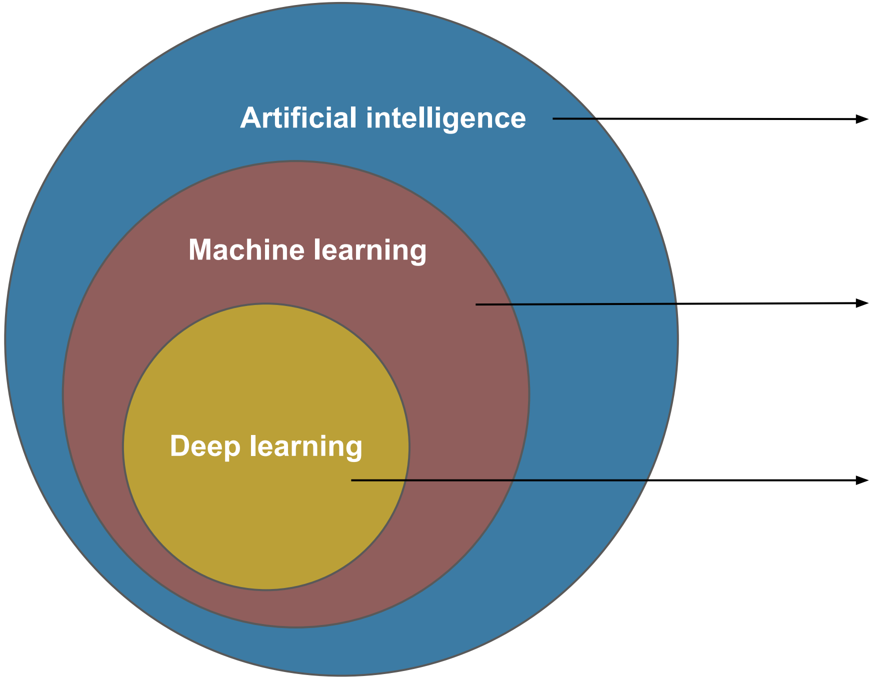


A machine/system capable of imitating human behavior

Mathematics and statistics to learn from data.

International Centre for Theoretical Physics

Introduction



A machine/system capable of imitating human behavior

Mathematics and statistics to learn from data.

Neural networks multi-layer

Introduction

- Why now? Three main components:

Introduction

- Why now? Three main components:

Big data

Introduction

- Why now? Three main components:

Big data

Software

Introduction

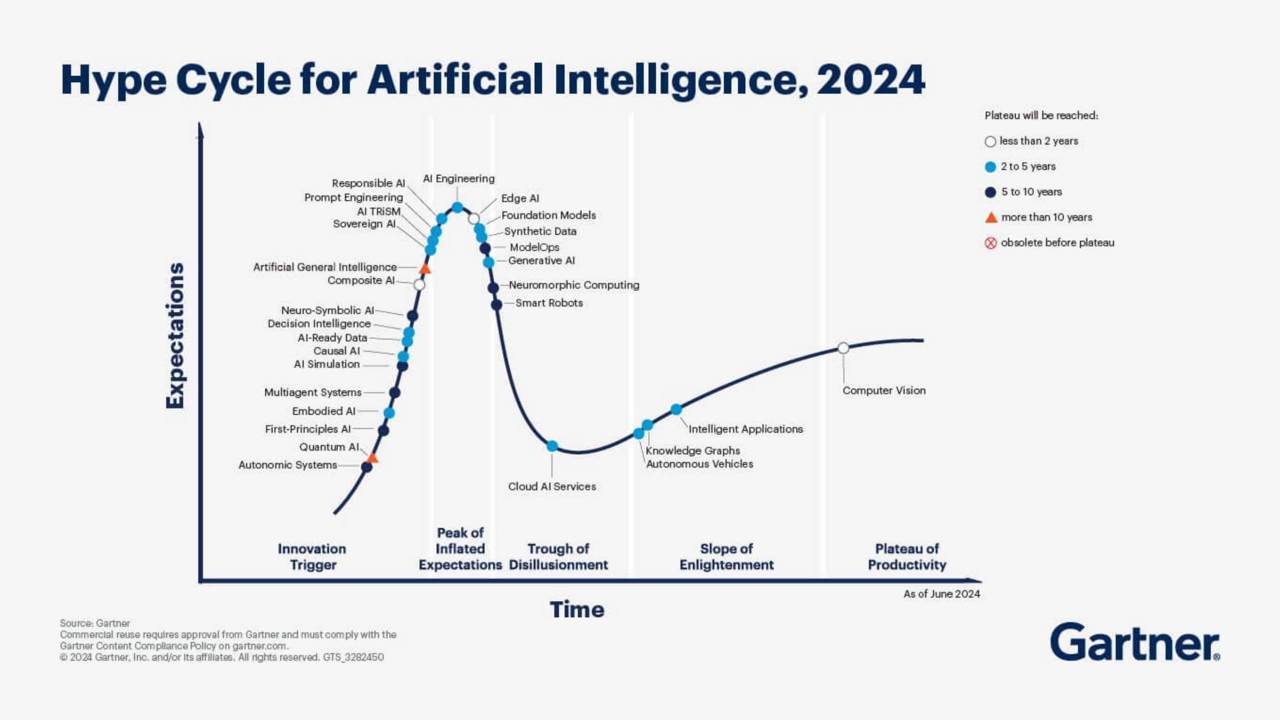
- Why now? Three main components:

Big data

Software

Hardware

Introduction



Graphically shows the maturity and adoption of technologies, helping to solve real problems and take advantage of opportunities.

Source: Gartner Hype Cycle on Artificial Intelligence (2023-2024) - https://www.gartner.com/en/articles/hype-cycle-for-artificial-intelligence

AI technologies

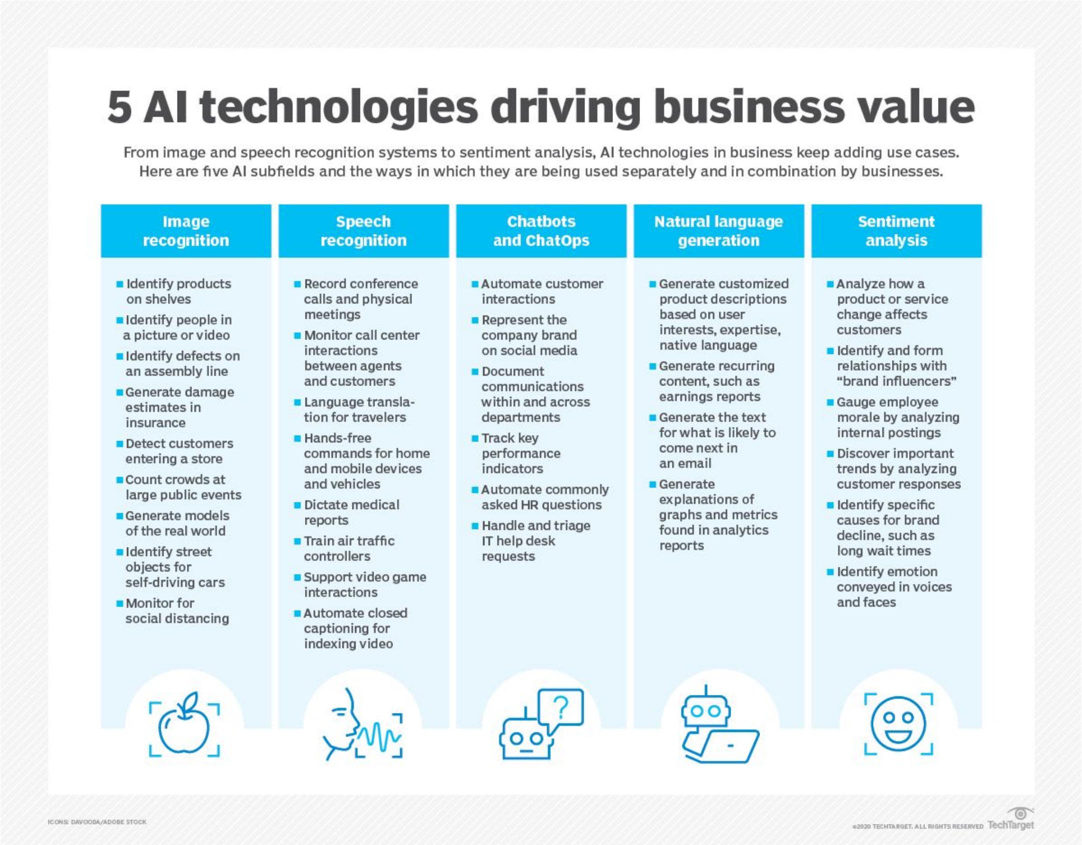


Image from
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- Advances in the following **technologies**:
  - Multimodal AI
  - AutoML
  - Embedded ML
  - MLOps
  - Low-code/No-code platforms
  - Reinforcement learning
  - Brain-computer interfaces
  - Neuromorphic processing
  - Digital Twins
  - Hardware platforms
  - Quantum computing
  - Among others

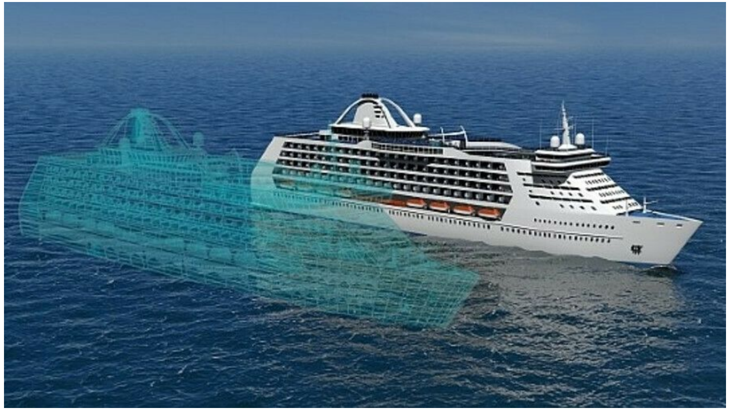


Image from https://www.rivieramm.com/news-content-hub/news-content-hub/digital-twin-developed-to-model-green-ship-technologynbsp-59419

From https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- Advances in the following **areas**:
  - Games
  - Autonomous driving
  - Cybersecurity
  - Intelligent drones
  - Precision agriculture
  - Education
  - Renewable Energy

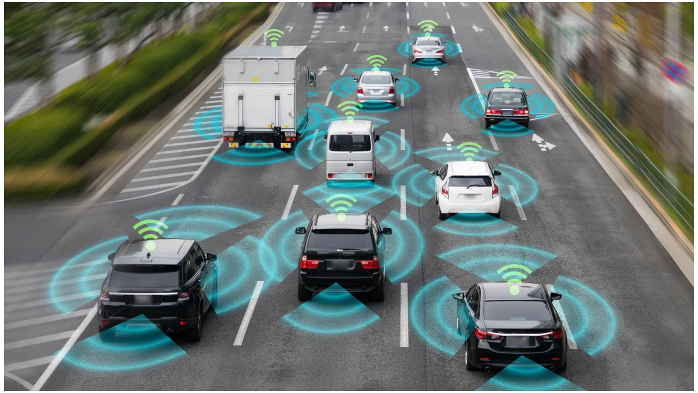


Image from
https://www.topgear.com/car%20news/what-are-sae-levels-autonomous-driving-uk

AI technologies beyond 2024

- **Market growth:**
  - The AI market is projected to grow at **35% annually**, surpassing **$1.3 trillion by 2030** (MarketsandMarkets).

From
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- **Market growth:**
  - The AI market is projected to grow at **35% annually**, surpassing **$1.3 trillion by 2030** (MarketsandMarkets).

- **Business applications:**
  - Gartner predicts a significant share will embed **conversational AI**.
  - Some new applications will be **automatically generated by AI**, without human intervention.

From
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- **Market growth:**
  - The AI market is projected to grow at **35% annually**, surpassing **$1.3 trillion by 2030** (MarketsandMarkets).

- **Business applications:**
  - Gartner predicts a significant share will embed **conversational AI**.
  - Some new applications will be **automatically generated by AI**, without human intervention.

- **Impact on businesses & jobs:**
  - Business models and job roles may undergo **rapid and unpredictable transformations**.

From
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline and https://www.marketsandmarkets.com/

AI technologies beyond 2024

- *Question: What do you think about the future of AI?*

AI-Specific Hardware Platforms

- **AI Chips & Processors**
  - **GPUs** (e.g., NVIDIA H100, AMD Instinct MI300) – Still the backbone of AI acceleration.
  - **TPUs** (Google Tensor Processing Units) – Optimized for deep learning workloads.
  - **NPUs** (Neural Processing Units) – Specialized for on-device AI in smartphones and edge devices.
  - **Analog & Optical AI Chips** – Emerging technology for ultra-low-power AI inference.

AI-Specific Hardware Platforms

- **AI Hardware Trends**
  - **Edge AI** – Custom low-power AI chips for real-time processing on IoT devices.
  - **RISC-V AI Processors** – Open-source architecture gaining traction for AI acceleration.
  - **Neuromorphic Computing** – Brain-inspired AI hardware mimicking human neural networks.

AI-Specific Hardware Platforms

- *Question: What do you think about FPGA and AI?*

AI-Specific Hardware Platforms

- **FPGA+AI Trends**
  - **AI-Specific FPGA Architectures** – New FPGA models optimized for deep learning (Xilinx Versal AI Core, Intel Stratix 10 NX, and Lattice Avant AI).
  - **Quantum + FPGA Hybrid Computing** – Research into integrating FPGA with quantum acceleration.
  - **FPGA as-a-Service (FaaS)** – Cloud-based FPGA solutions for scalable AI workloads.
  - **Embedded AI & Edge Computing** – FPGAs in IoT & industrial AI, enabling real-time decision-making with ultra-low power.

ML + FPGA: Shaping the Future

ML + FPGA: Shaping the Future

FPGA + ML in Robotics

- **Real-time vision**: FPGA accelerates neural network inference → robots with fast and accurate perception.
- **Low latency**: immediate decisions for navigation and motion control.
- **Energy efficiency**: essential for battery-powered mobile robots.
- **Example**: autonomous drones recognizing obstacles using CNNs on FPGA.

ML + FPGA: Shaping the Future

FPGA + ML in Bioengineering

- **Biomedical signal processing:** ECG, EEG, EMG analyzed in real time with ML on FPGA.
- **Wearable devices:** smart prosthetics or health monitors with low energy consumption.
- **Brain-computer interface (BCI):** fast classification of neural patterns to control robotic prosthetics.
- **Example:** hand prosthesis controlled by muscle signals processed on FPGA.

ML + FPGA: Shaping the Future

FPGA + ML in Automation and Control

- **Predictive maintenance:** ML detects failures → FPGA enables deployment in real-time industrial controllers.
- **Adaptive control:** embedded learning algorithms automatically tune parameters.
- **Industrial IoT:** FPGA as an intelligent node filtering data before sending to the cloud.
- **Example:** manufacturing plant with FPGA-based controllers optimizing processes.

ML + FPGA: Shaping the Future

FPGA + ML in Communications & Edge Computing

- **5G/6G networks:** FPGAs accelerate signal processing and enable flexible ML deployment in infrastructure.
- **Edge AI:** running ML models directly on nearby devices → ultra-low latency without relying on the cloud.
- **Cybersecurity:** real-time anomaly and intrusion detection with ML embedded on FPGA.
- **Example:** 5G base stations with FPGAs that process traffic and apply ML to optimize spectrum usage in real time.

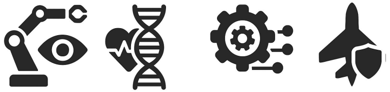
ML + FPGA: Shaping the Future

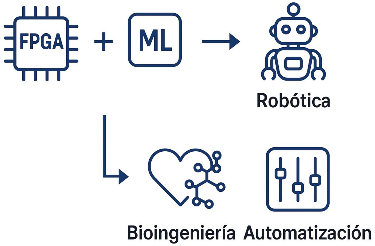
FPGA + ML in Aerospace & Defense

- **Autonomous navigation:** ML for path planning and obstacle detection, accelerated by FPGA for real-time decisions.
- **Radar and signal processing:** FPGAs handle massive parallel computations, while ML improves target recognition and tracking.
- **Mission-critical systems:** FPGA ensures reliability and low latency, ML adds adaptability and intelligence.
- **Example:** defense drones using FPGA-based ML to identify objects and react instantly in dynamic environments.

ML + FPGA: Shaping the Future

- **ML** provides intelligence (learning, prediction, classification).
- **FPGA** provides speed and efficiency (parallel processing, low latency, low power).
- **Together** → enable more autonomous robotics, more personalized bioengineering, and smarter control systems.





Edge AI

Edge AI

"**Edge artificial intelligence** (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure." [IBM]

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

"**Edge artificial intelligence** (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure." [IBM]

On the edge processing

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

"**Edge artificial intelligence** (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure." [IBM]

On the edge processing

Low latency

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

“**Edge artificial intelligence** (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.” [IBM]

On the edge processing

Privacy and security

Low latency

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

"**Edge artificial intelligence** (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure." [IBM]

- On the edge processing
- Privacy and security
- Low latency
- Reduced Bandwidth Usage

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

“**Edge artificial intelligence** (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.” [IBM]

On the edge processing

Privacy and security

Energy efficiency

Low latency

Reduced Bandwidth Usage

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

- By incorporating CPUs, GPUs, and FPGAs into a single system, **heterogeneous computing** becomes possible.

Edge AI

- By incorporating CPUs, GPUs, and FPGAs into a single system, **heterogeneous computing** becomes possible.

- This approach maximizes the strengths of each component, efficiently distributing edge workloads to improve both performance and energy consumption.

Edge AI

- **CPU:** handles general-purpose tasks like system management, control logic, and lightweight AI inference.

Edge AI

- **CPU:** handles general-purpose tasks like system management, control logic, and lightweight AI inference.

- **GPU:** accelerates parallelizable tasks such as deep learning inference, image processing, and high-performance computing.

Edge AI

- **CPU:** handles general-purpose tasks like system management, control logic, and lightweight AI inference.
- **GPU:** accelerates parallelizable tasks such as deep learning inference, image processing, and high-performance computing.
- **FPGA:** provides real-time, ultra-low-latency processing for tasks like sensor fusion, encryption, and custom AI accelerators.

Edge AI based on FPGA

FPGA / SoC-based on FPGA

Edge AI based on FPGA

Low latency

FPGA / SoC-based on FPGA

Edge AI based on FPGA

Low latency

Energy Efficiency

FPGA / SoC-based on FPGA

Edge AI based on FPGA

- Low latency
- Energy Efficiency
- High parallelism

FPGA / SoC-based on FPGA

Edge AI based on FPGA

- Low latency
- Energy Efficiency
- High parallelism
- Scalability

FPGA / SoC-based on FPGA

Edge AI based on FPGA

- Low latency
- Energy Efficiency
- High parallelism
- Scalability
- Customizable AI Acceleration

FPGA / SoC-based on FPGA

Edge AI based on FPGA

- Low latency
- Energy Efficiency
- High parallelism
- Scalability
- Customizable AI Acceleration

FPGA / SoC-based on FPGA

Resource-constrained devices

Edge AI based on FPGA

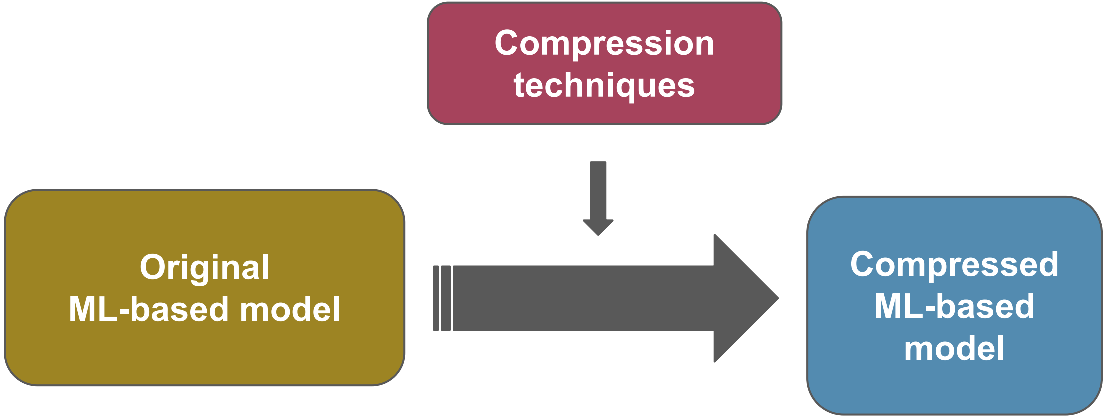
Original
ML-based model

Edge AI based on FPGA

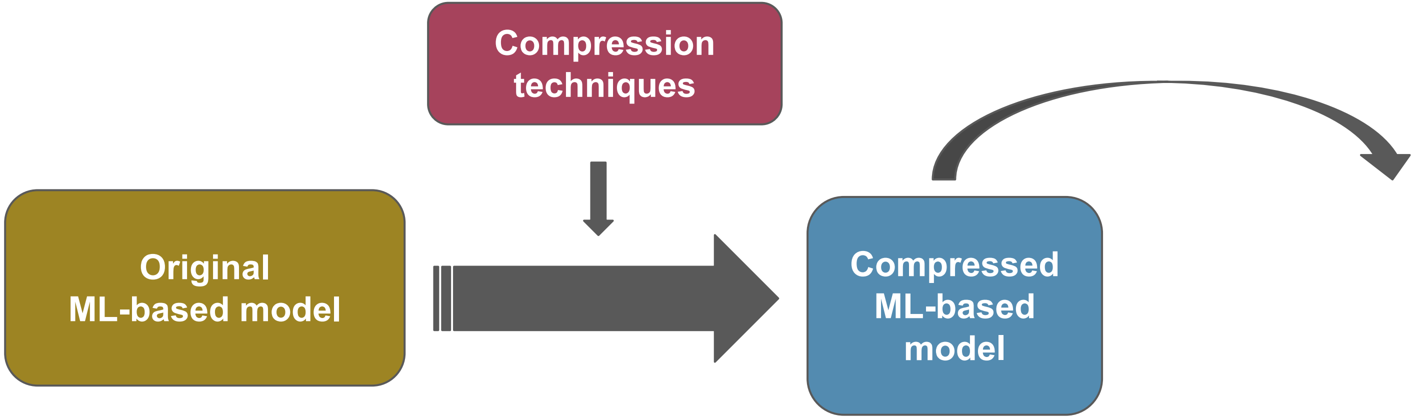
Original
ML-based model

Edge AI based on FPGA

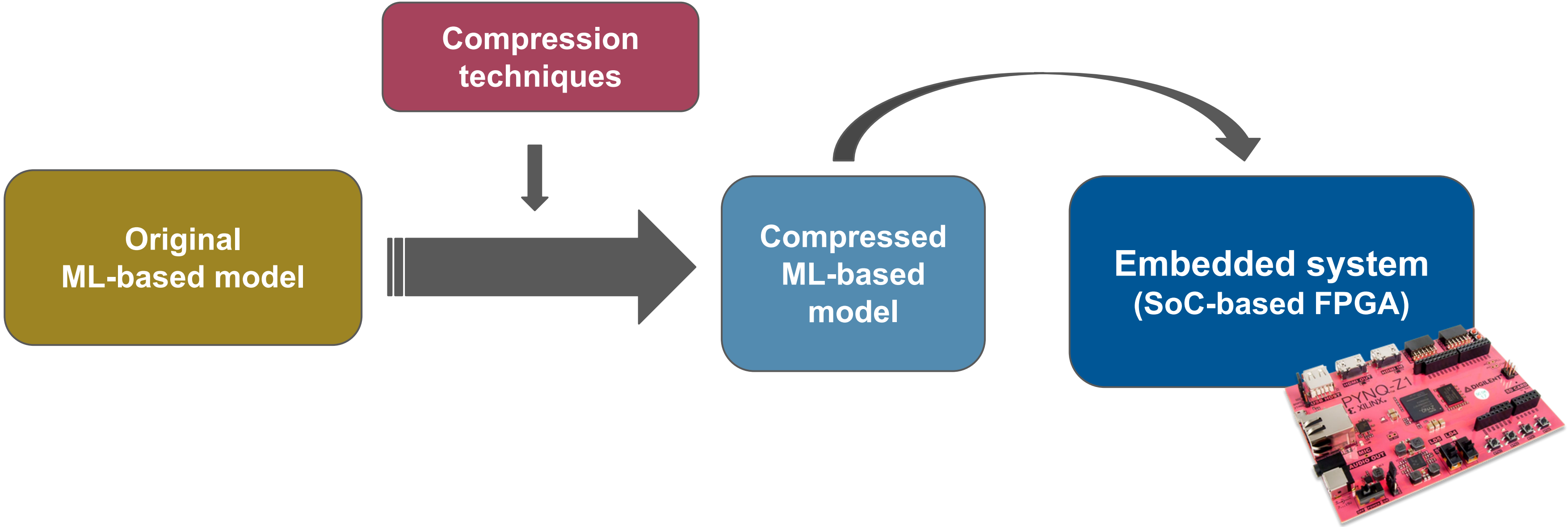
Edge AI based on FPGA



Edge AI based on FPGA



Edge AI based on FPGA



Remarks from the State-Of-The-Art

Remarks from the State-Of-The-Art

Memory footprint and latency

Remarks from the State-Of-The-Art

Memory footprint and latency

Ensemble of compression techniques

Remarks from the State-Of-The-Art

- Memory footprint and latency
- Ensemble of compression techniques
- On-chip memory deployment

Remarks from the State-Of-The-Art

Memory footprint and latency

Ensemble of compression techniques

On-chip memory deployment

End-to-end workflow

Remarks from the State-Of-The-Art

Memory footprint and latency

Ensemble of compression techniques

On-chip memory deployment

End-to-end workflow

Productivity

Remarks from the State-Of-The-Art

- *Question: What other features or challenges would you include?*

Optimizing every phase of the design and implementation process.

Optimizing every phase of the design and implementation process.

Software development

Model training

Model compression

Optimizing every phase of the design and implementation process.

Software development

Model training

Model compression

HW platform selection

Edge devices

Cloud

GPU/CPU

Others

Optimizing every phase of the design and implementation process.

Software development	HW platform selection	Firmware creation
Model training	Edge devices	C/C++ application
Model compression	Cloud	Python application
	GPU/CPU	HDL
	Others	

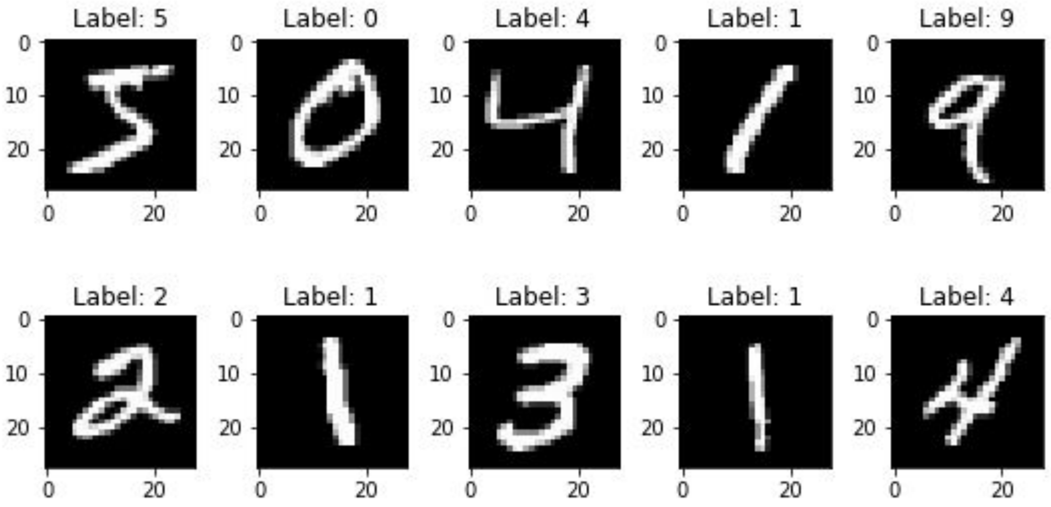
Optimizing every phase of the design and implementation process.

Software development	HW platform selection	Firmware creation	PCB design
Model training	Edge devices	C/C++ application	Key components
Model compression	Cloud	Python application	Signal and power integrity
	GPU/CPU	HDL	Methodological design
	Others		EMC and EMI

MNIST-based binary classification

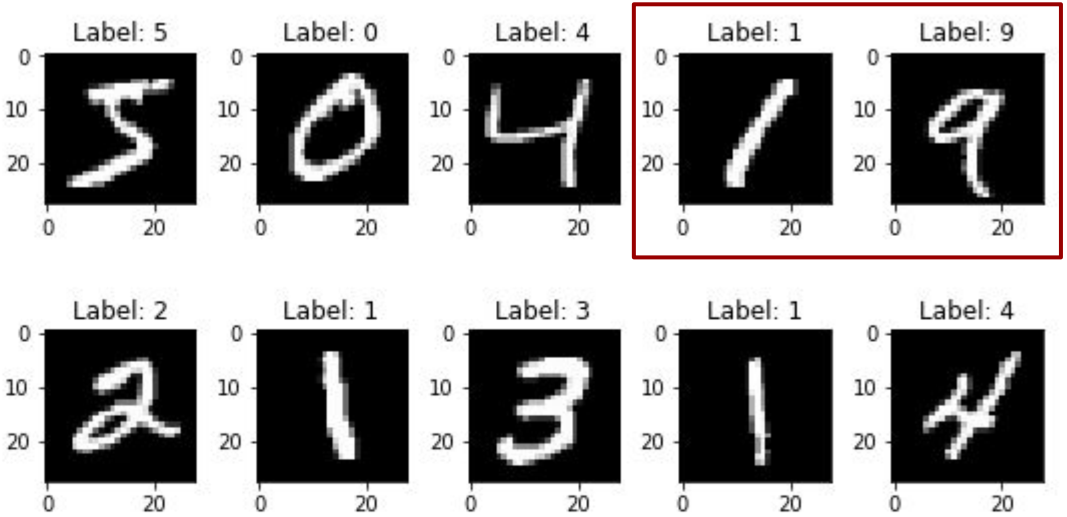
MNIST-based binary classification

MNIST-based
binary classification



MNIST-based binary classification

MNIST-based binary classification



MNIST-based binary classification

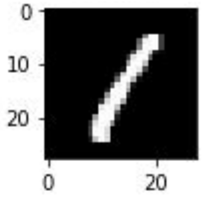
MNIST-based binary classification

- **Quantization-Aware Pruning**
  - 8-bits fixed point precision
  - 20% target sparsity
  - QKeras for model definition

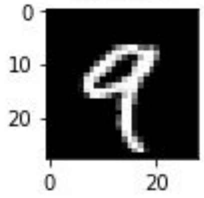
```
Layer (type)                 Output Shape              Param #
=================================================================
fc1_input (QDense)           (None, 5)                 3925
_________________________________________________________________
relu_input (QActivation)     (None, 5)                 0
_________________________________________________________________
fc1 (QDense)                 (None, 7)                 42
_________________________________________________________________
relu1 (QActivation)          (None, 7)                 0
_________________________________________________________________
fc2 (QDense)                 (None, 10)                80
_________________________________________________________________
relu2 (QActivation)          (None, 10)                0
_________________________________________________________________
output (QDense)              (None, 2)                 22
_________________________________________________________________
sigmoid (Activation)         (None, 2)                 0
=================================================================
Total params: 4,069
Trainable params: 4,069
Non-trainable params: 0
_________________________________________________________________
```

MNIST-based binary classification

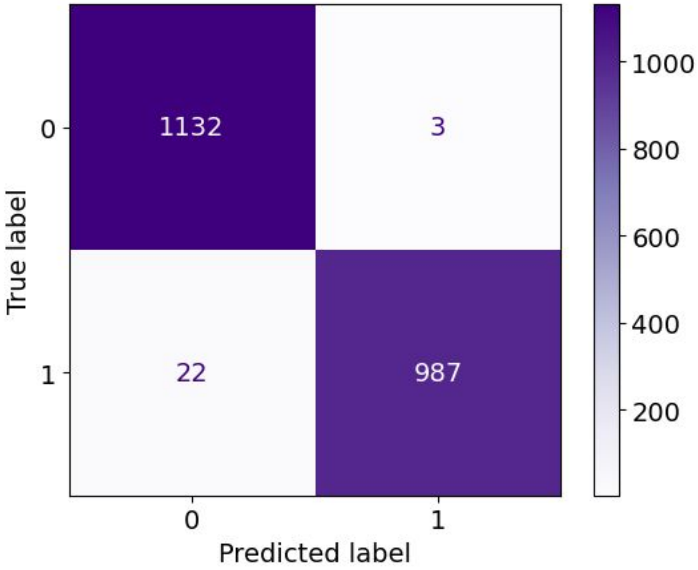
MNIST-based binary classification



Class 0

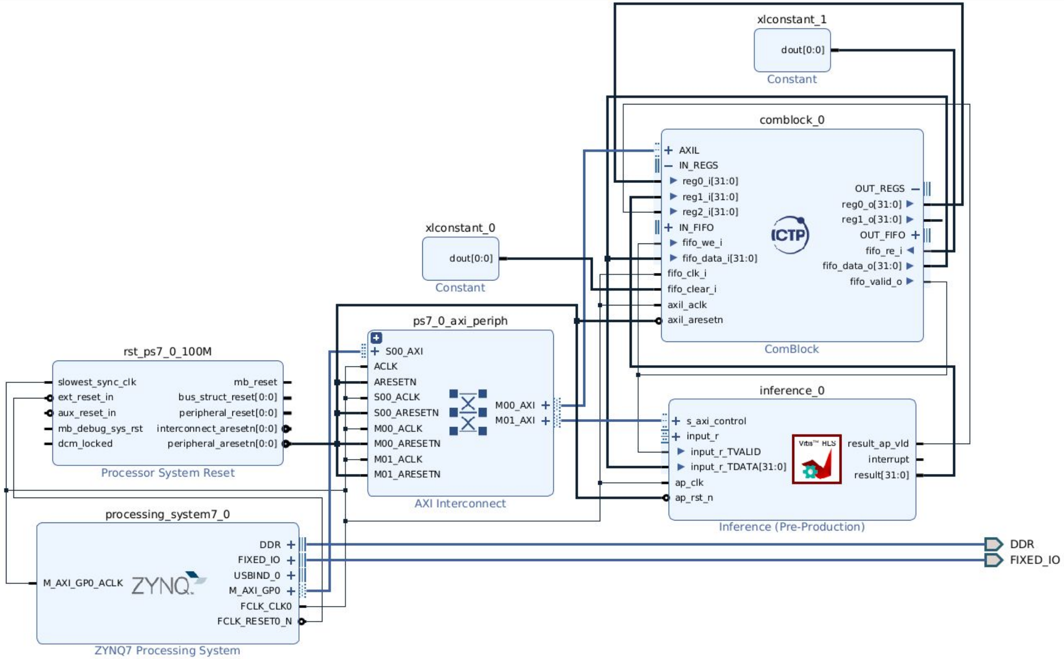


Class 1



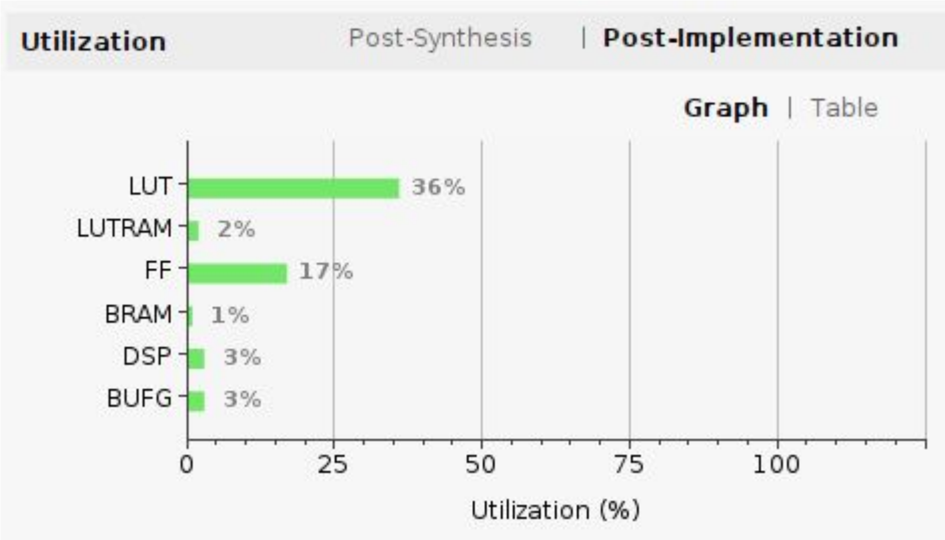
MNIST-based binary classification

MNIST-based
binary classification



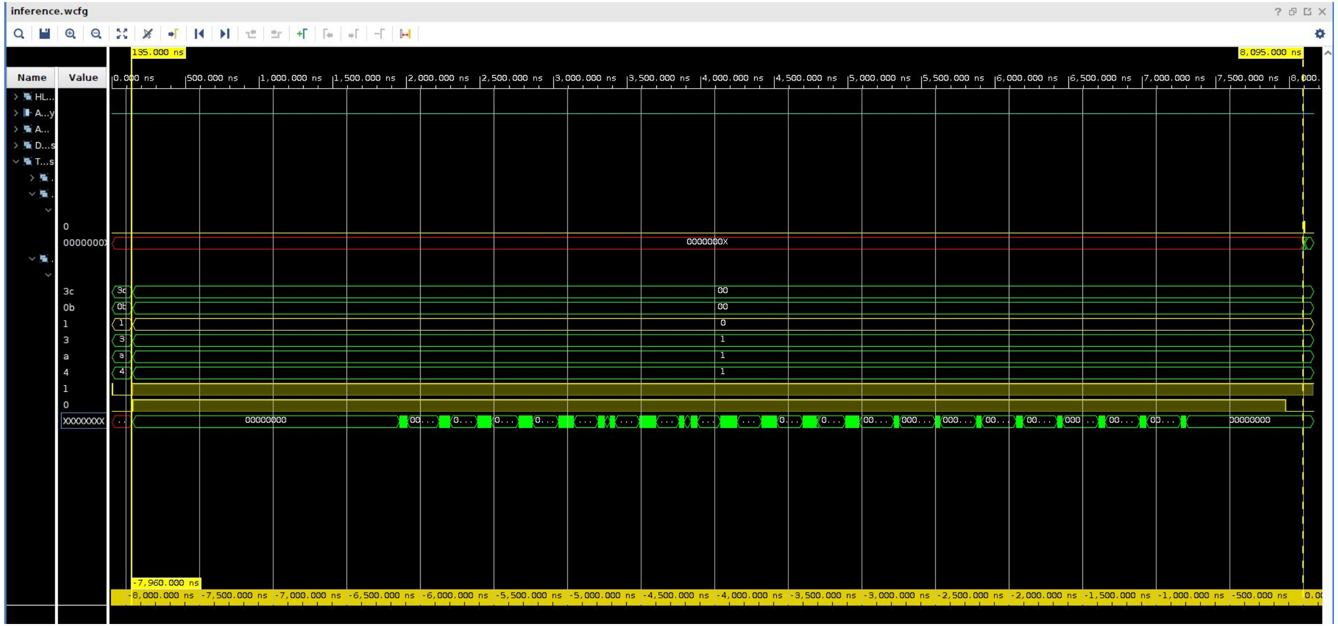
MNIST-based binary classification

MNIST-based
binary classification



MNIST-based binary classification

MNIST-based binary classification



MNIST-based binary classification

MNIST-based
binary classification

IP core based on ML integrated with PYNQ framework

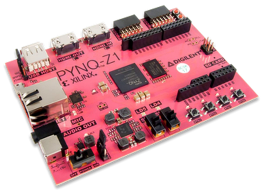
In [ ]:
```
from pynq import Overlay
from pynq import MMIO
import comblock as cbc

import numpy as np
import matplotlib.pyplot as plt
```

Load Overlay

In [ ]:
```
# Load the overlay (bitstream) onto the FPGA.

ol = Overlay("design_1_wrapper.xsa")
```

The information from the **comblock_0** block is read to verify everything that is obtained. Since the object is mapped to AXI Lite, it is noted that the AXI Full address is omitted.

In [ ]:
```
## Overlay information

ol.ip_dict
```



MNIST-based binary classification

MNIST-based
binary classification

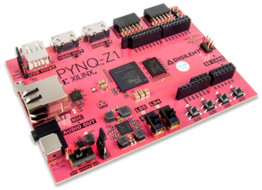
ComBlock information

ComBlock for PYNQ: https://github.com/Mballina42/PynQ_ComBlock

For convenience, the *comblock.py* Python script is established which contains useful constants for interacting with the ComBlock.

In [ ]:
```
ol.ip_dict['comblock_0']
```

In [ ]:
```
# The object is created based on the comblock_0 IP

cb = ol.comblock_0
```



PYNQ™

MNIST-based binary classification

MNIST-based
binary classification

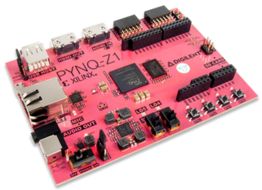
Data preparation

```
In [ ]: signal_1 = [

            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

          ]
```

```
In [ ]: imageArray = np.array(signal_1)

        image_2d = imageArray.reshape((28, 28))

        # Display as an image
        plt.imshow(image_2d, cmap='gray', interpolation='nearest')
        plt.colorbar()  # Optional: Show color scale
        plt.show()
```

```
In [ ]: signal_2 = [

            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

        ]
```

```
In [ ]: imageArray = np.array(signal_2)

        image_2d = imageArray.reshape((28, 28))

        # Display as an image
        plt.imshow(image_2d, cmap='gray', interpolation='nearest')
        plt.colorbar()  # Optional: Show color scale
        plt.show()
```



PYNQ™

MNIST-based binary classification

MNIST-based
binary classification

Interacting with ComBlock

Signal 1

Write FIFO - Send image to the FPGA

```
In [ ]: cb.write(cbc.OREG1, 1)

        # Send data to the ComBlock's FIFO
        data_size = 28*28
        for i in range(data_size):
                cb.write(cbc.OFIFO_VALUE, signal_1[i])
```

Read registers - Read inference result from the FPGA

```
In [ ]: # Read IREG1 to obtain the result of the inferece process
        cb.read(cbc.IREG1)
```

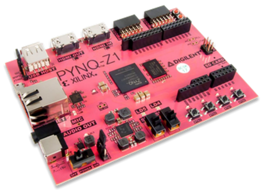
Signal 2

Write FIFO - Send image to the FPGA

```
In [ ]: cb.write(cbc.OREG1, 1)

        # Send data to the ComBlock's FIFO
        data_size = 28*28
        for i in range(data_size):
                cb.write(cbc.OFIFO_VALUE, signal_2[i])
```

Read registers - Read inference result from the FPGA

```
In [ ]: cb.read(cbc.IREG1)
```



PYNQ™

Machine Learning and FPGA:
Evolution, Current State of These Technologies, and Edge AI

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Perú - Online - 2025 -