Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Centro Nazionale di Ricerca in HPC, Big Data e Quantum Computing

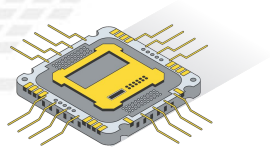
Giulio Bianchini, on behalf of Spoke 0 and Spoke 2 WP2.4

Sviluppo di firmware per acceleratori FPGA: esperienze e casi d'uso per il calcolo scientifico verso il paradigma cloud in ICSC
IFAE, Firenze, 5/04/2024

Introduzione

Acceleratori

Le crescenti sfide nel campo del calcolo scientifico richiedono hardware sempre più efficiente. Gli acceleratori hardware sono appositamente progettati per potenziare le prestazioni di workload specifici.



Central Processing Unit

CPU

Graphics Processing Unit

GPU

Field Programmable Gate Array

FPGA

Application-Specific Integrated Circuit

ASIC

Flessibilità, astrazione di programmazione

Performance, efficienza energetica

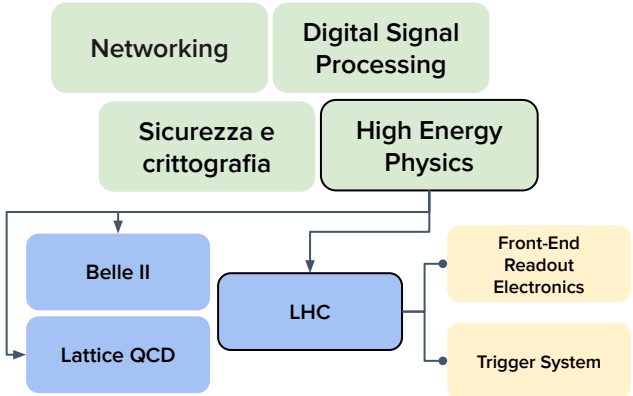
Field Programmable Gate Array

FPGA

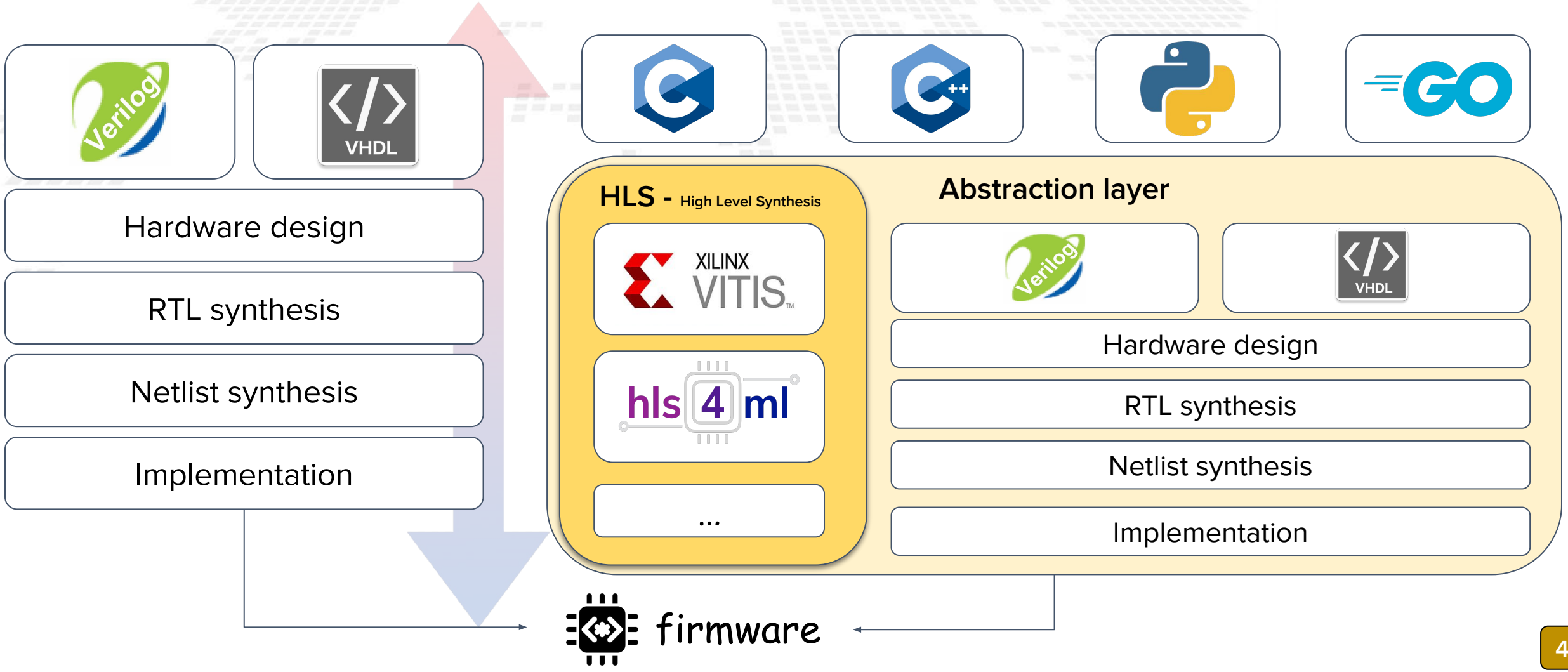
Circuito integrato la cui logica e' riprogrammabile

- Matrice di blocchi logici configurabili che operano in parallelo
- Interconnessioni e blocchi I/O programmabili
- Come si programma? **Firmware**

- Calcolo Parallelo
- Altamente specializzato
- A basso consumo energetico



FPGA - Firmware



Spoke 2 use cases - Ultra-fast algorithms running on FPGA

Due topic principali *1 Trigger, DAQ and on-line processing *2 Sviluppo FPGA tools

Projects	Institutions	Leaders
Development of algorithms based on neural networks and implementation on FPGAs, with application for trigger and anomaly detection at event level and object level for the Atlas experiment. *1	INFN and University of Genova, Napoli and Roma and Roma1	S.Giagu, V. Ippolito, C. Bini and E. Rossi
Development of a track reconstruction algorithm, at 30 MHz, on FPGA for LHC-b data acquisition. *1	University of Milano Bicocca and University of Pisa.	Maurizio Martinelli
Development of digital trigger logic for a “missing energy” experiment with a positron beam at CERN (POKER/NA64) *1	University and INFN Genova	Andrea Celentano
Development of quantum-inspired Tree Tensor Networks for classification in Trigger on FPGA *1	University of Padova and Politecnico di Milano	Jacopo Pazzini and Andrea Triossi
Di-tau trigger development for the CMS Level-1 trigger system *1	INFN and University of Milano Bicocca	Simone Gennai
Scouting and processing of Level-1 trigger data using FPGA to run on-the-fly momentum object calibration with ML based algorithms *1	INFN and University of Padova and INFN Milano Bicocca	Jacopo Pazzini, Andrea Triossi and Marco Zanetti
Development and testing of RDMA over converged ethernet (ROCE) on FPGA for data transfer from detectors’ front-end to computing servers *2	INFN and University of Padova	Jacopo Pazzini and Andrea Triossi
Development of a Customizable Framework for Multi-FPGA Accelerator Generation via architectures *2	**INFN and University of Perugia and University of Chieti-Pescara**	**Mirko Mariotti and Giulio Bianchini**

BondMachine use case

INFN

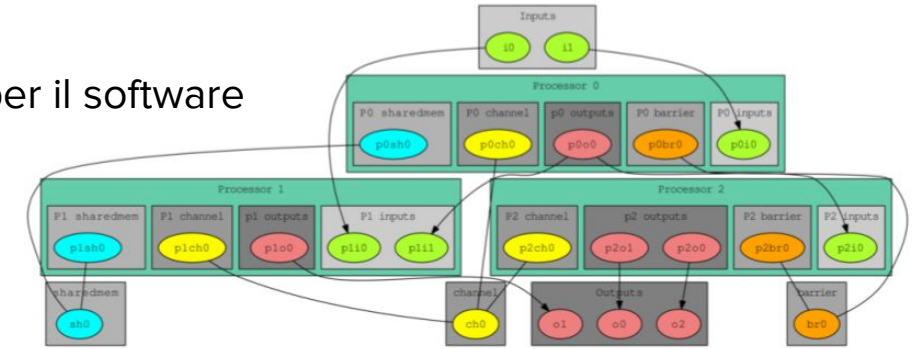
http://bondmachine.fisica.unipg.it/

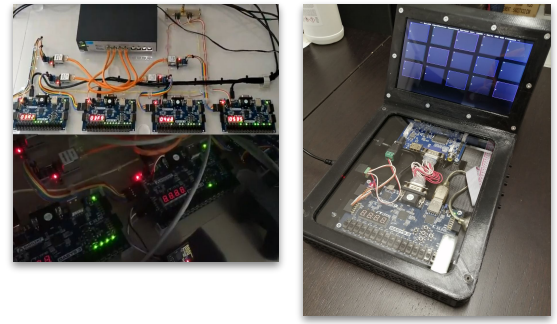
BondMachine

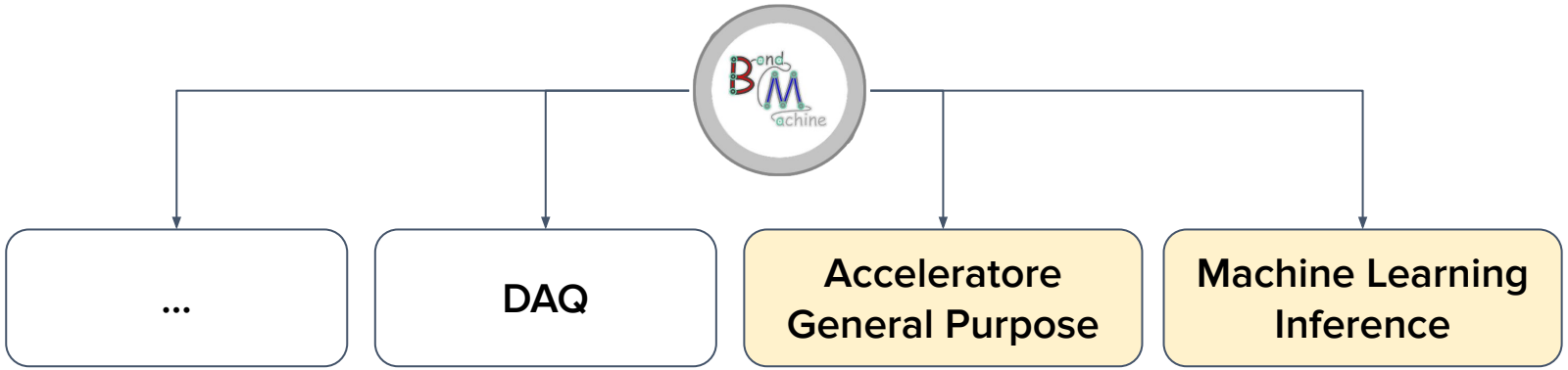
Un framework per costruire architetture di calcolo dinamiche

BondMachine è un ecosistema software Open Source (https://github.com/BondMachineHQ) per la generazione dinamica di architetture computazionali che possono essere sintetizzate su FPGA.

- Linguaggio di programmazione ad alto livello (Golang) sia per l'hardware che per il software
- Compilatore generatore di architetture
- Grafo computazionale e modelli di apprendimento automatico
- Indipendente dal venditore



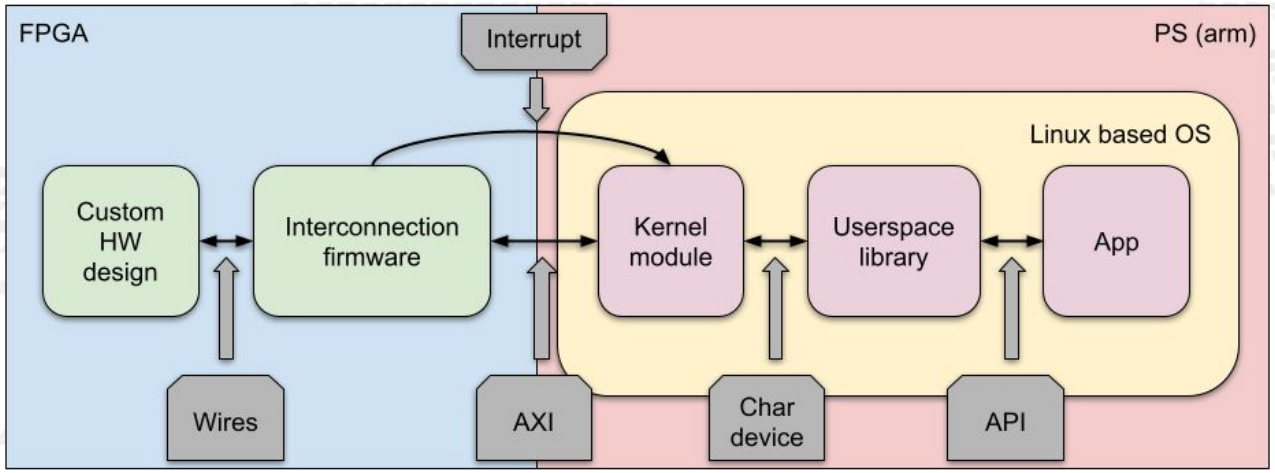


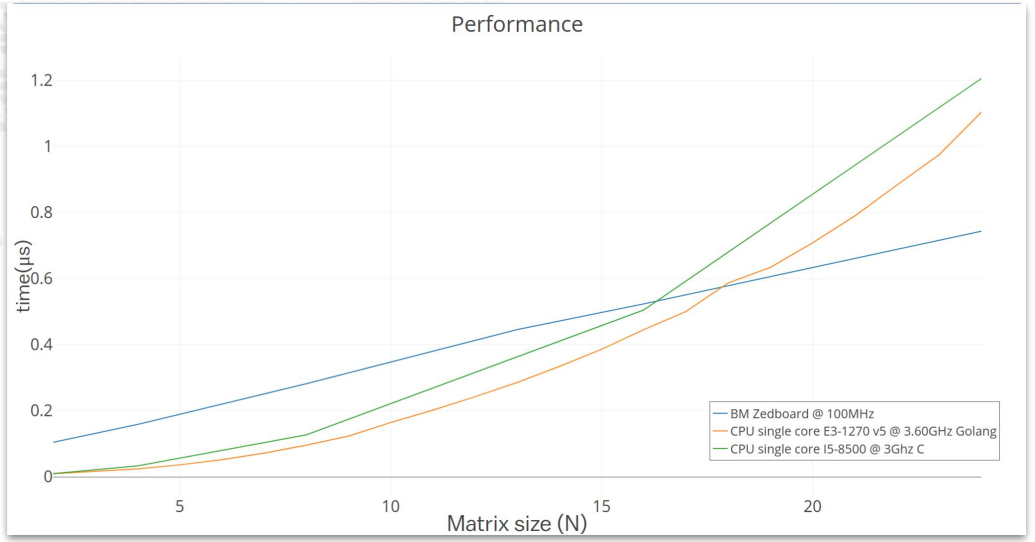


- Latenza
- Consumo/Efficienza energetica
- Utilizzo risorse

https://www.sciencedirect.com/science/article/abs/pii/S0167819121001150

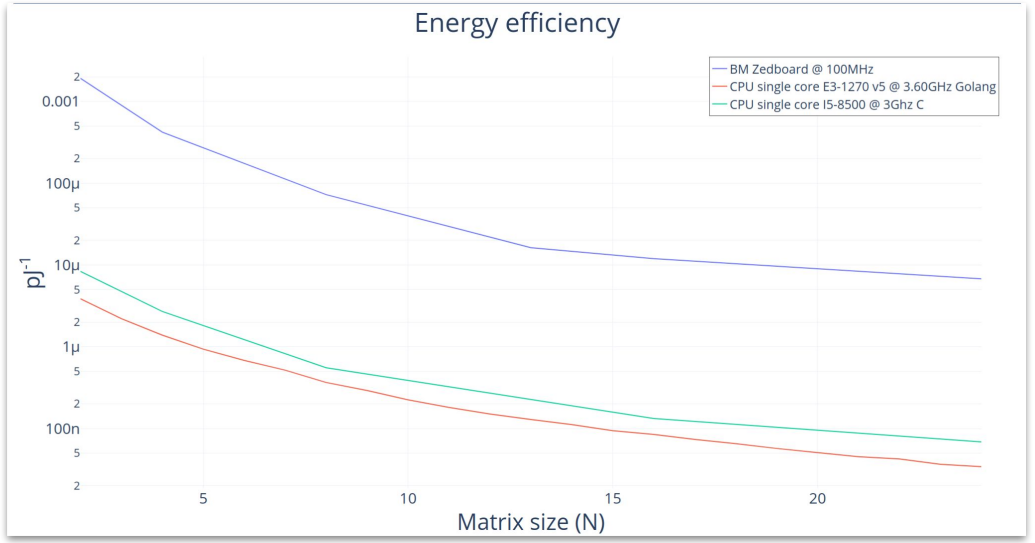
Sistema accelerato general purpose su processori ibridi (ZYNQ)





Creazione di un sistema accelerato su processori ibridi, in cui l'applicazione ad alto livello utilizza l'FPGA per ottimizzare l'esecuzione di specifici tasks, come ad esempio la moltiplicazione matriciale.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = [c_i]_{i=1}^{n} = \left[\sum_{k=1}^{n} a_{ik} b_k\right]_{i=1}^{n}$$



Machine Learning Inference

Il modello di machine learning è addestrato con framework standard e sintetizzato in FPGA come un grafo di processori eterogenei e interconnessi.

Codice ad alto livello

```
output_file = "modelBM.json"
output_path = os.getcwd()+"/tests/"

mlp_tf2bm(model, output_file=output_file, output_path=output_path)

prjHandler = BMProjectHandler("sample_project", "neuralnetwork",
"projects_tests")

prjHandler.check_dependencies()
prjHandler.create_project()

config = {
    "data_type": "float16",
    "register_size": "16",
    "source_neuralbond": f"{output_path}{output_file}",
    "flavor": "axist",
    "board": "zedboard"
}

prjHandler.setup_project(config)
prjHandler.build_firmware()
```

1. Addestramento modello
2. Conversione del modello addestrato
3. Creazione del progetto
4. Configurazione
5. Creazione del firmware

Il framework BM supporta molti tipi di FPGA anche di diversi venditori. Essendo anche un sistema multi-FPGA, l'obiettivo e' sfruttare le risorse di calcolo ICSC per testare casi d'uso sempre più' complessi.

Key points

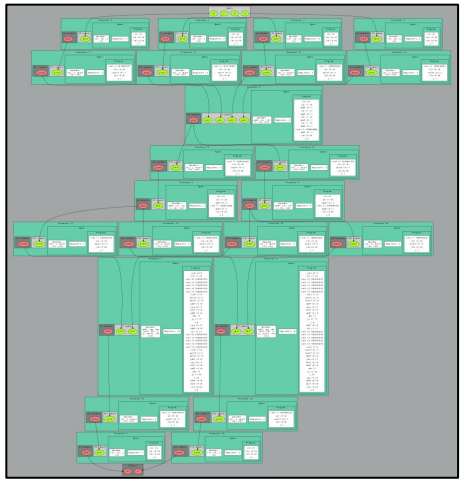
- Utilizzo ottimizzato delle risorse
- Altamente personalizzabile
- Utilizzabile ad alto livello
- Indipendente dal venditore
- User-friendly
- Estensione cloud

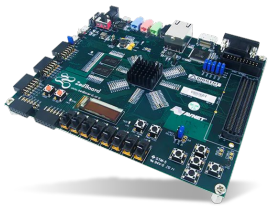
Machine Learning Inference

Implementare architetture personalizzate e parallele adattate a modelli di machine learning specifici, ottenendo velocità di elaborazione più rapide e minor consumo energetico rispetto alle tradizionali CPU o GPU.

Caso d'uso (semplice)

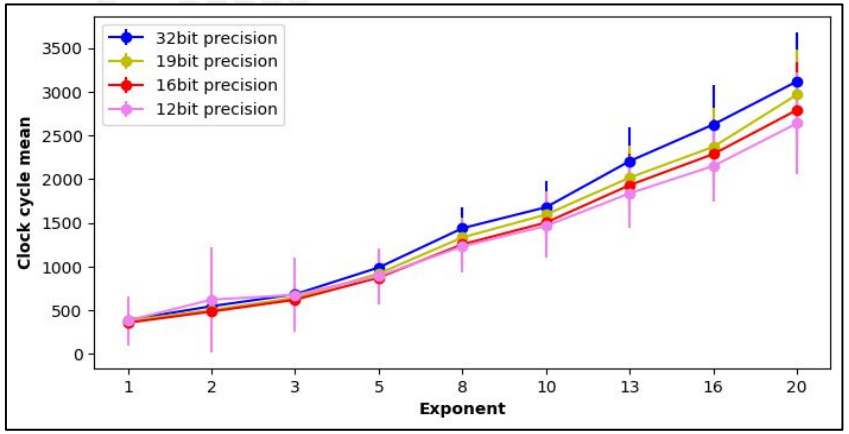
- Multi Layer Perceptron (MLP) totalmente connesso
- Dataset 4 features classificazione binaria





FPGA utilizzata: **ZedBoard**

Bit	Luts	Luts %	Regs	Regs %
32	14306	26.8%	9264	8.7%
19	7202	13.5%	5717	5.3%
16	7738	14.5%	5487	5.1%
12	4133	7.7%	5094	4.7%



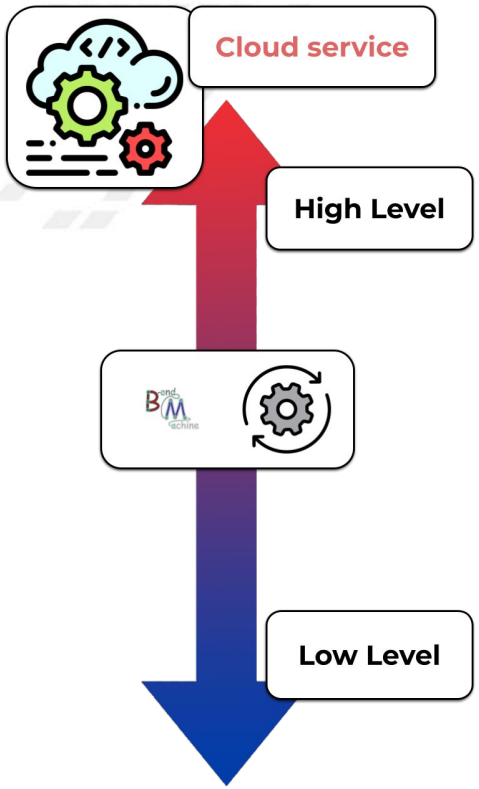
Bit	Static-Power (W)	Dynamic-Power (W)	Time / Inf (s)	En. / Inf (J)
32	0.037	0.055	6.84E-06	3.78E-07
19	0.013	0.022	6.44E-06	1.39E-07
16	0.017	0.024	6.21E-06	1.49E-07
12	0.02	0.012	6.76E-06	8.11E-08

CPU	Time / Inf (s)	En. / Inf (J)
ARM Cortex A9	10E-02	10E-06
Intel i7-1260P	10E-06	10E-04

A

Acceleratori in Cloud

Mettere a disposizione il sistema come servizio cloud, sfruttando appieno i vantaggi offerti dal paradigma del cloud computing.



Generazione di firmware per il calcolo accelerato (Low Level)

- Personalizzabile in base al tak

Facile da usare per l'utente (High level)

- Automatismi per la generazione del firmware e del Jupyter Notebook per interagire con la parte a basso livello

Vogliamo spostarci ancora più in alto per astrarre ancora di più la complessità realizzando un **servizio cloud**

Vantaggi

Facilità di utilizzo e flessibilità

Poter distribuire il proprio algoritmo su FPGA senza preoccuparsi di 'dove' si trovano le risorse

Accesso e gestione democratica delle risorse

Sfruttando gli strumenti nativi cloud/Kubernetes, è possibile riutilizzare un metodo consolidato per orchestrare le risorse e la distribuzione dei workloads.

Facilità di prototipizzazione

Automatizzazione del processo di compilazione e caricamento del firmware. Il framework gestisce i dettagli specifici della board richiesta.

B

Inference as a Service e Firmware as a Service

Effettuare inferenza su FPGA da remoto come servizio cloud

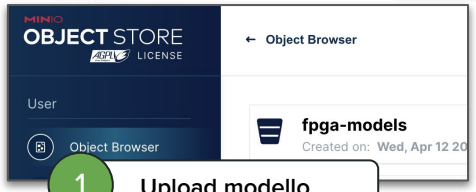
0 Addestramento modello

1 Upload del modello bucket S3

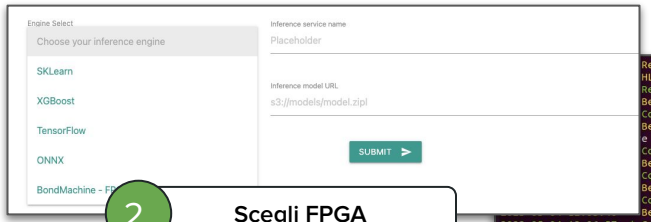
2 Scelta del motore HLS (i.e. BondMachine) e target board

3 Invio della richiesta e attesa

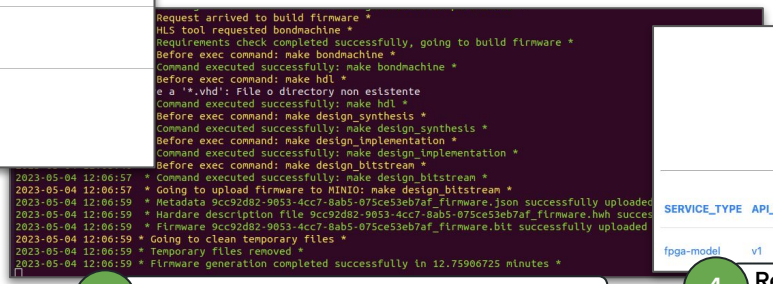
4 Endpoint ML inference



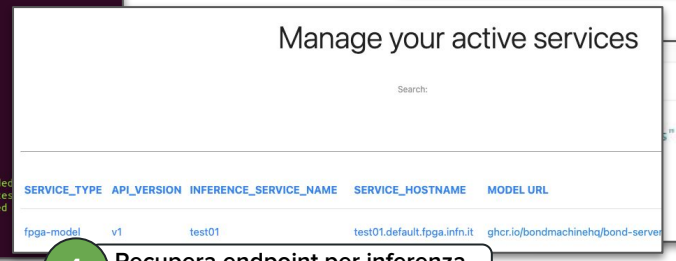
1 Upload modello



2 Scegli FPGA

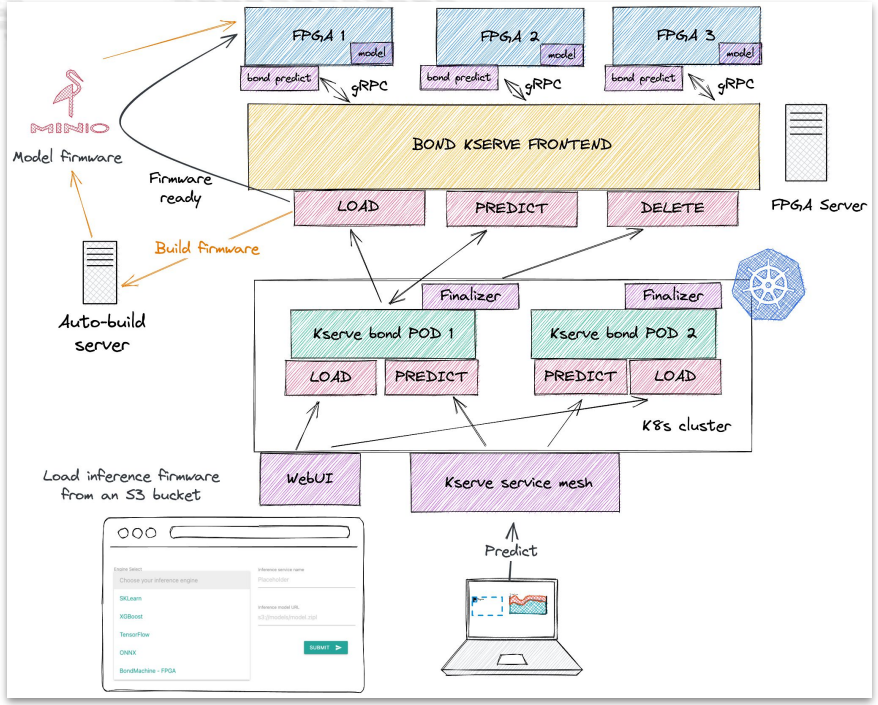


3 Attendi generazione firmware FPGA

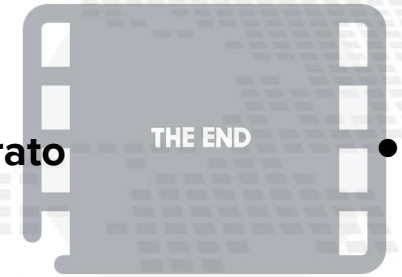


4 Recupera endpoint per inferenza remota

5 Esegui inferenza



Riepilogo



- **Generazione di firmware per il calcolo accelerato**
  - General purpose
  - Machine Learning Inference
  - Personalizzabile rispetto al task
- **Indipendente dal venditore**
  - AMD, Intel, Lattice
- **Trasparente all'utente**
  - Automatismi semplificano il processo di generazione del firmware
  - Librerie ad alto livello nei principali linguaggi di programmazione
  - Simulazione dell'architettura
- **Cloud**
  - Creare il proprio firmware senza preoccuparsi di avere le risorse necessarie
  - Condividere firmware nella comunità
  - Facilitare lo sviluppo e prototipazione
  - Utilizzare FPGAs da remoto

Sfide future

- **Per ML Inference**
  - Reti convolutive
  - Aggiungere nuove tecniche di ottimizzazione
- **Verso cluster di FPGAs (BM e' un sistema multi-FPGA)**
  - Integrare nuove board
  - Test su risorse ICSC (Spoke0)
- **Per l'implementazione cloud..**
  - Valutare altri casi d'uso oltre l'inferenza
  - Effettuare misurazioni ai vari step della flusso
  - FPGA provisioning attraverso servizi cloud

Website	*http://bondmachine.fisica.unipg.it/*
Paper	*https://www.sciencedirect.com/science/article/pii/S0167819121001150*
CHEP 2023	*https://indico.jlab.org/event/459/contributions/11826/*
CCR 2023	*https://agenda.infn.it/event/34683/contributions/197368/*
CCR 2022	*https://agenda.infn.it/event/30202/contributions/168531/*
InnovateFPGA 2018 Iron Award	*https://github.com/innovatefpga/2018-EM083*
Main repo	*https://github.com/BondMachineHQ*
ML inference on cloud repo	*https://github.com/BondMachineHQ/kserve-bond-extension.git*

D

Grazie per l’attenzione!