

EduFPGA-AI: Educational Platforms and Frameworks for Learning AI on FPGA Hardware

Mohamed Abdo

Hamm-Lippstadt University of Applied Sciences

Lippstadt, Germany

mohamed-sayed-mohamed.abdo@stud.hshl.de

Abstract—The intersection of artificial intelligence (AI) and field-programmable gate array (FPGA) technology has given rise to novel educational methodologies for hardware-accelerated machine learning. This paper provides a comprehensive review of educational platforms and frameworks that facilitate the implementation of AI algorithms on FPGA hardware. The architectural advantages of FPGAs for AI applications are examined, including reconfigurability, parallelism, low latency and power efficiency. The paper provides a systematic review of important deployment frameworks, such as PYNQ, Vitis AI, HLS4ML, FINN and OpenVINO, analysing their educational value and practical implementation workflows. Moreover, the discussion encompasses quantisation techniques that facilitate the effective implementation of models on FPGA platforms. These techniques have been shown to reduce model sizes by up to 75%, while maintaining or preserving inference accuracy. The survey includes recommendations for FPGA families that are suitable for educational use, ranging from edge devices such as Zynq UltraScale+ to data centre accelerators such as Alveo. This work contributes to the growing body of research on FPGA-AI education by providing educators and researchers with a structured overview of the available tools, methodologies and best practices for teaching hardware-accelerated AI in academic settings.

Index Terms—Artificial Intelligence, FPGA, Hardware Acceleration, Machine Learning Education, Quantization, PYNQ, Vitis AI, Edge Computing

I. INTRODUCTION

The rapid evolution of artificial intelligence (AI) and machine learning (ML) has created an increasing demand for specialised hardware capability of executing complex neural network computations efficiently and with low latency. Furthermore the traditional processors such as central processing units (CPUs) and graphics processing units (GPUs) have dominated AI deployment, field-programmable gate arrays (FPGAs) offer unique advantages in production systems and educational environments alike too. FPGAs provide hardware reconfigurability, enabling custom digital circuits to be implemented for specific AI workloads by eliminating the need for new silicon fabrication. This flexibility makes FPGAs particularly well-suited to academic AI settings, where students can experiment with various AI architectures and optimisation techniques, starting with simple models up to Data-centers FPGAs.

Educational platforms for FPGA-AI have emerged as critical tools for bridging the gap between theoretical machine learning concepts and practical hardware implementa-

tion. These platforms reduce the barriers to entry for students and researchers by offering high-level abstractions while still providing access to low-level hardware optimisation. The importance of FPGA education in hardware-accelerated AI is growing as industry demand for engineering students with expertise in both AI and hardware continues to increase. This paper surveys the current FPGA-AI educational platform landscape, providing educators and researchers with a comprehensive resource for implementing hardware-accelerated AI with FPGA.

II. FPGA ARCHITECTURE FOR AI APPLICATIONS

A. Standalone FPGA Architecture

FPGAs consist of three primary components: Starting with programmable logic blocks (PLBs), programmable interconnects, and hardened components. PLBs contain thousands to millions of identical units that comprising basic digital components, such as look-up tables (LUTs), flip-flops and logic gates. These blocks serve as the fundamental building blocks for implementing any digital circuit, eliminating the need for new silicon fabrication. The programmable interconnects form a re-routable wiring network that can connect the logic blocks in any configuration. This enables the creation of custom circuit pathways that are optimised for specific AI algorithms.

Modern FPGAs incorporate hardened components that provide fixed-function blocks for improved efficiency. These include digital signal processing (DSP) slices optimised for mathematical operations, block RAM (BRAM) for high-speed memory access and dedicated AI engines in advanced devices such as the AMD Versal. Unlike CPUs with fixed hardware architectures and GPUs with fixed-architecture processors, FPGAs are fully reconfigurable, enabling the hardware to adapt to evolving AI models and algorithms.

B. SoC FPGA Hybrid Architecture

System-on-Chip (SoC) FPGAs combine ARM processors with FPGA fabric on a single chip, to create a hybrid architecture which optimizes power efficiency and processing capabilities. In such devices, the CPU handles its own software tasks while the FPGA accelerates computationally intensive workloads. This division of labor is particularly effective for AI applications where certain operations benefit from hardware AI acceleration. A great example of this architecture is "The Zynq UltraScale+ MPSoC", which is widely used

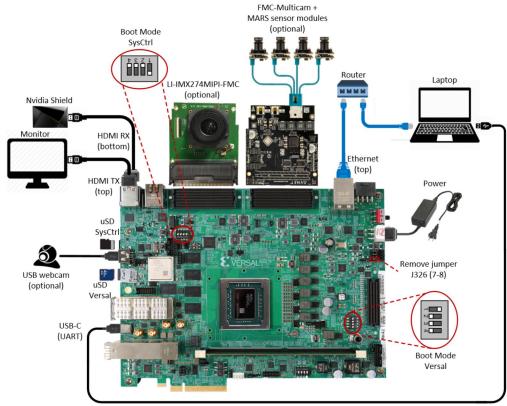


Fig. 1. AMD Versal Adaptive SoC featuring AI Engines and programmable logic for AI acceleration

in educational settings due to its balanced performance and accessibility [1].

C. High-End FPGA for Data Center Acceleration

For data center applications, high-end FPGAs such as Alveo accelerator cards provide massive computational resources. These devices feature the largest capacity FPGAs available, often equipped with high-bandwidth memory (HBM) offering 16-32GB capacity with bandwidth exceeding 460 GB/s [2]. The combination of massive parallel processing capabilities and ultra-fast memory access makes these platforms ideal for deploying large-scale AI models in cloud environments, providing valuable learning opportunities for students studying data center AI acceleration.

III. QUANTIZATION: THE MATHEMATICAL FOUNDATION

Quantization is a crucial technique for deploying AI models on FPGAs efficiently. It converts 32-bit floating-point parameters to lower-precision representations like 8-bit integers, dramatically reducing memory footprint and computational requirements. The quantization process can be mathematically expressed as:

$$Q(x) = \text{round}\left(\frac{x}{\Delta}\right) \times \Delta + Z \quad (1)$$

where:

- x is the floating-point value to be quantized
- Δ is the quantization step size (scale factor)
- Z is the zero-point (integer bias)
- $Q(x)$ is the quantized integer value

For symmetric quantization commonly used in FPGA implementations, $Z = 0$, and the range is symmetric around zero. The scale factor Δ is calculated as:

$$\Delta = \frac{\max(|x|)}{2^{b-1} - 1} \quad (2)$$

where b is the target bit-width (typically 8 bits).

That Quantization reduces the model sizes by approximately 75% (from 100 MB up to 25 MB for a typical models)

while maintaining the AI model accuracy within acceptable bounds. The computational benefits are substantial: 8-bit integer operations require 4x less memory bandwidth and can execute faster on FPGA hardware optimized for fixed-point arithmetic instead of the float format. This efficiency enables 2-4x acceleration compared to floating-point implementations while consuming significantly less power and achieving high efficiency.

IV. EDUCATIONAL FRAMEWORKS AND PLATFORMS

A. Vitis AI Framework

Xilinx's Vitis AI development stack provides comprehensive tools for AI inference on Xilinx hardware platforms, supporting both edge devices and data centre cards. The framework incorporates a model optimiser, an AI compiler, an AI quantiser, and an AI profiler, thereby providing students with experience of industrial-grade development tools. Vitis AI is compatible with prominent neural network (NNs) frameworks, including TensorFlow, PyTorch, and Caffe, facilitating a seamless transition from model development to hardware deployment.

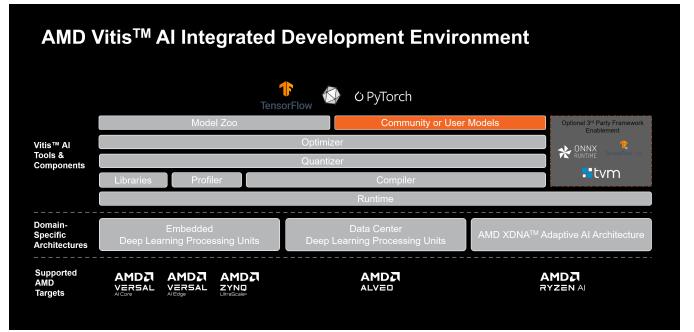


Fig. 2. Vitis AI development workflow from model to deployment

Recommended Hardware: Vitis AI has been demonstrated to provide support for a wide range of AMD Xilinx platforms including Zynq UltraScale+ MPSoCs for edge applications, Alveo accelerator cards for data centres, and Versal Adaptive SoCs for AI-optimised workloads. In the context of educational pursuits, the Zynq UltraScale+ ZU3EG. This may be regarded as an excellent combination of performance, cost-effectiveness and comprehensive documentation.

B. hls4ml Framework

The HLS4ML (High-Level Synthesis for Machine Learning) framework facilitates the translation of machine learning models into FPGA (field-programmable gate array) firmware through the process of high-level synthesis. The framework has been demonstrated to be particularly adept at applications requiring ultra-low latency, such as those utilising high-energy physics experiments. The programme's open-source nature and active community have rendered it a particularly suitable resource for academic research and education. The framework under discussion facilitates quantization-aware training and offers a range of optimisation options, enabling students to

investigate trade-offs between resource utilisation, latency, and accuracy.

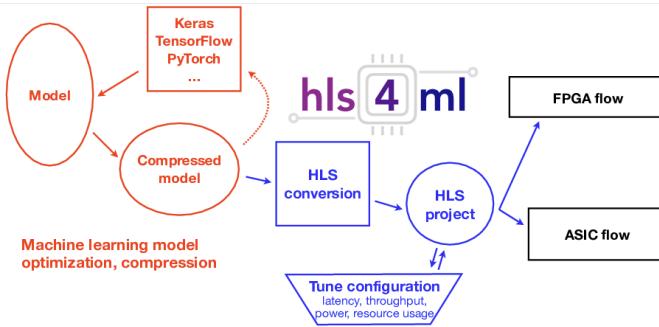


Fig. 3. hls4ml conversion pipeline from ML models to FPGA firmware

Recommended Hardware: HLS4ML is compatible with a variety of FPGA platforms; however, it is especially efficacious when utilised with the Zynq-7000 and Zynq UltraScale+ devices. The PYNQ-Z2 board is an optimal initiation into the field, offering both cost-effectiveness and comprehensive PYNQ assistance for accelerated prototyping. This is achieved by uploading a bitstream file, with no need even for VHDL or Verilog.

C. PYNQ Framework and DPU Integration

The PYNQ (Python Productivity for Zynq) framework signifies a substantial enhancement in the accessibility of FPGA for the purposes of AI education. PYNQ facilitates the utilisation of Python programming for Zynq devices through the medium of Jupyter notebooks. This enables students to implement AI applications without the prerequisite knowledge of hardware description language (HDL). This approach has been demonstrated to significantly reduce the learning curve, while ensuring that hardware acceleration remains available.

A fundamental element of PYNQ's AI capabilities is the Deep Learning Processing Unit (DPU), Xilinx's configurable AI accelerator IP that can be integrated into FPGA designs. The DPU provides:

- **Massive parallelism** for convolutional layers, pooling, batch normalization, and activation functions
- **Customizable architecture** adaptable to different AI workloads
- **Real-time performance** with 20x–100x faster inference compared to pure ARM CPU implementations
- **Lower latency and power consumption** ideal for edge AI devices

The integration with PYNQ is particularly elegant: educators can load a bitstream containing the DPU and then use pre-built Python APIs to load AI models (in .xmodel format), feed input tensors, and receive inference outputs—all without writing HDL or Vitis AI C++ code. This makes complex AI acceleration accessible to students with primarily software backgrounds.

Recommended Hardware: PYNQ officially supports numerous boards including PYNQ-Z1/Z2, Ultra96-V2, ZCU104,

and Kria KV260. For classroom settings, the PYNQ-Z2 offers the best combination of cost, documentation, and community support. However as startup and average price for students Ultra96-V2 is very good to starts with in AI application.

D. FINN Framework

The FINN (Fast Inference on Neural Networks) framework from Xilinx Research Labs is a dataflow compiler specifically designed for quantized neural network inference on FPGAs. The present framework is distinguished from traditional frameworks in that it does not map neural networks to existing hardware resources. Rather, it generates custom streaming architectures that are optimised for each model. This approach has been demonstrated to provide exceptional throughput and efficiency, particularly for binary and ternary neural networks.

Recommended Hardware: FINN has been developed for Xilinx UltraScale+ devices and is particularly well-suited for research-oriented courses. The Alveo U250 accelerator card provides the necessary resources for the exploration of large-scale quantized networks, while Zynq UltraScale+ devices, such as the ZCU102, are well-suited for edge-focused studies.

E. OpenVINO Framework

Intel's OpenVINO (Open Visual Inference and Neural Network Optimization) toolkit supports AI deployment across heterogeneous platforms including CPUs, GPUs, and FPGAs. Its cross-platform capabilities make it valuable for teaching comparative analysis of acceleration technologies. The 2025.4 release continues to expand FPGA support, particularly for Intel Agilex and Stratix devices.

Recommended Hardware: While OpenVINO supports multiple platforms, its FPGA capabilities are best demonstrated on Intel Development Kits featuring Agilex SoC FPGAs with integrated AI capabilities.

F. SensiML and RISC-V AI Ecosystem

SensiML's Picolo AI OS offers a complementary approach focused on edge AI development, providing a lightweight environment compatible with RISC-V and ARM Cortex-M processors. What makes SensiML particularly interesting for education is its support for the emerging RISC-V ecosystem, which represents an open-source alternative to proprietary processor architectures.

The RISC-V AI ecosystem is gaining momentum as an educational platform because:

- It provides **complete architectural transparency**—students can examine processor designs from RTL to software
- The **open-source nature** eliminates licensing barriers for academic institutions
- **Custom extensions** can be added for AI-specific instructions through the RISC-V Vector (V) extension and custom instruction set extensions
- It fosters understanding of **hardware-software co-design** principles

- RISC-V's modular architecture allows for specialized AI accelerators to be integrated alongside processor cores

According to industry analysis, RISC-V's open standard instruction set architecture is increasingly being adopted for AI and machine learning applications in embedded systems due to its flexibility and customizability. This makes it particularly suitable for educational environments where students can experiment with both processor design and AI algorithm optimization.

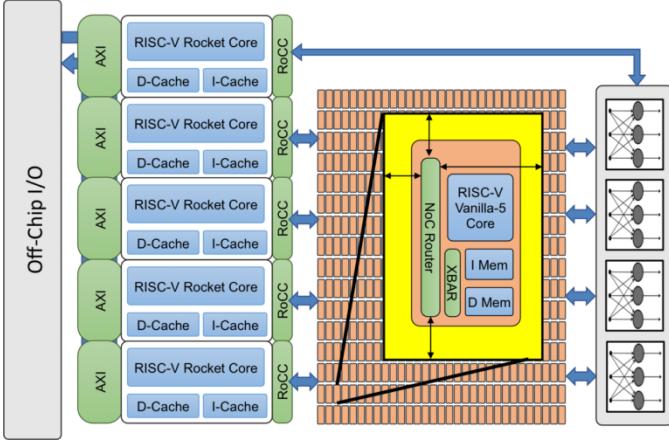


Fig. 4. RISC-V processor core with AI acceleration extensions

Recommended Hardware: Several FPGA boards support RISC-V implementations suitable for AI education. The Digilent Arty A7 with a soft-core RISC-V processor offers an affordable entry point, while more capable systems like the SiFive HiFive Unleashed provide higher performance for complex AI workloads. For advanced courses, platforms like the Microchip PolarFire SoC FPGA with hardened RISC-V cores offer industry-relevant experience with commercial RISC-V implementations.

V. DATASETS AND EDUCATIONAL PROJECTS

Effective FPGA-AI education relies on appropriate datasets and structured projects that demonstrate practical implementation. Commonly used datasets include MNIST for basic image classification, CIFAR-10 and CIFAR-100 for more complex computer vision tasks, and various TensorFlow Datasets for diverse applications. These datasets are well-documented, publicly available, and appropriately scaled for educational FPGA implementations.

Educational projects typically progress from simple implementations like MNIST digit classification on PYNQ boards to more complex applications such as real-time object detection on Kria system-on-modules. Intermediate projects might involve comparing quantization techniques using hls4ml or implementing custom neural network layers in FINN. By structuring projects around these established datasets and gradually increasing complexity, educators can create comprehensive learning pathways that develop both theoretical understanding and practical skills.

VI. CASE STUDIES AND IMPLEMENTATION PROJECTS EXAMPLES

A. MNIST Digit Classification on PYNQ-Z2

A classic introductory project involves implementing MNIST digit classification on the PYNQ-Z2 platform. This case study typically begins with training a simple convolutional neural network (CNN) in Python using TensorFlow or PyTorch, achieving approximately 98-99% accuracy on the test set. The model is then quantized to 8-bit integers using the TensorFlow Lite or PyTorch quantization tools, reducing the model size from around 2-3 MB to 500-750 KB.

The deployment workflow includes converting the quantized model to a format compatible with the DPU (Deep Learning Processing Unit) using the Vitis AI tools. Students load the generated .xmodel file onto the PYNQ-Z2 board and use the PYNQ Python APIs to perform inference. This hands-on exercise demonstrates the complete pipeline from software training to hardware deployment while highlighting the trade-offs between accuracy, model size, and inference latency. Typical results show 20-50x acceleration compared to running inference on the ARM processor alone, with latency reductions from 50-100 ms to 2-5 ms per image.

B. Real-time Object Detection on Kria KV260

For intermediate courses, implementing real-time object detection on the Kria KV260 provides valuable experience with computer vision applications. Using pre-trained models like MobileNet-SSD or YOLOv3-tiny, students learn to adapt models for FPGA deployment. The Kria platform's pre-built vision pipelines and accelerated processing simplify implementation while maintaining educational value.

Key learning outcomes include understanding model optimization techniques such as channel pruning, layer fusion, and precision calibration. Students experiment with different quantization strategies (post-training vs. quantization-aware training) and measure their impact on both accuracy and frame rate. The project typically achieves 15-30 FPS for 640x480 resolution video while maintaining reasonable accuracy (mAP of 0.6-0.7 for COCO classes), demonstrating the practical benefits of FPGA acceleration for real-time applications.

C. Low-Power Audio Classification on Ultra96-V2

Edge AI applications benefit significantly from FPGA acceleration, particularly in power-constrained environments. A case study involving audio classification on the Ultra96-V2 showcases these advantages. Using datasets like Google's Speech Commands, students implement keyword spotting with lightweight neural networks (typically depthwise separable convolutions or temporal convolutions).

The project emphasizes power efficiency measurements, comparing FPGA implementations against microcontroller (ARM Cortex-M) and GPU alternatives. Results typically show 10-20x lower power consumption for similar inference latency, with the FPGA solution achieving 1-2 mJ per inference compared to 10-30 mJ for software implementations. This

case study reinforces the importance of hardware-aware algorithm design and introduces students to power profiling tools and optimization techniques specific to edge deployments.

D. Research Project: Accelerated GAN Inference on Alveo U50

For advanced or graduate-level courses, implementing accelerated Generative Adversarial Network (GAN) inference on Alveo accelerator cards demonstrates data center AI capabilities. Students work with models like StyleGAN or CycleGAN, focusing on throughput optimization rather than just latency reduction.

This project introduces concepts like pipeline parallelism, memory hierarchy optimization, and HBM utilization. Using the Vitis AI development flow, students learn to partition models across multiple compute units and optimize data movement patterns. Performance analysis includes measuring throughput (images/second), power efficiency (images/Joule), and comparing against GPU baselines. Typical results show 2-3x higher throughput per watt compared to high-end GPUs for certain GAN architectures, highlighting FPGA advantages for specific workload patterns.

VII. HARDWARE SELECTION GUIDE FOR EDUCATORS

Selecting appropriate hardware platforms for FPGA-AI education requires balancing technical capabilities, educational objectives, and resource constraints. Different platforms serve distinct pedagogical needs, from introductory concepts to advanced research. Table I presents a framework for matching platforms to educational use cases.

For foundational courses introducing FPGA-AI concepts, the PYNQ-Z2 offers the most accessible starting point with its Python-centric approach and strong community support. Intermediate courses focusing on specialized applications can benefit from platforms like the Ultra96-V2 for edge computing or Kria KV260 for computer vision. Advanced curricula requiring professional-grade capabilities may utilize ZCU104 boards, while research-oriented programs can leverage Alveo accelerator cards for data center AI exploration. This tiered approach allows educators to progressively introduce hardware complexity while maintaining pedagogical clarity.

VIII. CURRICULUM DEVELOPMENT GUIDELINES

Developing effective FPGA-AI curricula requires careful consideration of prerequisite knowledge, learning objectives, and available resources. A suggested progression begins with fundamental concepts: digital logic basics, neural network theory, and Python programming. Subsequent modules introduce FPGA architecture, hardware description languages (with emphasis on high-level synthesis alternatives), and quantization principles.

Practical laboratory sessions should follow a scaffolded approach, starting with pre-built examples on PYNQ boards and gradually increasing student autonomy. Intermediate assignments might involve modifying neural network architectures for FPGA implementation, while advanced projects could

TABLE I
HARDWARE PLATFORM SELECTION GUIDE FOR FPGA-AI EDUCATION

Platform	Educational Use Cases and Strengths
PYNQ-Z2	<ul style="list-style-type: none"> Ideal for: Introductory courses, first exposure to FPGA-AI concepts Strengths: Beginner-friendly, extensive documentation, Python-based workflow Features: Large user community, abundant pre-built examples, good for basic prototyping
Ultra96-V2	<ul style="list-style-type: none"> Ideal for: IoT-focused courses, edge AI projects, portable applications Strengths: Compact form factor with wireless connectivity Features: Mobile and edge computing demonstrations, power-efficient design
Kria KV260	<ul style="list-style-type: none"> Ideal for: Computer vision curricula, robotics applications Strengths: Vision-optimized with pre-built applications, system-on-module design Features: Embedded vision systems, ready-to-use vision pipelines
ZCU104	<ul style="list-style-type: none"> Ideal for: Advanced undergraduate courses, complex model deployment Strengths: High-performance capabilities, professional development features Features: Supports complex models, industry-standard interfaces
Alveo U50	<ul style="list-style-type: none"> Ideal for: Graduate research, data center AI studies Strengths: High-bandwidth memory (HBM), cloud integration capabilities Features: Data center acceleration, high-performance computing applications

require complete model development and optimization for specific applications. Assessment strategies should balance theoretical understanding with practical implementation skills, including code reviews, project demonstrations, and performance analysis reports.

Integration with existing computer science and electrical engineering curricula is essential for sustainable program development. FPGA-AI topics naturally complement courses in computer architecture, embedded systems, machine learning, and parallel computing. Cross-disciplinary collaborations can enhance educational outcomes by addressing both hardware and software perspectives of AI system design.

IX. CHALLENGES AND SOLUTIONS IN FPGA-AI EDUCATION

Implementing effective FPGA-AI education presents several interconnected challenges spanning technical, pedagogical, and institutional dimensions. These challenges, while significant, can be addressed through thoughtful approaches that have emerged from educational practice and research. Understanding these obstacles and their potential solutions is essential for educators seeking to establish or enhance FPGA-AI programs.

A. Technical Implementation Challenges

The technical landscape of FPGA-AI education presents several practical hurdles that institutions must navigate. First,

hardware accessibility remains a persistent concern. Although FPGA development boards have become more affordable in recent years, equipping laboratories with sufficient quantities for hands-on student work represents a substantial financial investment. This challenge is particularly acute for institutions seeking to deploy high-end platforms like Alveo accelerator cards or Versal adaptive SoCs. A multi-faceted approach to hardware access has emerged as a viable solution strategy. Cloud-based FPGA platforms, such as AWS F1 instances or Nimbix, offer remote access to sophisticated hardware resources, allowing students to experiment with data-center-grade acceleration without local infrastructure costs. Complementing cloud resources with judicious hardware sharing arrangements can maximize utilization of limited physical boards through carefully coordinated laboratory schedules. Furthermore, beginning with entry-level platforms like the PYNQ-Z2 provides an accessible starting point before progressing to more advanced hardware, creating a cost-effective pathway for skill development.

A second technical challenge lies in the steep learning curve associated with FPGA-AI development. Students must simultaneously master artificial intelligence concepts, hardware design principles, and specialized toolchains—a formidable cognitive load. Educational approaches that scaffold learning complexity have proven effective in addressing this challenge. Beginning with high-level frameworks like PYNQ, which emphasizes Python programming and abstracted hardware access, allows students to achieve early success with AI applications. As proficiency develops, students can progressively engage with more detailed tools like Vitis AI and eventually hls4ml, which provide greater control but require deeper hardware understanding. This graduated approach, supported by comprehensive libraries of pre-built examples and visual design tools, helps students build confidence while developing sophisticated skills incrementally.

The complexity of FPGA toolchains represents a third significant technical challenge. Unlike traditional software development environments with relatively straightforward setup procedures, FPGA toolchains involve multiple interdependent components with intricate configuration requirements. These setup complexities can consume valuable instructional time and create frustrating barriers for novice learners. Educational institutions have developed several strategies to mitigate these challenges. Docker containers with pre-configured development environments eliminate installation difficulties and ensure consistent setups across different student machines. Simplified workflows using curated scripts and templates abstract away some of the toolchain complexity while still exposing key concepts. Additionally, meticulously detailed step-by-step tutorials that guide students through each phase of development—from environment setup through synthesis and deployment—help demystify the process and build procedural knowledge systematically.

B. Pedagogical Design Challenges

Beyond technical implementation, FPGA-AI education faces distinctive pedagogical challenges related to curriculum design, assessment, and content currency. The interdisciplinary nature of the subject matter creates particular complexities. FPGA-AI sits at the convergence of computer science, electrical engineering, and applied mathematics, requiring students to integrate knowledge from domains that are typically taught separately. Effective curriculum design addresses this interdisciplinary requirement through modular content organization. Self-contained modules covering specific topics—such as quantization algorithms, parallel hardware design, or power optimization—can be flexibly integrated into different course structures across departments. Clear prerequisite mapping helps students navigate the knowledge landscape, while cross-disciplinary team projects leverage diverse student backgrounds, encouraging knowledge sharing and collaborative problem-solving.

Assessment design presents another significant pedagogical challenge. Traditional examination formats struggle to capture the practical, project-based skills essential for FPGA-AI proficiency. Alternative assessment approaches have emerged that better align with learning objectives. Portfolio-based assessment, where students compile collections of projects demonstrating progressive skill development, provides a comprehensive view of learning achievements. Performance-based evaluation metrics—such as achieved inference latency, power efficiency, or model accuracy—offer concrete measures of technical proficiency. Structured code reviews, conducted by both peers and instructors, develop professional practice while providing detailed feedback on implementation quality. These assessment approaches collectively emphasize applied skills while maintaining academic rigor.

The rapid pace of technological evolution in both AI algorithms and FPGA architectures presents ongoing challenges for curriculum development. Tools, frameworks, and best practices evolve continuously, potentially rendering specific content obsolete within relatively short timeframes. Educational programs address this challenge through several complementary strategies. Emphasizing core principles—such as parallel computation, memory hierarchy optimization, or quantization theory—ensures that students develop durable knowledge that remains relevant despite technological changes. Modular curriculum design allows specific tools or frameworks to be updated without overhauling entire courses. Industry partnerships provide valuable insights into emerging trends and technologies, helping educators maintain curriculum relevance while connecting students with real-world applications and potential career pathways.

C. Institutional and Resource Challenges

Institutional factors significantly influence the successful implementation of FPGA-AI education programs. Faculty expertise represents a foundational resource that is often limited, as relatively few academics possess deep experience in both artificial intelligence and FPGA hardware design. This

expertise gap can be addressed through targeted professional development initiatives, including workshops, summer institutes, and collaborative research projects that help existing faculty develop necessary skills. Team teaching arrangements, pairing computer science faculty with electrical engineering colleagues, leverage complementary expertise while modeling interdisciplinary collaboration. Inviting industry experts as guest lecturers or adjunct faculty brings practical perspectives into the classroom while supplementing institutional expertise.

Laboratory infrastructure requirements present substantial institutional challenges. Establishing and maintaining FPGA development laboratories involves significant capital investment and ongoing technical support. Phased implementation approaches, beginning with modest setups that can be expanded based on demonstrated student interest and learning outcomes, allow institutions to manage costs while building program momentum. Cloud resources can supplement physical laboratories, providing access to specialized hardware without corresponding capital expenditures. Strategic pursuit of educational grants and industry partnerships can provide funding specifically targeted at laboratory development, helping institutions overcome initial resource barriers.

Curriculum integration requires careful institutional planning to find appropriate placements for FPGA-AI content within existing academic structures. Specialized elective courses allow interested students to pursue in-depth study while providing manageable enrollment numbers for initial program offerings. Integrating FPGA-AI modules into established courses—such as machine learning, computer architecture, or embedded systems—introduces the subject to broader student populations. Capstone project options enable advanced students to apply FPGA-AI concepts to substantial independent work, often in collaboration with industry partners or research faculty. This multi-tiered approach to curriculum integration creates multiple pathways for student engagement while allowing programs to scale appropriately based on resources and demand.

Collectively, these challenges and solutions paint a comprehensive picture of the FPGA-AI educational landscape. While implementing effective programs requires thoughtful attention to technical, pedagogical, and institutional factors, the strategies described here provide actionable approaches for educators and administrators. By addressing these challenges systematically, academic institutions can develop robust FPGA-AI education programs that prepare students for careers at the intersection of artificial intelligence and hardware acceleration, contributing to a workforce capable of advancing this critical technological domain.

X. CONCLUSION AND FUTURE DIRECTIONS

This comprehensive survey has examined the current landscape of educational platforms for learning AI on FPGA hardware. The unique advantages of FPGAs—including reconfigurability, parallelism, low latency, and power efficiency—make them particularly valuable for teaching hardware-accelerated machine learning concepts. Frameworks such as PYNQ, Vitis

AI, hls4ml, and FINN have significantly lowered barriers to entry, enabling students to implement sophisticated AI applications without extensive hardware design expertise.

Future developments in FPGA-AI education will likely focus on several key areas: increased integration with cloud-based development environments, expanded support for emerging neural network architectures, enhanced tools for performance analysis and debugging, and greater emphasis on ethical considerations in hardware-accelerated AI. Additionally, the growing adoption of RISC-V processor cores in FPGA fabrics presents opportunities for teaching open-source hardware concepts alongside AI implementation. The RISC-V ecosystem, with its open standard and customizable architecture, is particularly well-suited for educational exploration of AI hardware acceleration.

As AI continues to permeate diverse application domains, education in hardware-accelerated implementations will become increasingly important. The platforms and methodologies surveyed in this paper provide a foundation for developing effective curricula that prepare students for careers at the intersection of artificial intelligence and hardware design. By leveraging these resources, educational institutions can contribute to developing a workforce capable of optimizing AI systems across the computing continuum from edge to cloud.

ACKNOWLEDGMENTS

The authors acknowledge support from Hamm-Lippstadt University of Applied Sciences and thank the FPGA and AI research communities for their contributions to open-source educational resources.

XI. DECLARATION OF ORIGINALITY

I am Mohamed Abdo, herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker

03/04/2025 Lippstadt

Mohamed Abdo

REFERENCES

- [1] Xilinx/AMD, “Pynq: Python productivity for zynq,” 2024. [Online]. Available: <http://www.pynq.io>
- [2] ——, “Alveo accelerator cards,” 2025. [Online]. Available: <https://www.xilinx.com/products/boards-and-kits/alveo.html>