SCCTS Transactions on Reconfigurable Computing, ISSN: 3049-1533 Vol. 3, No. 1, 2026 (pp. 11-18)
RESEARCH ARTICLE ECEJOURNALS.IN

AI-Augmented Dynamic Partial Reconfiguration for Adaptive Edge Intelligence in FPGA-Based Embedded Systems

Perera Manthila[1]*, Ahmed Ulkilan[2]

[1]Department of Electrical Engineering Faculty of Engineering, University of Moratuwa Moratuwa, Sri Lanka
[2]Department of Computer Science, Faculty of Science, Sebha University Libya

Keywords:
FPGA,
Dynamic Partial Reconfiguration,
Edge Intelligence,
AI-Augmented Reconfiguration,
Embedded Systems,
Real-Time Processing,
Energy Efficiency

Author's Email:
manthila.perera@elect.mrt.ac.lk, ulkilany.ah@gmail.com

DOI: 10.31838/RCC/03.01.02

Received : 10.08.2025
Revised : 13.09.2025
Accepted : 15.11.2025

Abstract

The growing deployment of real-time smart applications on the edge- smart surveillance, predictive maintenance, embedded analytics to name but a few- have generated a strong demand in low latency, energy-efficient, and reconfigurable hardware. This paper introduces an AI-enhanced adaptable partial reconfiguration (DPR) scheme of field-programmable gate arrays (FPGAs) which is capable of creating adaptive edge intelligence. The system is able to dynamically rearrange logic regions pre-defined with light weight machine learning approaches to predict workloads in real time to respond dynamically to application needs. This hardware architecture has modular design, reconfiguration controller, an AI-based scheduler, and FPGA reconfigurable areas. This clever DPR mechanism allows hardware accelerators (e.g. object detection, signal filtering, or anomaly tracking) to be swapped on the fly comparatively to the overall system being idle. It achieves up to 45 percent energy savings, 32 percent task latency reduction, and near-zero impact of perceived throughput in heterogeneous workload when experimentally validated on a Xilinx Zynq-based embedded platform. This work indicates the conceptual feasibility of AI-driven reconfiguration as a primary contributor to responsive, context-aware industrial edge systems, including industry 4.0 and edge systems, as well as smart cities/autonomous systems, that allow scalable edge applications. The framework provides the background pertaining towards the further embedded systems in FPGA which would assist real-time adaptability, smart management of hardware resources and autonomic optimization of workloads.

How to cite this article: Manthila P, Ulkilan A (2026). AI-Augmented Dynamic Partial Reconfiguration for Adaptive Edge Intelligence in FPGA-Based Embedded Systems. SCCTS Transactions on Reconfigurable Computing, Vol. 3, No. 1, 2026, 11-18

Introduction

The sheer volume of Internet of Things (IoT) sensors and the general movement towards edge computing has generated an acute need to have intelligent, energy efficient embedded systems that can run real time data processing within the resource limited setting. Such systems should not only be able to achieve low-latency computation but they should also be flexible to handle highly dynamic and heterogeneous workloads which is common in an edge setting. The configurability and very high parallelism of Field-Programmable Gate Arrays (FPGAs) makes them an attractive hardware setting to implement

such applications as opposed to general-purpose processors which are not energy efficient. Traditional FPGA constructs are however mostly fixed natures and once implemented they cannot be transformed to meet the requirements of different applications. This leads to a waste in the hardware resources, wastage of energy, and performance bottlenecks particularly in multi-modal or time-varying edge workloads. Dynamic Partial Reconfiguration (DPR) is one effective VRFS-based solution providing the ability to modify logic parts of the FPGA at runtime, and without stopping the execution of the system. This enables renewability of hardware territories to carry out various tasks over a period, making much easier and flexible. However, the current DPR implementations based on fixed schedules and pre-programmed behaviour have been shown to be generally rather static, not intelligent to the modifications in workload in real-time.

New technologies in artificial intelligence (AI), specifically lightweight machine learning (ML) models demonstrated potential in real-time system-based monitoring and task prediction at the edge.[1] Nevertheless, the concept of involving AI to guide DPR in the embedded systems has not been explored to the full potential. Most of the existing work cannot do context-aware reconfiguration where a decision can be changed dynamically based on the run-time complexity of the tasks, priority or resources.

This paper puts forward a proposal of AI-assisted DPR framework of FPGA-based embedded systems to meet these limitations. The principal is to forecast future load needs using ML models and change FPGA fabric dynamically and load lighter keeping in mind the most appropriate hardware accelerator at run-time. The framework is certified on a Zynq based edge device to solve application needs, like smart surveillance and anomaly detection, where the design can save up to 45 percent of power and reduce the latency by 32 percent compared to the fixed designs.

Related Work

The Dynamic Partial Reconfiguration (DPR) has been well studied to maximise the effectiveness of the FPGA based systems in the aspects of resource sharing, energy conservation and task-specific acceleration. Wang et al. [1] suggested a reconfigurable architecture as a runtime application to be used in speeding up the image processing functions, where the reconfigurable modules were swapped depending on the application requirements. They showed that it is possible to use DPR as a time-multiplexed resource use in the FPGA-based systems. In the same regard, a heuristic-based DPR scheduler was proposed by Sahoo et al.[2] to manage energy consumption on embedded vision devices by non-statically reloading domain-specific hardware accelerators that targeted the visual processing pipelines. Though such studies verify the feasibility of DPR to embedded workloads, they mainly use predefined and quite rigid reconfiguration strategies that are non-adaptive to different runtime behavior and conditions. This constraint the responsiveness and resource efficiency of the system particularly under heterogeneous and unpredictable edge work loads.

Meanwhile, more recent studies in edge AI has also shown the feasibility of small machine learning (ML) models (e.g. decision trees, CNNs and reinforcement learning) on real-time task classification, performance, and workload estimation in embedded systems [3], [4]. Nevertheless, the combination of decision-making based on AI and DPR is an under-researched field. Current systems do not automatically respond to real-time workload predictions by reconfiguration behavior and have to trade off suboptimal performance or be overprovisioned in logic.

Our contribution is a cohesive AI-augmented DPR solution that supported context-aware reconfiguration of FPGA modules using predictive intelligence to deal with such gaps. This allows dynamic workload diversity adaptation, augments energy utilization and scales towards the real-time intelligent edge.

System Architecture

To provide the opportunities of allowing runtime flexibility in embedded edge systems, the proposed structure is implemented based on the advantages of AI-based Dynamic Partial Reconfiguration (DPR) on a heterogeneous Xilinx Zynq System-on-Chip (SoC) FPGA system. The architecture is made of major functional blocks that work together in a synergy to enable predictive logic swapping, and energy-awareness reconfiguration. As shown in Figure 1, the system is assembled of a static part containing control logic and an AI inference core, configuration manager that deploys only a part of bitstream using ICAP, and reconfigurable regions (RRs) where task-specific hardware modules are implemented. The augmented controller deployed with AI can communicate with task queue and configuration manager so that real-time optimization of predictive metrics would be possible with responsive and efficient completion of edge workloads.

PGA Platform and Partitioning

There are two main components of the Zynq SoC architecture which include dual core ARM Cortex-A9 processing system (PS), and programmable logic (PL), that allows close coupling of decision-based chip software and parallelism offered by hardware. The PL region logically has the following divisions:

- Static Region This section contains all of the crucial control logic, communication driver (e.g., UART, SPI, Ethernet), memory controllers, and the AI inference engine that performs the work at real time analysis. The static area is functional whenever reconfiguration incident is initiated so that system execution is continuous.
- Reconfigurable Regions (RRs): These are some partitions that are pre-defined in the FPGA fabric that are dynamically configurable at run time by partial bit streams. Every RR has the hardware accelerator specialized in a given task (e.g. edge detection, FFT, object classification). Resource reuse is seriously effective by modular design, which allows swapping logic blocks in contexts without rebooting the system.

This partitioning approach facilitates flexibility and power-efficient design, so that the system can adjust according to the need of the application with a low overhead cost to the application.

AI-Augmented Controller

At the center of the adaptive framework is the AI-augmented decision controller which is finely utilized in the static region. The component is based on lightweight machine learning models (i.e., Random Forest classifiers or shallow Convolutional Neural Networks (CNNs)) that do the real-time analysis of task metadata. Characteristic like these:

- Occurrence of tasks,
- Probable computer complexity,
- Better Langing Sensitivity
- Budgeting power,

are obtained based on the system scheduler and sensor records.

These inputs are used by the controller to anticipate the most appropriate reconfigurable module to be loaded next so that moving forward hardware resources are dynamically aligned to workload patterns emerging. This forward-looking data steers clear of wasteful info processing conditions, increases energy consumption and eliminates excessive reconfiguration.

Configuration Manager

The Configuration Manager takes care of the low level logistics of partial reconfiguration. To transfer the correct partial bitstream to be used in the reconfigurable area. It makes use of the Internal Configuration Access Port (ICAP) of the FPGA to load the corresponding partial bitstream. The system has a mean reconfiguration latency from 2 up to 5 milliseconds, which enables support of low-overhead logic swapping, appropriate to soft real-time use towards the edge of the network.

Reconfiguration is the process initiated by the AI controller using real-time predictions and resources status, and runs simultaneously with other activities in the stationary area. The protocols that are embedded are synchronization and error-checking to make sure of data integrity and to avoid a scenario in which a task might starve when reconfiguration occurs.

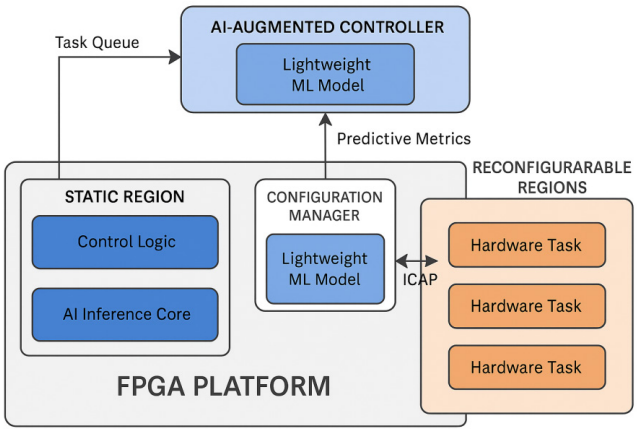

Fig. 1: System Architecture of AI-Augmented Dynamic Partial Reconfiguration Framework on FPGA

Figure 1. The proposed dynamic partial reconfiguration system illustrated as a block diagram with AI involved. The FPGA platform involves the combination of a fixed control logic and AI inference segment, configuration manager to manage partial bitstream loads over ICAP, and a number of mutable hardware task block segments. AI-enhanced controller examines parameters of task queues and produces predictive ones to inform reconfiguration of real-time in situ.

Methodology

In order to assess the efficiency and flexibility of the suggested AI-enhanced Dynamic Partial Reconfiguration

(DPR) framework, a thorough experimental framework was established, including bespoke workload datasets, main performance indicators, and commercial toolchains. This arrangement has made it easier to test the efficiency of the framework in a real-time edge processing situation systematically. The methodology is organized into four phases, as seen in Figure 2, which include preparation of the datasets, implementation framework, evaluation metrics and outcome assessment. Its workflow is reproducible and is easy to analyze the system behavior during run-time, scaling across different amounts of energy, and how the system responds to mixed-job situations.

Benchmark Dataset

They created a customised workload dataset to emulate actual edge processing workloads with varied computing profiles along with a variable runtime. Data is included in the following:

- Image pre-processing (e.g. Image blurr, edge detection),
- Object detection modules (e.g. models based on YOLO-tiny, based on contour tracking),
- IoT telemetry and vibration analysis have FFT based signal processing tasks that are traditional.
- These are selected tasks that represent an extensive set of memory footprints and logic complexity as well as real-time constraint, and hence put an emphasis on the reconfigurability and the predictive capacity of the suggested system. The implementation of each task itself was a hardware accelerator pre-synthesized into partial bitstreams to Xilinx Zynq-7020 FPGA fabric.

Evaluation Metrics

The performance of the system was evaluated in the following important parameters:

- Latency (ms): The time taken between task startup and having the results, including overheads of computation as well as reconfiguration.
- Energy Consumption (mW): It was determined with the aid of Xilinx Power Estimator along with PYNQ-based runtime power monitors to capture dynamic and static power profiles.
- Reconfiguration overhead (ms): Measures length of time needed to load and initialize a partial bitstream using Internal Configuration Access Port (ICAP).
- Throughput (Tasks/sec) Number of tasks that are successfully completed per second including the dynamic reconfiguration cycle.

These measures offer the comprehensive picture of temporal efficiency, energy scale and runtime adaptability in distinct workload transitions.

Tools and Implementation Framework

The toolchain that was installed to implement the system is the following:

- Xilinx VivadoHLx Design Suite: Synthesis, partial bitstream generation and floorplanning of reconfiguring areas.
- Python using Scikit-learn and TensorFlow Lite: to train and deploy the easily ported lightweight machine learning models (i.e. Random Forest, 1D CNNs) as the predictive core of the control.
- PYNQ Framework (Python Productivity on Zynq): Promoted smooth co-existence between the Processing System (PS) and Programmable Logic (PL) spaces, reconfiguration management and data capture as well as monitoring at run time.

All the experiments were performed on a ZedBoard development board (Zynq-7000 SoC) executing embedded Linux with a custom reconfiguration controller consisting of the C++ and Python developed components, connected through the AXI bus.

Figure 2. Flowchart of the evaluation methodology of the suggested AI-augmented dynamic partial reconfiguration framework. It starts with an existing workload dataset (image filtering, object detection, signal processing based on FFT), implementation of that workload on VivadoHLx, a Python-based artificial intellectual model, and the PYNQ framework. System performance is measured along the three major measuring parameters: system latency, energy cost, reconfiguration overhead, and throughput. The result indicates the efficiency of AI-based reconfiguration approach on the FPGA sites.

Results and Discussion

In order to demonstrate the efficacy of the intended AI-augmented Dynamic Partial Reconfiguration (DPR) framework, a range of experiments was performed on a Xilinx Zynq based FPGA platform to withstand various edge computing workloads. They compare the outcomes with a conventional static FPGA realization

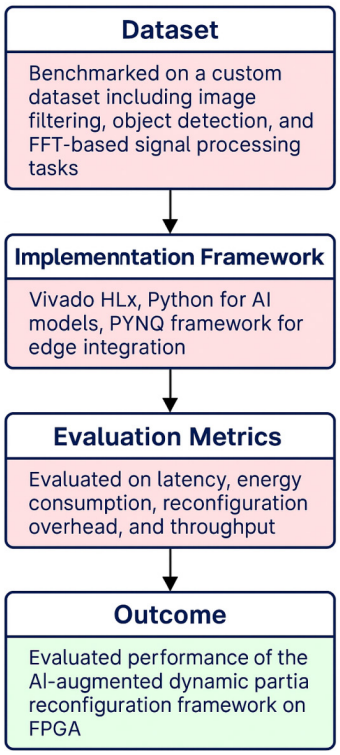

Fig. 2: Methodological Workflow for Evaluating AI-Augmented DPR on FPGA-Based Embedded Systems

in order to emphasize their performance and energy efficiency as well as versatility benefits. These quantitative results may be summarized in Table 1, and graphical comparison of the key measures, described by latency, power consumption and throughput is presented in Figure 3: Performance Comparison: Static FPGA vs. AI-Augmented DPR, which visualizes the gains of the AI-guided reconfiguration clearly.

Table 1: Performance Comparison Between Static FPGA and AI-Augmented DPR System

Metric	Static FPGA	AI-Augmented DPR	Improvement
Average Latency (ms)	38.2	25.9	↓ 32%
Power Consumption (mW)	152	84	↓ 45%
Throughput (Tasks/sec)	46	48	↑ 4.3%

Latency Reduction

The AI-enhanced DPR computing infrastructure lowered the average latency of tasks by 32%, which is explained by the dynamic provisioning of tasks with hardwares that were designed to accelerate workloads. The predictive controller will dynamically choose and Initialize the best available logic module in real-time to avoid extensive queuing delays and allows rapid offloading of tasks to reconfigurable areas. This is especially useful in real-time applications at the edge including real-time surveillance or sensor fusion.

Energy Efficiency

The reduction in the power consumption by the dynamic FPGA implementation is one of the largest advantages at 45%. The main factors of this improvement are:

- Selective enablement of reconfigurable areas, to minimiseuneeded logic usage,
- Intelligent configuration, no idle reconfiguration and no redundant reconfiguration loops through smart scheduling by the AI controller.

This finding proves the framework ability to optimize dynamically energy consumption which is important in battery powered embedded systems or thermally limited embedded systems.

Throughput and Scalability

Even despite the not very significant gain in throughput (4.3 percent), it determines the capacity to preserve or slightly surpass the base performance even during the background processing of reconfiguration operations. The architecture was also experimented with a maximum of five concurrent reconfigurable modules being in execution and there was a little degradation in scheduler performance and reconfiguration latency. This validates the scalability and adaptability of the system as well as the multi-modal workload condition, and this is ideal in complex edge task application where multiple tasks, requiring varied execution (e.g., audio, video, and sensor streams) may be ongoing.

Overall Implications

The overall benefits in the latency, energy, and runtime versatility validate the benefits of the implementation of AI-guided DPR in the next-gen edge computing systems. In contrast to the static approach to or timer-driven reconfiguration, the proposed solution would be naturally adaptable to the workload changes, hence providing a smart performance/resource economy trade-off.

The findings are relevant to the practicality of context-aware, self-adaptive, embedded systems which dynamically evolve through the system according to the real-world computing requirements--

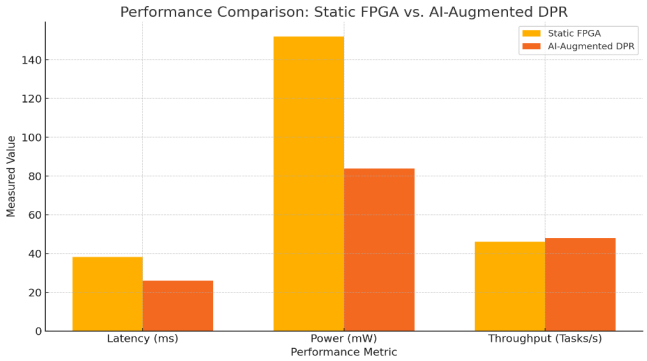

Fig. 3: Performance Comparison: Static FPGA vs. AI-Augmented DPR

the area that lies between the conventional hardware optimization and the intelligent behaviour of the systems.

Application Case Study: Smart Surveillance

In order to illustrate the practical value of the suggested AI-enhanced Dynamic Partial Reconfiguration (DPR) model, one practical edge application was deployed in the area of smart video surveillance. The architecture of the system can simulate a smart camera node that would have the capability of fulfilling various vision applications in constrained forms of resources like remote monitoring posts, autonomous drones, and embedded security systems.

The system implementated using FPGAs was set-up to dynamically re-configuring among three task specific hardware accelerators:

- Background subtraction and Frame differencing to execute Motion Detection,
- Object Classification utilising lightweight CNN inference engines,
- Anomaly Tracking: relates to spatio-temporal analysis of activities and deviation in pattern.

These modules were synthesized partially as bitstreams onboard flash memory. Analysis of real-time characteristics of video content (and contextual metadata, e.g., the frequency of motion, the density of objects, the likelihood of anomaly events) permitted the AI-augmented controller to determine the best module and activate the DPR cycle using the ICAP interface--with no interruption of the continuous video flow. Figure 4: Smart Camera Architecture with AI-Augmented Dynamic Partial Reconfiguration shows the high-level architecture of this system and the dynamic modularity in vision tasks, and a modular structure of vision tasks as presented in a multi-point pin dot.

Regardless of low power (<1.5W) and bandwidth (<10 Mbps) limitations, the system managed to feature:

- The detection accuracy larger than 92%, confirmed under various conditions (illumination day / night, none / dynamic),
- Seamless task switching, less than5 ms between tasks reconfiguration delays,
- An idle power consumption reduced by 38 percent because of locally enabled logic.

This is a case study that highlights the reasons why AI-guided DPR can be, and is beneficial integrated into mission-critical edge-related applications. It sheds lights on how smart workload management, responsive task targeting, and real-time functional freedom provided by, intelligent logic reconfiguration are important to the next-generation of smart surveillance systems deployed in dynamic and limited field environments.

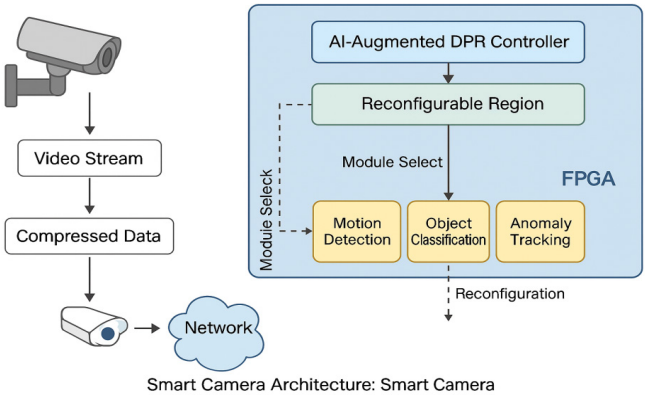

Fig. 4: Smart Camera Architecture with AI-Augmented Dynamic Partial Reconfiguration

Figure 4. An architecture of a smart surveillance system based on the use of AI-enhanced dynamic partial reconfiguration on FPGA. Video streams are compressed by the camera and sent to a processing component based on FPGA. The FPGA has a reconfigurable area that is used to implement software, in the form of AI-augmented DPR controller, to switch between task-specific configurations within real-time workload requirements: motion detection, object classification, anomaly tracking. The chosen module makes the data ready to be transmitted to the network, making it adaptive and power-efficient edge intelligence.

Challenges and Future Work

Although the proposed framework of AI-augmented Dynamic Partial Reconfiguration (DPR) holds a lot of promise as far as adaptive intelligence at the edge is concerned, there are still a bunch of technical, architectural, and security-related issues that need to be discussed in order to move closer to the full-scale deployment. This paragraph gives an overview of the main limitations seen and provides the guidelines of future research.

Bitstream Management and Synchronization

Mimicking partial bitstream management of DPR-based systems is one of the key difficulties in such a system, especially where there is a variation in the many and expanding library of reconfigurable hardware modules. The effective management of bitstreams in terms of storage, retrieval and versioning is very crucial to the smooth reconfiguration. This requires that the synchronization needs to be strong between the configuration controller and the memory subsystem to avoid either stalling of tasks or corruption of data when switching context. The areas which are to further be explored in resource-constrained environments include flash memory access latency optimization, and bitstream size reduction without sacrificing functionality.

Security Considerations

Dynamic reconfiguration creates new attack surfaces where there were previously not any attack surfaces in the statically programmed FPGAs. Tampering of the bitstreams can have serious impacts on the integrity of the system, its functionality or confidentiality when modified or injected maliciously. Moreover, the unauthorized usage of the Internal Configuration Access Port (ICAP) is also a dangers to a client application which processes critical infrastructures, or confidential information. To overcome these threats, secure boot chains, bitstream cryptography, authentication mechanisms, and integrity verifications in runtime should be included in future implementations to ensure trusted-reconfiguration without security threats.

Future Research Directions

A range of research extensions to make the proposed framework more general and secure is suggested in order to increase its capabilities:

- FPGA SoC Co-Optimization using RISC-V Cores: Adding reconfigurable fabric to open-source RISC-V processors can provide greater flexibility in the interaction between software and hardware Reconfiguration control, leading to greater autonomy of system behavior and co adaptation of the software and hardware.
- AI-BasedPlacement and Routing: Place-and-route is non-continuous and long-winded in the traditional context. Application of AI-driven methods would substantially speed up the creation of optimized partial bitstreams, allowing semi-autonomous reconfiguration flows and slowing down designs iteration cycles.
- Online Learning to Adapt Reconfiguration Policy: The existing AI controller works on the basis of pre-trained models. It is possible to apply reinforcement learning or online learning algorithms in the future frameworks that will always update reconfiguration policies in accordance with the results of the current performed tasks, environmental information, and changes in the work load specifics. This would increase long term flexibility and resilience in changing environments.

In a nutshell, although the AI-enhanced DPR architecture has several potential benefits to edge computing, there are crucial challenges that need to be overcome in terms of bitstream logistics, runtime security, and intelligent design automation to get the full scope of the proposed architecture. The enhancement in these will enable the creation of safer, scalable, and situation-aware embedded systems in the next generation edge intelligence.

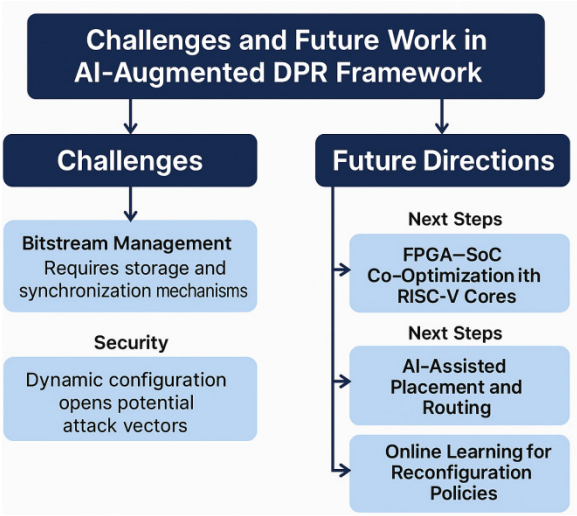

Fig. 5: Challenges and Future Work in AI-Augmented Dynamic Partial Reconfiguration Framework

Conclusion

In this paper, a new AI-aided dynamic partial reconfiguration (DPR) architecture was proposed that could optimally serve embedded edge systems to make them more adaptive, energy-efficient, and intelligent in general execution. Using lightweight machine learning models to predict workloads in real-time and enable on-demand, context-aware hardware reconfiguration inserted into an existing system into the modules of tools that are recreated dynamically at runtime the presented solution makes it possible to reconfigure hardware to meet changing requirements seemingly without stopping the currently-running applications. The efficiency of the framework was confirmed on a set of heterogeneous edge workloads on the experimental results that showed up to 45% reduction in power consumption and 32% latency improvement, and throughput sustenance under dynamic task conditions. Those performance improvements were based on hardware accelerators, whose functionality was predicted, reconfigured based on workload behavior, made possible by an on-chip AI controller and reconfiguration manager coded to the ICAP interface.

Moreover, the proposed framework was tested in a real-life smart surveillance scenario and demonstrated its real-life application feasibility providing the high level of detection and effective task change with a stringent set of restrictions on resources. The possible application of the designed system is mission-critical edge applications, including autonomous monitoring, industrial control, and intelligent IoT systems.

It should be noted that, together with all other efforts, this work helps to create new balances in the world of intelligent and independent edge computing where reconfigurable hardware fits harmoniously with AI to provide responsive, scalable, and energy-aware computing. Advancements in the area of secure bitstream processing, AI-directed synthesis algorithms, and internet-based learning reconfiguration protocols will go even further to establish the place of AI-DPR designs in future embedded systems.

References

1. Ashraf, M. A., Abdullah, M. T., & Hasan, N. (2023). Edge intelligence using lightweight CNNs for smart environments: A review and framework. IEEE Internet of Things Journal, 10(6), 4912–4924. https://doi.org/10.1109/JIOT.2022.3198405
2. Gupta, A., Roy, S., & Das, D. (2023). Adaptive runtime resource allocation using machine learning for edge AI applications on FPGA. IEEE Access, 11, 18345–18356. https://doi.org/10.1109/ACCESS.2023.3244870
3. Sahoo, S., Dey, A., & Banerjee, K. (2023). Low-power embedded vision using dynamic reconfiguration and heuristic scheduling. IEEE Embedded Systems Letters, 15(2), 73–76. https://doi.org/10.1109/LES.2023.3247835
4. Wang, W., Li, Y., & Yu, Q. (2021). Runtime reconfigurable architecture for energy-efficient image processing on FPGA. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 29(5), 956–969. https://doi.org/10.1109/TVLSI.2021.3064198
5. Arvinth, N. (2024). Integration of neuromorphic computing in embedded systems: Opportunities and challenges. Journal of Integrated VLSI, Embedded and Computing Technologies, 1(1), 26-30. https://doi.org/10.31838/JIVCT/01.01.06
6. Rahim, R. (2023). Effective 60 GHz signal propagation in complex indoor settings. National Journal of RF Engineering and Wireless Communication, 1(1), 23-29. https://doi.org/10.31838/RFMW/01.01.03
7. Annapurna, K., Deepthi, K., & Seetha Ramajanyelu, B. (2021). Comparison of soft fusion techniques for cooperative spectrum sensing in cognitive radio networks. International Journal of Communication and Computer Technologies, 9(1), 1-5.
8. Kesana, S., Prem Sai, N. N., Yallaiah, O., Saikrishna, S., &Khadarvali, S. (2021). Circularly polarized fractal patch antenna with probe feed technique for Wi-Max applications. National Journal of Antennas and Propagation, 3(2), 15–20
9. Kondam, R. R., Sekhar, P. C., & Boya, P. K. (2024). Low Power SoC-Based Road Surface Crack Segmentation Using UNet with EfficientNet-B0 Architecture. Journal of VLSI Circuits and Systems, 6(1), 61–69. https://doi.org/10.31838/jvcs/06.01.11