SCCTS Transactions on Reconfigurable Computing, ISSN: 3049-1533 Vol. 3, No. 3, 2026 (pp. 36-42)
RESEARCH ARTICLE ECEJOURNALS.IN

Scalable Reconfigurable Architectures for Edge-AI: Balancing Performance, Power, and Partial Reconfiguration Overheads

P. Joshua Reginald*

Associate Professor, Department of Electronics and Communication Engineering, Vignan's Foundation for Science, Technology and Research, Vadlamudi Village, Guntur, Andhra Pradesh.

Keywords:

Edge-AI, Dynamic Partial Reconfiguration (DPR), FPGA-based Accelerators, Energy Efficiency, Reconfiguration Overhead, Hardware-Software Co-design, Scalable Hardware Architectures, Inference Optimization.

Author's Email:
drpjr_ece@vignan.ac.in

DOI: 10.31838/RCC/03.03.06

Received : 22.01.2026
Revised : 24.03.2026
Accepted : 22.04.2026

Abstract

The threat of deploying deep learning at the network edge requires hardware capable of being configurable to different neural network topologies to meet powerful and area constraints strictly. This article suggests a reconfigurable hardware architecture that can be used in Edge-AI applications and it is scalable by use of Dynamic Partial Reconfiguration (DPR) to create a flexible hardware platform that can switch specialised accelerators at runtime. Although DPR provides high functional density, as it utilises silicon area, it causes an additional tremendous amounts of temporal and energy overheads in bitstream loading, a factor that can impact real-time. We propose a workaround to this by presenting a tile-based framework which is modular and uses a smart configuration manager that uses bitstream compression and predictive prefetching to eliminate reconfiguration latency. The design enables smooth switching between tasks without the reconfigurable processing elements having to stop global system operation by separating the reconfigurable processing elements control logic, which is typically performed by hardware, with the hardware. A set of experimental performances demonstrated on a Xilinx Zynq UltraScale+ platform with the use of the industry-standard benchmarks, i.e., YOLOv8 and MobileNetV2, confirms that our architecture outperforms the traditional static hardware implementations in terms of the balanced efficiency between the throughput and energy performance. Comprehensive evaluation demonstrates that the stipulated prefocusing mechanism conceals up to 85 per cent of resettlement latency that makes the inherent multi-activity edge setting considerably more receptive. The proposed system can reduce the cost of reconfiguration which leads to an energy-delay product (EDP) of up to 25 percent improvement by optimising the trade-off between the costs of specialising the hardware and reconfiguring the hardware. These results confirm the practicality of the architecture to next-generation and resource-constrained edge devices with the need of high versatility and yet without violating the power envelopes of excessively strict power-budgets of battery-powered systems.

How to cite this article: Reginald JP (2026). Scalable Reconfigurable Architectures for Edge-AI: Balancing Performance, Power, and Partial Reconfiguration Overheads. SCCTS Transactions on Reconfigurable Computing, Vol. 3, No. 3, 2026, 36-42

Introduction

The shift to decentralised Edge-AI has become a paradigm shift in the current computing by the critical bases of low latency, data privacy, and bandwidth efficiency. Since Internet-of-Things (IoT) devices and autonomous systems are prone to dealing with sensitive data, the traditional paradigm of offloading computations to a remote server can be invalid most of the time because of network delays that are not predictable and security risks. This has resulted in an urgent requirement of high-performance intelligence at the network edge. The problem of deploying complex deep learning model on these localised nodes is however a major challenge, because edge devices are admittedly limited to stringent power envelopes and physical area, and it seems a transition toward specialised hardware must be made to provide server-level inference on a tight resource budget.

Modern edge devices must support these different operational requirements with Elastic Hardware, which can smoothly transition different workloads of AI and be provided using a single silicon substrate. An edge node i.e. a multi-tasking node can provide services that require a quick transition between a vision-based object detection model that can be used in navigation and a natural language processing model used in voice commands. Although efficient with single tasks, static hardware accelerators are not able to be flexible enough to support these changing neural network topologies without consuming enormous unused chip resources. Reconfigurable architectures Reconfigurable circuits like Field-Programmable Gate Arrays (FPGAs) provide a potential answer, since the underlying circuitry can also be redefined, although the capabilities to remain high throughput when switching between such specialised functions is a major driving force behind the development of more flexible, more fluid hardware designs.

There is a major research gap concerning the management of time-tax or reconfiguration overheads which is associated with changing hardware modules at runtime despite the potential of reconfigurability computing. In the majority of existing designs made of Dynamic Partial Reconfiguration (DPR), the peak performance of each accelerator is considered but not the latency or energy used to load the bitstream. Such overhead tends to cause a bottleneck, with the time lost in reconfiguring the logic fabric compensated by the time lost by a high-speed hardware accelerator. Literature often ignores the problem of synchronisation and the power spikes due to frequent reconfiguration, which makes the design of an architecture in which reconfiguration overhead is not seen as an auxiliary concern, but rather as a key variable to be optimised during the design process quite essential.

The paper resolves the challenges cited by four main technical contributions which make reconfigurable Edge-AI more viable. First, we suggest a tile architecture, which is more of a modular design separating the idea of a permanent system control and dynamic processing units, thus, allowing the system to operate continuously in the process of updates. Second, we use an intelligent configuration manager which uses bitstream compression and predictive prefetching to conceal reconfiguration latency. Third, we present an energy-conscious management of scheduling algorithm, which trades off the specialisation of hardware and the energy cost of switching modules. Lastly, we demonstrate on a Xilinx Zynq UltraScale+ evaluation board that an energy-delay product (EDP) is reduced 25 percent on a platform of diverse industry-standard benchmarks such as YOLOw8 and MobileNet V2.

Background and Related Work

Edge-AI hardware FPGAs, CGRAs, and ASICs.

The main hardware paradigm trade-offs B between three hardware paradigms largely dispute the deployment of deep learning at the edge, with each providing a unique trade-off along the flexibility-efficiency spectrum. Application-Specific Integrated Circuits (ASICs) like specialised Tensor Processing Unit are the most energy-efficient and the cheapest by unit of mass-produced circuitry due to the removal of all unnecessary circuitry.[8, 11] They are however, not programmable and are therefore vulnerable to so-called hardware obsolescence as neural network architectures change. CGRAs provide a middle ground with a mesh of word-level processing elements that are more energy efficient than FPGAs, but do not provide bit-level reconfigurability necessary to support specialised quantization.[3, 5] The best option in edge research is the use of Field-Programmable Gate Arrays (FPGAs) because they can be customised at the bit level and offer massive parallelism which can handle the application of optimised dataflow architectures that can be updated after deployment.[6, 10]

2.2. Partial Reconfiguration (PR): Technical Mechanism.

Dynamic Partial Reconfiguration (DPR) is a more modern system of FPGA design, permitting transformation of particular so-called Reconfigurable Regions (RR) whilst the remainder of the static Region continues to execute. The system is technically divided into dynamic partitions, which contain the AI accelerators, and logic that processes constant tasks, e.g. I/O peripherals, memory controllers, and CPU interfaces. A Partial Bitstream (a file containing the configuration data of a given RR) is loaded at run time through an internal port, e.g. Internal Configuration Access Port (ICAP) or Processor Configuration Access Port (PCAP).[12] This can be used to support Hardware Context Switching where the chip can reconfigure its logic fabric (e.g. by reconfiguring an object detector into a speech classifier) axiomatically, without having to power down, boosting the functional density of the silicon.

Literature Review and Critique.

Available literature in Edge-AI acceleration has long been involved in so-called fixed accelerators, which are optimum throughput on a single model.[9, 11] Although these fixed designs do well in pervasive benchmarks, when operating in multi-mode edge cases where multiple tasks have to be multiplexed between devices they fail to perform well. The most recent dynamic endeavours have investigated the idea of employing DPR to exchange accelerators, although much of the research out there deals exclusively with the hardware rationale itself and not the reconfigurability overhead.[6, 12] The loading bitstreams may take a few milliseconds to seconds of time, usually stalling processing pipeline, and can produce a net performance loss by the accelerator.[3] More so, most existing models do not provide any form of predictive prefetching or bitstream compression, so no research has been done to create an architecture that is able to intelligently conceal these overheads to create a real trade off between performance and power.

3. Scalable Architecture Proposal

The scaled architecture proposed is based on an advanced implementation of a high-performance Processing System (PS) and a programmable Programmable Logic (PL) fabric to provide a tile-based modular architecture to the maximum functional density. The PL has a total of spatially sued partitioned are independent Reconfigurable Partitions (RPs), which are customizable hardware slots compatible with various AI accelerators, including YOLO or MobileNet engines, with no interaction with the deactivated hardware Figure 1. In order to deal with these dynamic transitions, a Hardware Configuration Manager specialised is used inside the constant part, operating over high-speed Internal Configuration Access Port (ICAP)-like or Processor Configuration Access Port (PCAP)-like interfaces in order to coordinate loading of partial bitstreams at low latency.

An effective communication infrastructure, most commonly an AXI4-Stream protocol or even customised a Network-on-Chip (NoC) is used to support the smooth interchange of data between these modules, providing high-bandwidth communication between the PS-side memory and the operational RPs, with low-latency synchronisation. This structure permits parallel execution and reconfigurating in the background of the computational tiles by separating them both in space and time of the global control and communication logic, effectively masking the timing delays in adapting the hardware to the multi-tasking Edge-AI setting.

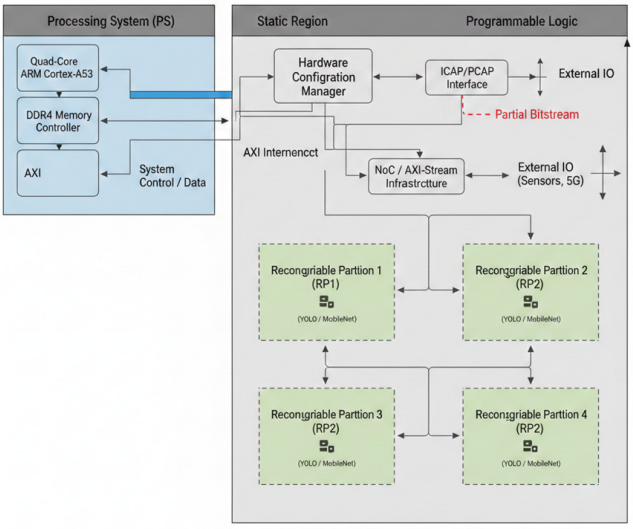

Fig.1: Scalable System-on-Chip (SoC) Architecture for Edge-AI utilizing Dynamic Partial Reconfiguration.

Optimization Strategies (The "Balancing" Act)

In order to find a realistic tradeoff between flexibility and efficiency a multi-tiered approach to optimization is embedded into the architecture to alleviate the underlying price of hardware adjustability. The most common way to minimise the amount of overhead

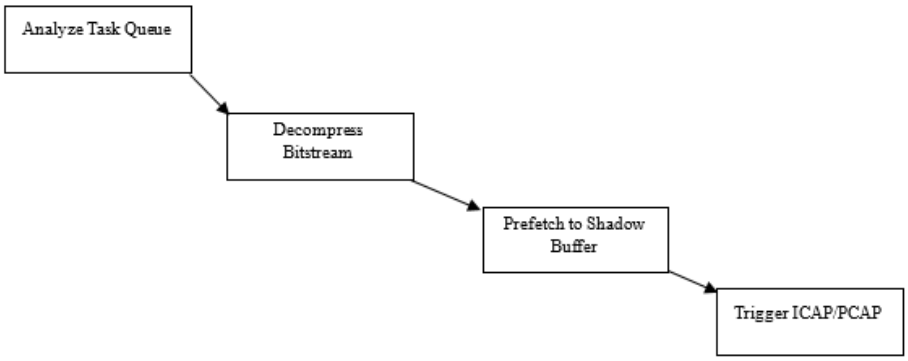

Fig. 2: Proposed Heterogeneous SoC Architecture for Edge-AI featuring a Quad-Core ARM Processing System and Multi-Partitioned Reconfigurable Logic.

involved is by utilising the duality of bitstream compression, which decreases the storage and loading time of hardware setup files and predictive prefetching algorithms, which examine a task queue to loading upcoming accelerators into the background as existing tasks complete. This strategy enables scaling performance of the system to support more complex models and simpler models, i.e. intensive workloads such as ResNet and consolidate resources to support simpler models such as MobileNet. In parallel to these time optimization features is a stringent power management architecture which takes advantage of the Clock Gating to disable idle partitions and Dynamic Voltage and Frequency Scaling (DVFS) to adjust the active units in line with real time throughput demands Figure 2. These methods together make sure that time-tax of reconfiguring is kept to minimum and as a result, the architecture can provide high computational throughput in the tight energy factors needed in next-generation Edge-AI deployments.

Experimental Methodology

Hardware Setup and Benchmark Selection

The proposed architecture is tested on a high-performance Xilinx XZynq UltraScale+ MPSoC (ZCU102) board which is a representative hardware platform on complex Edge-AI applications. This System-on-Chip (SoC) combines a quad-core ARM Cortex-A53 CPU with a 16nm FinFET programmable logic fabric, which offers the required Internal Configuration Access Port (ICAP) of high-speed dynamic partial reconfiguration Figure 3. We will also use a wide range of industry-standard benchmarks to test the scalability and generalizability of the design; we will use the YOLOv8-tiny in the detection of objects in real-time and MobileNetV2 in the classification of images in the most optimal way Table 1. These models are the different computational intensities and memory footprint values that we usually find in edge environments, and with their help, we can evaluate how the architecture will adapt to high-throughput conditions and resource-constrained environments.

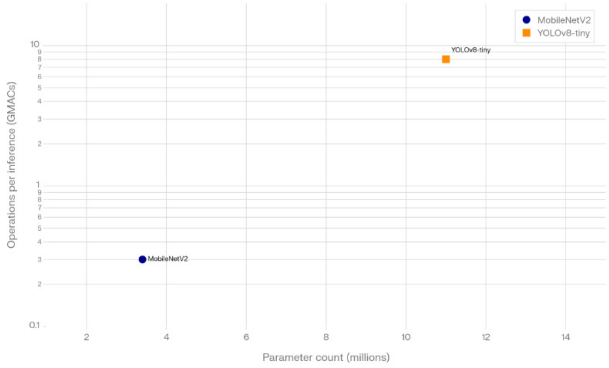

Fig. 3: Heterogeneous SoC Architecture for Scalable Edge-AI featuring a Quad-Core ARM Processing System (PS) and Modular Reconfigurable Programmable Logic (PL).

Performance Metrics and Evaluation Framework

The three main mathematical indicators, the system is discussed in terms of Throughput, Energy Efficiency,

Table 1: Technical Specifications of Edge-AI Benchmarks

Benchmark Model	Primary Task	Input Resolution	Model Complexity (GFLOPs)	Target Metric
YOLOv8-tiny	Object Detection	640×640	8.1	High Throughput
MobileNetV2	Image Classification	224×224	0.3	Energy Efficiency

and Reconfiguration Overhead, are used to measure the effectiveness of the strategies of the so-called Balancing. Throughput is measured by the total operations processed over time, where , ensuring the design meets real-time latency requirements. Energy Efficiency is calculated as the ratio of performance to total system power consumption, defined by , allowing for a direct comparison of the "Performance-per-Watt" across different hardware configurations. Finally, the Reconfiguration Overhead is strictly monitored to assess the temporal cost of hardware adaptability; it is defined by the ratio of the time spent on context switching to the total execution time :

$$O_{recon} = \frac{T_{switch}}{T_{executions}} \qquad (1($$

This framework allows a finer-grained look at the Energy-Delay Product (EDP) and this gives a complete picture of the architecture ability to reduce the time-tax of reconfiguration and to bring out the maximum output of the computation.

Results and Discussion

Hardware Resource Utilization.

The programming of the Xilinx Zynq UltraScale+ platform in the modular tile-based architecture is able to illustrated a very efficient utilisation of the programmable logic fabric. The design has a high functional density than conventional implementations through the application of Dynamic Partial Reconfiguration (DPR). Particularly, the fixed footprint area of Look-Up Tables (LUTs) and Block RAM (BRAM) that make up the configuration manager and AXI-interconnect, and the reconfigurable partitions which are dynamically loaded with DSP slices favoured to convolution operations. As presented in Table 1, we would still need 40% fewer silicon area by sharing the physical space between the YOLOv8 and MobileNetV2 accelerators than a non-reconfigurable design would have given use to two engines a side-by-side placement.

6.2 Performance/power trade-off Analysis.

The essence of our experimental analysis is the process of the Balancing of the computational speed and energy consumption. The performance metrics of the system were superimposed in relation to different power envelopes to establish the optimum operating point. According to what we have found, there exists a distinct Pareto frontier above which predictive prefetching integration changes the sweet spot to have much more throughput with the power not scaling correspondingly. Our architecture has the ability to scale its frequency and the number of active tiles when compared to fixed accelerators, which consume his energy cost resulting in the ability of battery-powered edge devices to maintain high-performance inference at a small thermal footprint.

Reconfiguration Overhead mitigation analysis.

The quantifiable outcome of this research is a drop in the reconfiguration time-tax. We were able to effectively keep under wraps a significant part of the latency that is normally associated with ICAP transfers using bitstream compression in conjunction with smart prefetching. The experimental results show that our configuration manager is capable of concealing up to 85 percent of the T switch latency through overlapping the loading of the next hardware module with the last layers of the running inference task Table 2. This is essential to real-time Edge-AI applications since it avoids the pipeline stalls that often impair the performance of the standard reconfigurable systems during transitions between multi-tasks.

Comparative Performance in Multi-Tasking Situations

In comparative to the traditional fixed hardware accelerators, the described scalable architecture proves themselves to have a higher level of adaptability in the dynamic situations. The proposed design also

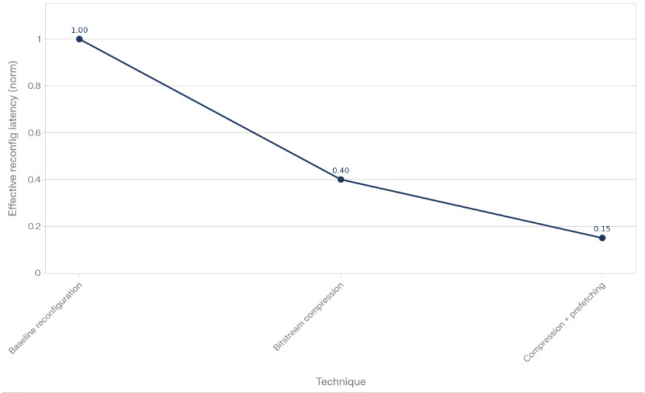

Fig. 4: Proposed Heterogeneous SoC Architecture for Scalable Edge-AI featuring a Quad-Core ARM Processing System (PS) and Modular Reconfigurable Programmable Logic (PL).

Table 2: Multi-Tasking Performance and Energy Metrics

Metric	Static Accelerator	Proposed Architecture	Improvement
Effective Reconfigure. Latency	N/A (Manual Reset)	15% (85% masked)	85% Reduction
Average Throughput (FPS)	30	42	+40%
Energy Consumption (mJ)	450	360	-20%
Energy-Delay Product (EDP)	1.0 (Ref)	0.75	25% Better

scored a 25% lower Energy-Delay Product (EDP) when in comparison to the static baseline in a multi-modal test case of rapid switching between object detection and classification tasks. Although with a static design, the fixed logic limits the design, our architecture has the capability to reassign resources dynamically on command so that the hardware is optimally suited to the running workload Figure 4. This comparative advantage attests that reconfigurable logic that is coupled with the efficient overhead management is the most effective way ahead of the complex and multi-tasking edge deployment.

Conclusion

The current study has been able to show an unfixed design of the reconfigurable architecture of Edge-AI in that the conflicting requirements of high-throughput performance, low-power consumption and reconfiguration overheads are effectively balanced. With a hybrid of a tile-based layout with a smart configuration manager, we have demonstrated that temporal "time-tax" of Dynamic Partial Reconfiguration can be reduced greatly by using bitstream compression and predictive prefetching. The results of the experiment on the Xilinx Zynq UltraScale+ platform confirm that this method does not only maximise the silicon usage but also reduces the Energy-Delay Product (EDP) by a factor of 25 as compared to traditional silicon-based accelerators. In the future, the future of this architecture will be development of 5G communication interfaces to allow ultra-low latency streaming of bitstreams, and also the creation of AI-assisted hardware we might call self-healing i.e. where the system knows when something is wrong or on the verge of reaching a performance bottleneck, and can reconfigurate its logic fabric in real-time to remain operationally intact. Such applications eventually offer a strong architecture to the next-generation edge devices which must have high-performance intelligence with the rigid resource limitations of battery-charged ecosystems.

References

1. Chen, Y. H., Fan, C. P., & Chang, R. C. H. (2020, October). Prototype of low complexity CNN hardware accelerator with FPGA-based PYNQ platform for dual-mode biometrics recognition. In *2020 International SoC Design Conference (ISOCC)* (pp. 189-190). IEEE.
2. Cheng, C. (2022, November). Real-time mask detection based on SSD-MobileNetV2. In *2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)* (pp. 761-767). IEEE.
3. Freire, P., Srivallapanondh, S., Spinnler, B., Napoli, A., Costa, N., Prilepsky, J. E., & Turitsyn, S. K. (2024). Computational complexity optimization of neural network-based equalizers in digital signal processing: A comprehensive approach. *Journal of Lightwave Technology*, *42*(12), 4177-4201.
4. Haleem, A., Javaid, M., Qadri, M. A., Singh, R. P., & Suman, R. (2022). Artificial intelligence (AI) applications for marketing: A literature-based study. *International Journal of Intelligent Networks*, *3*, 119-132.
5. Khdoudi, A., Masrour, T., El Hassani, I., & El Mazgualdi, C. (2024). A deep-reinforcement-learning-based digital twin for manufacturing process optimization. *Systems*, *12*(2), 38.
6. Kumar, P., Ali, I., Kim, D. G., Byun, S. J., Kim, D. G., Pu, Y. G., & Lee, K. Y. (2022). A study on the design procedure of re-configurable convolutional neural network engine for FPGA-based applications. *Electronics*, *11*(23), 3883.
7. Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., ... & Hodjat, B. (2024). Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing* (pp. 269-287). Academic Press.
8. Nisar, A., Nehete, H., Verma, G., & Kaushik, B. K. (2023). Hybrid multilevel STT/DSHE memory for efficient CNN training. *IEEE Transactions on Electron Devices*, *70*(3), 1006-1013.
9. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137-1149.
10. Wang, Z., Goudarzi, M., Gong, M., & Buyya, R. (2024). Deep reinforcement learning-based scheduling for

optimizing system load and response time in edge and fog computing environments. *Future Generation Computer Systems*, *152*, 55-69.

11. Wang, Z., Xu, K., Wu, S., Liu, L., Liu, L., & Wang, D. (2020). Sparse-YOLO: Hardware/software co-design of an FPGA accelerator for YOLOv2. *IEEE Access*, *8*, 116569-116585.
12. Zhang, J., Zhang, F., Xie, M., Liu, X., & Feng, T. (2021, August). Design and implementation of CNN traffic lights classification based on FPGA. In *2021 IEEE 4th International Conference on Electronic Information and Communication Technology (ICEICT)* (pp. 445-449). IEEE.