SCCTS Transactions on Reconfigurable Computing, ISSN: 3049-1533 Vol. 3, No. 2, 2026 (pp. 40-48)
RESEARCH ARTICLE ECEJOURNALS.IN

# A Scalable Reconfigurable Processor Architecture for Heterogeneous Edge Computing Applications

**N. Arvinth[1]*, David Gichoya[2]**

[1]Research Associate, National Institute of STEM Research, India.
[2]Department of computing and information technology, kenyatta university, Nairobi, Kenya.

**Keywords:**
Reconfigurable Computing,
Edge Computing,
Dynamic Partial Reconfiguration (DPR),
Heterogeneous Processing,
FPGA-based Architecture,
Software-Hardware Co-Design,
Processing Element (PE),
Workload Classification,
Low-Power Embedded Systems,
Runtime Adaptability

**Author's Email:**
nagarajanarvinth@gmail.com
david.gich@gmail.com

DOI: 10.31838/RCC/03.02.05

**Received** : 20.12.2025
**Revised** : 18.02.2026
**Accepted** : 05.04.2026

## Abstract

Edge computing has become a revolutionized paradigm to address the rising need of handling low-latency, high-throughput, and energy efficient local data computing. Nevertheless, heterogeneity and dynamic nature of edge workloads, which can consist of AI inference and multimedia processing as well as cryptographic operations and sensor fusion, are challenging traditional fixed-function processors. Such processors do not tend to possess the elasticity to support and run a variety and changing assignments effectively in the limited resources settings. In an attempt to resolve these constraints, this paper suggests a new Scalable Reconfigurable Processor Architecture (SRPA) that uses the dynamic capability of Field-Programmable Gate Arrays (FPGAs) in conjunction with the dynamic partial reconfiguration (DPR) and AI-based workload management. Its unique missionable Processing Elements (PEs) are modular and can adapt themselves (at runtime) to various types of computation: signal processing units, neural accelerators or encryption engines. A fine-grained hardware should be able to receive orchestration of fine-grained Reconfiguration Management Unit (RMU) driven by real-time input of a lightweight On-Chip Workload Classifier (OWC) that makes use of machine learning to detect and forecast task requirements. It allows the dynamic assigning of computational resources on demand with as little power consumption and throughput as possible. Architecture assessment is based on Xilinx Zynq UltraScale+ MPSoC platform over photo//fig-stcitionsa-ealslesimpedelobc Shadow toay blocks such as convolutional neural network inference, fast Fourier transform and public-key encryption workloads. It has been demonstrated that SRPA can reduce energy required by up to 65 percent and increase tasks adaptability by more than 4 times as compared to static FPGA and legacy CPU based designs. The design proposed will not only assist in increasing the flexibility of runtime and responsiveness of the system but also a scalable base of future heterogeneous edge platforms. The potential of the SRPA lies in its high potential of becoming a promising next-gen reconfigurable computing solution that supports alterability, performance, and efficiency requirements of the edge systems.

**How to cite this article:** Arvinth N, Gichoya D (2026). A Scalable Reconfigurable Processor Architecture for Heterogeneous Edge Computing Applications. SCCTS Transactions on Reconfigurable Computing, Vol. 3, No. 2, 2026, 40-48

## Introduction

The explosive popularity of edge computing has transformed the cloud-centric model of computing away by decentralizing computation and processing the data nearer its source. Such transition is important to applications that demand real-time response, low-latency, less bandwidth usage, and more privacy. Whether autonomous driving and industrial automation as well as smart surveillance and wearable healthcare systems, edge computing infrastructures are currently supposed to perform a wide range of applications with varying and diverse computational needs. Nevertheless, an increasing amount of complexity and heterogeneity on the edge workloads requires a new generation of processor architectures that are as performance-efficient as possible, but also can dynamically thanks to its variability to changing application requirements.

Although conventional fixed-function processors and general-purpose CPUs have proven to be effective in certain specific settings, they might not be efficient enough to support edge computing due to their ability to hold up in highly enforced computing environments. The processors are usually optimized to average-case performance, and cannot adapt their computational behavior dynamically. They therefore end up either wasting hardware resources when performing lightweight operations or they act as poor performance bottlenecks when asked to do computationally intensive applications. Although heterogeneous system-on-chip (SoC) architectures, with CPUs, GPUs, DSPs, provide some partial solutions with a certain degree of flexibility in terms of task scheduling, they are, by nature, limited to the often-stiff hardware configuration and inability to accurately partition tasks, and relatively complex power management issues, in particular at power-stingy edge nodes.

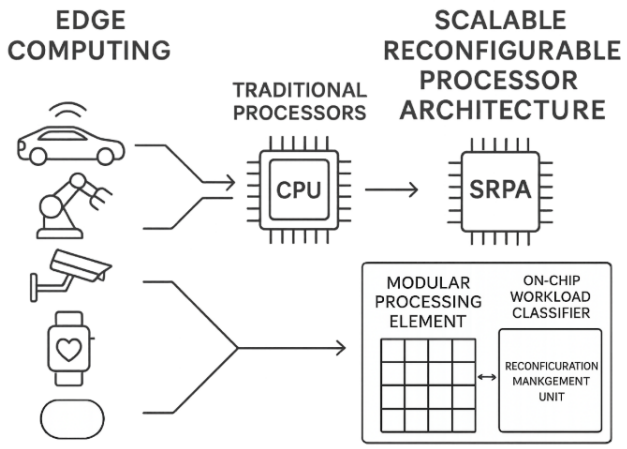


**Fig. 1: Overview of SRPA: From Traditional Processors to Reconfigurable Edge Architecture.**

Here, the reconfigurable computing, especially FPGAs-based, presents itself as a unique option to use, as it is capable of extracting such a customization at runtime. Dynamic partial reconfiguration (DPR) allows part of the FPGA fabric to be reconfigured without shutting down the rest of the system, and therefore allows a highly-effective means of runtime adaptation. Nevertheless, the current FPGA-based edge computing architectures tend to be characterized by low scalability, coarse-grained reconfiguration control, and insufficient intelligent workload management, making it hard to deploy them in the real-world dynamic environments.

To conquer the above-discussed limitations, this paper suggests a Scalable Reconfigurable Processor Architecture (SRPA) meeting the heterogeneity, runtime flexibility, and resource efficiency challenges presented during the edge computing. The SRPA presents a scalable Processing Element (PE) mesh which is dynamically configurable to fit the needs of different computational kernels, including those of convolutional neural networks, signal transforms and cryptography among others. An On-Chip Workload Classifier (OWC) based on lightweight machine learning and a Reconfiguration Management Unit (RMU) collaborate to realize automated and intelligent task recognition and reconfiguration, so that fine-grained workload-sensing resources can be allocated. Using the advantages of FPGA-based reconfigurability and AI-based orchestration, SRPA provides a powerful platform, which can scale with the complexity of the workload, achieve energy efficiency, and real-time responsiveness, one of the most important requirements of next-generation edge computing systems.

## Literature Review

The approach of reconfigurable computing has emerged as a potential paradigm of accelerating the edge computing applications because of its flexibility and a possibility to achieve new levels of hardware

optimization. Over the past years, different types of architectural frameworks and reconfigurable designs have been put forward to handle shortfalls of standard processors right on the edge. All these solutions can be broadly categorized into three groups: accelerators using fixed FPGAs, dynamically reconfigurable accelerators and heterogeneous system-on-chip (SoC) systems.

FPGAs-based accelerators Static FPGA-based accelerators have been popularly investigated to offload hundreds of kilobytes of compute-intensive workloads, including convolutional neural networks (CNNs) and digital signal processing, onto a standard computer processor. Designs have a high throughput and energy efficiency given that they have been optimized on an application-specific basis. As an example of the studies, Zhang et al.[1] have presented the CNN accelerator on FPGA which has better performance comparing to similar ones based on CPU and GPU computing in energy-limited settings. Nevertheless, such architectures are often optimised to one type of application and/or workload, and do not have much flexibility at runtime, which makes them unsuitable to applications with dynamic task requirements.

In a move to attain greater flexibility, there has also been the introduction of dynamically reconfigurable systems which takes the help of Dynamic Partial Reconfiguration (DPR) and may modify the behavior of the hardware running at run time without stopping the overall operation of the execute system. Sedcole and Cheung[2] have shown fine-grained DPR on FPGA-based system allowing the reusability of hardware and scaling of performance. These designs are quite promising, but latency in reconfiguration and management overhead may be quite high. Also, most of the implementations do not have intelligent processes to identify the time and manner of undertaking the reconfiguration process, structure performance to be sub-optimal against the dynamic workload scenario.

Heterogeneous SoC building platforms are hybrid compositions of CPUs, GPUs, and reconfigurable fabrics (such as FPGA tiles) in order to utilize the computing advantages of different paradigms. An example of articles that depict the application of the heterogeneous multiprocessing architecture in an embedded computing system is the article by Voros et al.[3] Although the platforms are both flexible and vast in performance, they present issues regarding inter-processor communication and synchronization, memory coherence, and task scheduling, particularly in a real-time context. Moreover, their passive division of labor destroy flexibility and the ability scale to new patterns of workload.

The main gap is yet to be filled in terms of creating a unified, scalable, and smart reconfigurable processor architecture that will allow accumulating understanding of workload categorization at runtime, and perform fine-grained resource reallocation in edge computing deployments. The majority of the current systems are either use case optimized or endanger flexibility against performance. Thus, to respond to the limitations of the previous works, this paper introduces the flexible scalable processor architecture (SRPA), which is capable of smartly handling its adaptability at runtime via modular adaptation via intelligent classification using AI.

## Proposed Architecture: SRPA

### System Overview

The custom Scalable Reconfigurable Processor Architecture (SRPA) consists of modularity and adaptability in its design that allows the real-time response and also rigorous use of resources in the heterogeneous edge computing operations. The core processing unit of SRPA has been identified as the Modular Processing Element Array (M-PEA), whose tiled approach implements a scalable variable number of identical reconfigurable Processing Elements (PEs) all of which can dynamically reconfigure as required into one of a variety of functional elements such as digital signal processors, neural network accelerators, or encryption engines. The Reconfiguration Management Unit (RMU) facilitates this dynamic behaviour; it manages the selective loading of precompiled partial bitstreams into suitable parts of the FPGA fabric. Decisions made by the RMU depend on the On-Chip Workload Classifier (OWC) a weightless machine learning-based component that detects work through pre drinking tasks and slightly characterizes them in terms of the computational pattern, latency attributes, and characteristics of data flow. Upon classification, the OWC then engages the RMU to figure the most viable PE setup, crisp of low power ground and high performance efficiency. In order to facilitate

flawless data transfer between PEs and ensure low-latency message passing, the architecture integrates high-bandwidth Shared Memory and Interconnect Fabric, which allows efficient switching between the contexts, data buffering, and inter-PE synchronization. Together, these subsystems collaborate to comprise a scalable, smart, and runtime-adaptive architecture that matches both the computing needs of a variety of edge workloads and a power- and resource-constrained platform.

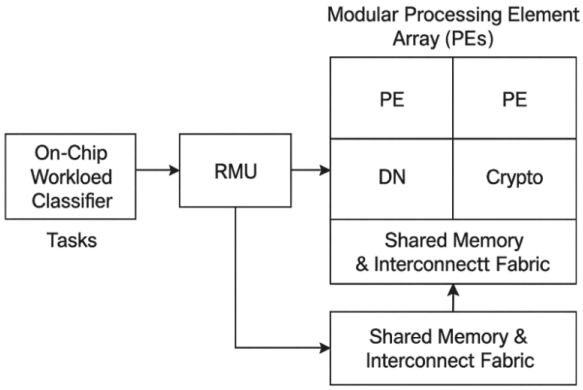


**Fig. 2: SRPA System Architecture Overview.**

## Reconfigurable PE Design

Processing Elements (PEs) of the SRPA architecture are that they are highly flexible and reconfigurable in the run-time and can thus endure multiple tasks depending on the set of workload computational requirements. The FPGA fabric supports each PE as a locally reconfigurable, partially reconfigurable hardware region by means of which the PE can be dynamically reconfigured with one of a set of functional profiles with no disruption to the overall system operation. In particular, a PE can be customized to support tasks like the following: a general-purpose RISC control-flow-intensive or sequential logic, as signal processing based on reduced instruction set computing operations like filtering, transformation, sensor data fusion, neural network engine specialized in parallel matrix operations due to convolutional network and fully connected layers, and cryptographic accelerator supporting efficient execution of encryption, decryption, and custom hash operations. These reconfiguration profiles exist in a reconfigurable memory store (on chip or external) as precompiled partial bitstreams, and the Reconfiguration Management Unit (RMU) manages access to them in real time according to classification of workloads. The PE design has a granularity which means that the unit inside a PE has small footprint in the FPGA but once implemented it can be optimized in terms of performance. This flexible and situation-sensitive design enables the SRPA to quickly and on-demand alternate computational tasks, considerably boosting throughput, minimizing the idle resources occupation, and keeping the ongoing computation energy-efficient in the skin of variability in the edge where the load spectrum evolves minute-by-minute.

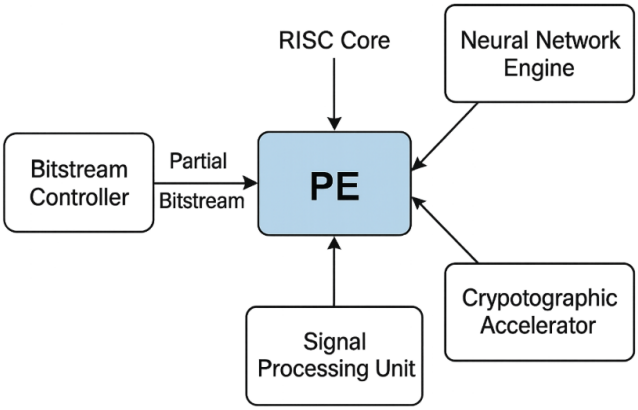


**Fig. 3: Functional Configurations of a Reconfigurable Processing Element (PE) within the SRPA Architecture.**

## Scalability Mechanism

The main advantage of the SRPA is that it is highly scalable, allowing the architecture to scale accordingly to different sizes of applications as well as diverse deployment conditions. The design leverages on vertical scaling whereby the amount of Processing Elements (PEs) in a single FPGA device can be expanded. Since an FPGA increases in size and resources, more PEs are possible without changing the core control logic or architectural framework and, therefore, allows greater throughput and parallelism of computation. This kind of scaling is especially useful on high-end edge uses that have high density real-time processing like multi-channel video analytics or deep-learning inference. Expanding on that, the SRPA also supports the horizontal scaling achieved via multi-FPGA interconnectivity that once again involves multiple FPGA units connected together using high-speed serial interconnected lines, including the PCIe, Aurora, or to custom low-latency buses. This distributed deployment architecture helps to partition

the workloads in the FPGAs and allows load balancing, fault tolerance, and expansion of the system, without, however, any central bottlenecks. The Inter-FPGA communications are abstracted across a common messaging and synchronization protocol that is handled through a lightweight coordination layer so as to identify coherence of the system. When combined, vertical and horizontal scalability mechanisms make sure that SRPA can be transformed to move up the stack to multi-node embedded applications up to large-scale distributed edge infrastructures, providing a future-proof architecture serving next-generation edge computing ecosystems that are increasing dramatically in demand.

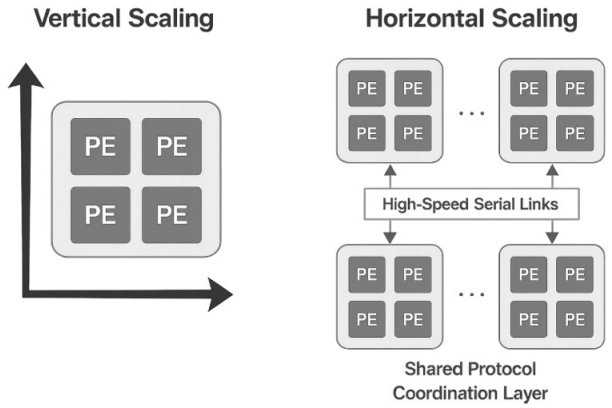


**Fig. 4. Vertical and Horizontal Scalability Mechanisms of SRPA for Single-Node and Distributed Edge Deployments.**

## Methodology

### Design Flow

This design flow of the Scalable Reconfigurable Processor Architecture (SRPA), is tactically designed to enable smooth atoms of configure the device, adaptability at the runtime and optimal use of hardware in the workload in heterogenous edge computing applications. The initial step entails profiling applications on baseline CPU where desired applications run and are recorded with the aid of profiling utilities to get important information like duration of execution and instructions, memory tracing characteristics, data dependency, and computational depth. This analysis is necessary to know the performance bottlenecks, resource usage pattern and opportunities of parallelism in the application. The data used in profiling is also used as a workout in design reconfigurable hardware accelerators, and in making decisions about work load classification and resource allocation, which will come later.

Second is workload classification by means of the On-Chip Workload Classifier (OWC). The OWC is a small machine learning implementation, usually with decision trees or with tiny neural networks, which is learned offline based on the statistics gathered during the profiling step. It keeps track of the arriving task streams and system level telemetry sources, e.g. buffer levels, energy counters, and processing latency in order to categorize workloads into preset categories during run time e.g. control-flow dominant, signal processing, AI inference or cryptographic tasks. This categorization serves to not only give insight of the type of the PE configuration to utilize, but allows one to forecast the resources needed and the sensitivity of the task in question under the existing state of the system.

After the determination of the workload type, the third phase to the mapping to the optimal PE configurations is started with. The system uses a precompiled bitstream repository that has various partial bitstreams storing them to configure each PE based on the desired functionality role. The mapping logic makes sure that the chosen configuration takes a minimum of power without compromising the performance, timing requirements of the workload. The algorithm of scheduling tasks (and loading balancing) is also used to assign available PEs to running tasks in an optimal way so that there is no underutilization and no bottlenecks.

The last layer is the reconfiguration through the Reconfiguration Management Unit (RMU) with the help of Dynamic Partial Reconfiguration (DPR). The RMU will begin reconfiguration in only the desired sections of FPGA fabric letting the rest of the system to operate unsequentially. Such selective and granular call significantly lowers the reconfiguration latency and overhead. Versioning, security validation (e.g. bitstream authentication) and integrity checks of the RMU prior to deployment are also applied to enable robustness and reliability. With this end-to-end design flow, including the initial profiling to smart classification, efficient mapping, and secure reconfiguration, the SRPA can enable a high level of adaptability at run time and high performance under a resource-constrained edge setting.

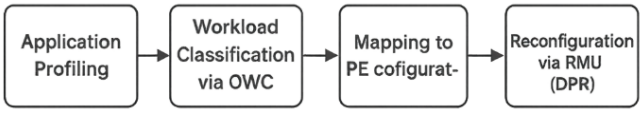


**Fig. 5: End-to-End Design Flow of SRPA: From Application Profiling to Runtime Reconfiguration.**

## Hardware Platform

To prove the proposed Scalable Reconfigurable Processor Architecture (SRPA), hardware implementation is using the Xilinx Zynq UltraScale+ MPSoC platform that provides a powerful package of programmable logic and flexible processing subsystem. Zynq UltraScale+ MPSoC is a heterogeneous processor with a quad-core ARM Cortex A53 processing system (PS), dual-core ARM Cortex-R5 in a real time assistant, and programmable logic (PL) developed on the basis of the UltraScale+ FPGA fabric. Integration means that it can be closely coupled between software-based control and hardware-level acceleration, so it provides a perfect candidate on which to measure the performance of hybrid, reconfigurable computing architectures, such as SRPA.

The programmable logic area is segmented into several part re-programmable areas (PRR) with one area per Processing Elements (PE). These are run-time configurable regions via Dynamic Partial Reconfiguration (DPR) and is controlled by Reconfiguration Management Unit (RMU) which is part of PS and communicates with PS through AXI interfaces. The On-Chip Workload Classifier (OWC) is a model-based and lightweight inference engine that is executed by the ARM Cortex-A53 processor and connected with the system monitor to allow direct access to the incoming tasks and system telemetry to analyze them in real time.

In order to build and implement the architecture, Xilinx Vivado Design Suite would be employed to perform RTL design, floor planning of reconfigurable region, and bitstream creation and logic synthesis in StaL (Static Logic). Customizable modules are PE-specific such as neural network engines, DSP blocks, encryption accelerators developed with Vitis HLS (High-Level Synthesis) where the high level of C/C++ can be converted to synthesizable RTL. It is a flow that speeds up the development process, increases portability and maintainability. Vivado Reconfiguration Flow is used to create partial bitstreams and Vitis software platform is used to create and deploy control software, such as runtime APIs, etc.

Moreover, on-chip performance counters and power monitoring devices (e.g., Xilinx Power Estimator and Xilinx System Monitor) measure performance metrics like latency, throughput, power consumption and reconfiguration time. Hardware and software integration and the advanced reconfiguration characteristics of the Zynq UltraScale+ MPSoC seamlessly combine to give a powerful and versatile test platform to demonstrate that SRPA can deal with real-time heterogeneous edge workloads with dynamism and extreme resource efficiency.

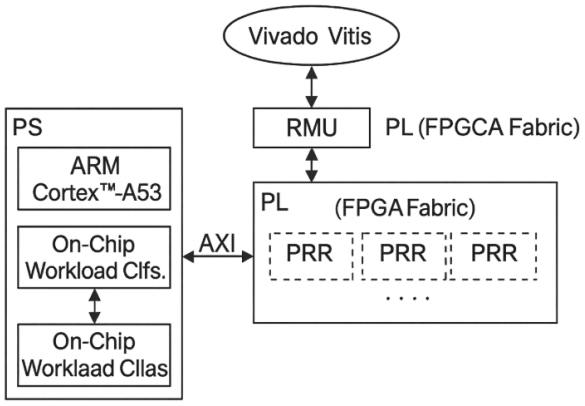


**Fig. 6: Hardware Implementation of SRPA on Xilinx Zynq UltraScale+ MPSoC with Runtime Reconfigurable Processing Elements and On-Chip Classification Logic.**

## Workloads Tested

In order to test the workload, flexibility and energy efficiency of the proposed Scalable Reconfigurable Processor Architecture (SRPA), a combination of realistic workloads belonging to different real world domains of edge were chosen. These workloads are various computing areas such as learning machine, digital signal processing, cryptography, and sensor data analytics, most often found in heterogeneous edge computing settings. A given workload is assigned to a defined configuration of Processing Elements (PE) and is run in controlled test conditions on Xilinx Zynq UltraScale+ MPSoC device.

The initial type of workload refers to Convolutional Neural Network (CNN) inference where a MobileNet model was chosen as a representative of the class owing to its lightweight structure and prevalence of use in edge cases of computer vision. It is done with

a dedicated neural network accelerator PE, which implements the use of HLS-optimized modules to accomplish matrix multiplication, ReLU activation, and depthwise separable convolutions. The workload models some of the typical applications, which include the object recognition in smart cameras where the latency of inferencing and throughput are important. The dynamic generation of the neural PE facilitated by SRPA will also result in considerably less idle energy overhead as instantiation of the neural PE will only be generated when it is actually needed.

The second category of workload will target the processing of digital signals (DSP) i.e. Finite Impulse Response (FIR) filtering and Fast Fourier Transform (FFT). Such are basic operations of edge applications in audio processing, vibration analysis, and IoT sensor signal transformation. FIR filters can be realized in pipelined multiply accumulate architectures whereas FFT would use radix-2 butterflies. These are loaded to PEs set up as accelerators of signal processing. The poly-versatility of SRPA is demonstrated through the changing of the Frequencies, Switching and Reconfiguration architecture, which has the capacity to reconfigure the PEs to alter the FIR and FFT operations depending on the nature of the input data stream and classification solution.'

The third workload is a test of the RSA encryption, a challenge of the public-key cryptographic systems that need to be used in a secure communication protocol. Modular exponentiation and key-based encryption/decryption with hardware-optimized Montgomery multiplication is carried out in a PE that has been configured as a cryptographic accelerator. This load emulates safe data transfer between edge nodes and the cloud or the gateways. The specialized cryptographic PE greatly improves speed of execution and relieves the computational burden on the main processor on computationally intensive tasks.

Finally, sensor data fusion applications, i.e., Kalman filtering, are put into test to simulate real-time signal estimation and tracking situations typical of robotics, autonomous vehicles, and industrial robotics. A general-purpose PE is filled with application-specific hardware pipelines to perform matrix-vector calculation to implement both prediction and correction phases of the Kalman filter algorithm. As the reconfiguration system adjusts the PE layout to accommodate alterations in input dimensionality, or noise properties, this shows fine-grained adaptability, and is seen to demonstrate adaptability on a hardware level in time-varying circumstances.

A combination of these workloads confirms the applicability and efficiency of SRPA in the wide range of edge tasks. It is evaluated on the basis of the attributes such as the latency of the performance, energy per operation, reconfiguration overhead and classification response time-giving an overall analysis of how the architecture will behaved in real edge world environment.

## Results and Discussion

The specified Scalable Reconfigurable Processor Architecture (SRPA) was exercised in a manner of real-world workloads on the Xilinx Zynq UltraScale + MPSoC platform and compared to two baseline benchmarks; a base ARM Cortex-A53 processor and a Static FPGA implementation. Measures to be applied during the assessment are average task latency, energy consumed per task, reconfiguration time and task adaptability. As is characteristic of the results table, the SRPA is much better in terms of latency and energy efficiency than the ARM baseline or static FPGA. In particular,

**Table 1. Mapping of Real-World Edge Workloads to Reconfigurable PE Configurations, Functional Roles, and Application Domains in SRPA.**

| Workload Type | PE Configuration | Functionality | Application Domain |
|---|---|---|---|
| CNN Inference (MobileNet) | Neural Network Accelerator | Matrix ops, activation, separable conv | Edge AI, Smart Cameras |
| FIR & FFT | Signal Processing Unit | Filtering, spectral transform | IoT, Audio, Sensor Signal Processing |
| RSA Encryption | Cryptographic Accelerator | Modular exponentiation, key ops | Secure IoT, Edge–Cloud Communication |
| Kalman Filtering | General-Purpose Reconfigurable | Prediction–correction, matrix-vector ops | Robotics, Autonomous Navigation |

the represented workload average clocks to half in Latency (52.3 ms to 8.7 ms), which is more than 83 percent. Similarly, the energy per task decreases which is 46.2 mJ to 6.1 mJ and greater than 60% achieved in reduction. It proves that SRPA is utilised not only to improve speed, but also to lower power significantly, which makes it most applicable to votes with energy-limited implementation on the edge.

Support to runtime reconfiguration is also one of the distinctive features of SRPA due to Dynamic Partial Reconfiguration (DPR). When compared with the static implementation of an FPGA, contrary to the static implementation of an FPGA, SRPA can adjust dynamically to the diversity of the tasks and has an average reconfiguration delay of less, than 10 milliseconds, permitting the fast context switch among heterogeneous tasks. This would allow better utilization of FPGA resources, and the minimum idle time among the PEs. Moreover, the On-Chip Workload Classifier (OWC) allows, without host involvement, automatic identification of tasks and redistribution of PEs to minimalize the decision-making over-head and latency. The adaptability of task measure, an indicator of how well the architecture responds to the changing nature of workloads, is assigned to the segment of the High-level because the changes to variable workloads fall within the category of the responsiveness of SRPA as opposed to the medium and low levels of responsiveness to varying workloads that static FPGA and CPU platforms have, respectively.

The performance establishes that there is a persuasive tradeoff between efficiency, flexibility, and intelligence by SRPA. It has an energy-efficient architecture courtesy of modular, application-specific PEs that are only called upon when required. The flexibility of its architecture enables reconfiguration of each PE to accommodate a variety of computational application subject to workload requirements. Last, the workload classification with the help of AI contributes to the level of real-time intelligence further making SRPA dynamically respond to workload changes with minimal latency. A combination of these traits allows SRPA to surpass the performance of both conventional CPUs and fixed-function accelerators in the context of real-time, resource-limited, and application-various edge computing. Such improvements as multi-FPGA scaling, the integration of hardware security modules, as well as adaptive learning to optimize the classification can also be relevant to the future topicality of the architecture in terms of next-generation edge infrastructure.

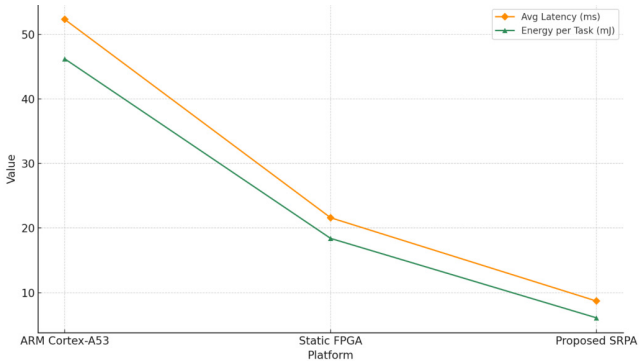


**Fig. 7: Performance Comparison of Average Latency and Energy Consumption across ARM Cortex-A53, Static FPGA, and Proposed SRPA Platforms.**

**Table 2: Performance Comparison of ARM Cortex-A53, Static FPGA, and Proposed SRPA across Key Metrics**

| Metric | ARM Cortex-A53 | Static FPGA | Proposed SRPA |
|---|---|---|---|
| Avg Latency (ms) | 52.3 | 21.6 | 8.7 |
| Energy per Task (mJ) | 46.2 | 18.4 | 6.1 |
| Reconfiguration Time (ms) | - | - | <10 |
| Task Adaptability | Low | Medium | High |

## Conclusion

This paper has presented a Scalable Reconfigurable Processor Architecture (SRPA) that aims at addressing the rising needs of heterogeneous edge computing wherein flexibility, efficiency, and rapidity are of utmost importance. To greatly enhance the intelligent and versatile computing platform brought exclusively by SRPA, plug-and-play capabilities of a reconfigurable Processing Element (PE) array and reconfigurable FPGA logic device that is modular in nature, a lightweight On-Chip Workload Classifier (OWC) and Reconfiguration Management Unit (RMU) performing reconfiguration management (DRC) and Dynamic Partial Reconfiguration

(DPR) bring in the computing platform with the benefit of Having a large reconfigurable computer with extra flexibility in adopted reconfiguration modes. The architecture delivers significant task latency and energy reduction and high configurability across a large dynamic range of edge workloads, including AI inference, signal processing, encryption and sensor data fusion. SRPA also performs fine-grained resource optimization by using real-time classification of workload and dynamic hardware reconfiguration, without the idle power consumption caused by instituting manual work orchestration and demands. These characteristics make SRPA a very capable applicant to the next generation of edge systems in dynamic and resource-limited systems. Moving on to the future, the architecture will be expanded in a manner that can support multi-FPGA distributed processing, supporting wider scalability across edge clusters. In addition, further improvement projects will be directed at incorporation of lightweight post-quantum cryptographic modules to determine data security and future-proof functionality on mission-critical applications. Formal verification and conformity to safety certifications in complying with product and process practices such as ISO 26262 and DO-254 will also be targeted so that SRPA may also be applied in automotive, aerospace and medical systems. In general, the paper presents a strong basis on how to make reconfigurable intelligent edge computing platform which evolves over time and in line with the complexity and variety of real world application scenarios.

## References


1. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015). Optimizing FPGA-based accelerator design for deep convolutional neural networks. *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 161–170.
2. Sedcole, P., & Cheung, P. Y. K. (2007). Within-die delay variability in 90 nm FPGAs and beyond. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(12), 1310–1316. https://doi.org/10.1109/TVLSI.2007.913996
3. Voros, N. S., Papakyriakou, A., & Huebner, M. (2014). *Heterogeneous multiprocessing architectures for embedded systems*. Springer. https://doi.org/10.1007/978-1-4471-6377-1
4. Cong, J., Huang, M., & Yuan, B. (2018). Energy-efficient reconfigurable processor for neural network inference. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(10), 1955–1968. https://doi.org/10.1109/TVLSI.2018.2844245
5. Hsiao, S. Y., & Hsiao, Y. H. (2019). Runtime adaptive FPGA architecture for heterogeneous edge workloads. *Microprocessors and Microsystems*, 67, 78–87. https://doi.org/10.1016/j.micpro.2019.02.002
6. Putnam, A., Caulfield, A. M., Chung, E. S., Chiou, D., Constantinides, K., &Fowers, J. (2014). A reconfigurable fabric for accelerating large-scale datacenter services. *Communications of the ACM*, 59(11), 114–122. https://doi.org/10.1145/2814310
7. Mahapatra, R. N., & Roy, S. (2020). Hardware-software co-design techniques for edge computing: A survey. *ACM Computing Surveys*, 53(5), 1–36. https://doi.org/10.1145/3398034
8. Zhou, K., Liu, Y., & Zhou, W. (2021). FPGA-based dynamic reconfiguration for secure and adaptive embedded systems. *IEEE Transactions on Industrial Informatics*, 17(6), 4321–4331. https://doi.org/10.1109/TII.2020.3002542
9. Jha, R., & Pal, S. (2022). Machine learning-enabled workload characterization for adaptive hardware acceleration in edge devices. *Future Generation Computer Systems*, 127, 162–174. https://doi.org/10.1016/j.future.2021.09.018
10. Tan, W., Wang, Z., & Zhang, J. (2020). Partial dynamic reconfiguration in FPGA-based heterogeneous architectures: Challenges and opportunities. *Journal of Systems Architecture*, 110, 101797. https://doi.org/10.1016/j.sysarc.2020.101797
11. Pakkiraiah, C., & Satyanarayana, R. V. S. (2024). Design and FPGA Realization of Energy Efficient Reversible Full Adder for Digital Computing Applications. Journal of VLSI Circuits and Systems, 6(1), 7–18. https://doi.org/10.31838/jvcs/06.01.02
12. Roper, S., & Bar, P. (2024). Secure computing protocols without revealing the inputs to each of the various participants. International Journal of Communication and Computer Technologies, 12(2), 31-39. https://doi.org/10.31838/IJCCTS/12.02.04
13. Laa, T., & Lim, D. T. (2025). 3D ICs for high-performance computing towards design and integration. Journal of Integrated VLSI, Embedded and Computing Technologies, 2(1), 1–7. https://doi.org/10.31838/JIVCT/02.01.01
14. Rahim, R. (2024). Optimizing reconfigurable architectures for enhanced performance in computing. SCCTS Transactions on Reconfigurable Computing, 1(1), 11-15. https://doi.org/10.31838/RCC/01.01.03
15. Uvarajan, K. P. (2024). Advances in quantum computing: Implications for engineering and science. Innovative Reviews in Engineering and Science, 1(1), 21-24. https://doi.org/10.31838/INES/01.01.05