Quantum-Inspired Reconfigurable Computing for Accelerated Matrix Computations in High-Performance Embedded Systems

Abd-El-Hamid Bensafi[1]*, Layla H. Sulaiman[2]

[1]Abou Bekr Belkaid University of Tlemcen, Algeria, Algeria
[2]Doha Institute of Systemic Foresight, Qatar

Keywords:
Quantum-inspired computing,
matrix computation,
reconfigurable architecture,
embedded systems,
FPGA acceleration,
dynamic partial reconfiguration,
pseudo-superposition,
parallelism.

Author's Email:
aeh.bensafi@gmail.com

https://doi.org/10.31838/ESA/03.02.06

Received : 20.01.2026
Revised : 06.02.2026
Accepted : 08.03.2026

Abstract

The emergence of data-hungry systems in the form of artificial intelligence, real-time signal processing, and scientific computing systems inside embedded systems has set extremely high requirements to the computational capabilities, electrical energy efficiency and flexibility. The computational backbone of many of these applications is matrix computations, that typically face bottlenecks in traditional processor architectures because of individual resources having no parallelism, and a lack of dynamism in resource allocation. In order to overcome such limitations, this paper introduces a new reconfigurable computing implementation, called Quantum-Inspired Matrix Engine (QIME) that emulates all three of these important quantum computing concepts of superposition, parallelism, and entanglement in a classical hardware setting. The system of the proposal will be applied to the Field-Programmable Gate Arrays (FPGAs), and here, the system will be used to incorporate a pseudo-quantum state encoding process that enables the representation and processing of multiple matrix states simultaneously. QIME also has dynamic partial reconfiguration (DPR) so as to further have flexibility and efficiency in matching the architecture in real time with the nature and the complexity of the operations performed in the matrix. Also, there is approximate computing, where computational precision is traded off against performance where the tradeoff is allowed to increase energy profiles under workload variability. Experimental analysis of some typical benchmarks on matrices, including multiplication, decomposition, and convolution, shows that for representative operations QIME provides up to 6.3x speedup and 4.1x energy efficiency improvements over typical matrix co-processors and embedded GPUs. These findings will support the viability of quantum-inspired design paradigms in moving forward the performance and scalability of high-performance embedded systems, especially in the edge, and constrained circumstances where conventional quantum hardware is infeasible.

How to cite this article: Bensafi AEH, Sulaiman LH (2026). Quantum-Inspired Reconfigurable Computing for Accelerated Matrix Computations in High-Performance Embedded Systems. SCCTS Journal of Embedded Systems Design and Applications, Vol. 3, No. 2, 2026, 48-58

Introduction

Matrix operations are indispensable as basic processing primitives in a tremendous range of embedded systems applications, such as but not confined to digital signal processing (DSP), computer vision, control systems, robotics or machine learning inference. Such programs are becoming more demanding in more than just high computational throughput, as they need to become responsive in real-time as well as with low power usage, particularly in edge computing where resources are extremely limited. General-purpose computing platforms (GPPs), graphics processors (GPUs), and digital signal processors (DSPs) are the typical examples of conventional computing platforms that do not provide as adequate a trade-off between energy efficiency and computation speed, on the one hand, versus to architectural versatility and programmability, on the other hand.

At the same time, through the development of quantum computing, it has become clear that a whole new set of paradigms exist to speed up computational tasks through use of principles like superposition, entanglement, and quantum parallelism. Though laboratory quantum hardware is in an early development stage, theoretical quantum computation has important advantages, including the exponential representation of the state space and inherently parallel computation, appealing uses of such properties are potential improvements to classical computing hardware. Another innovation has come in the form of a new set of algorithms and architectures, which collectively have become known as quantum-inspired computing: exploring quantum-like behaviours in classical hardware settings.

Although remarkable progress has been made in the topic of the matrix acceleration, some fundamental issues still exist and hamper scalability and flexibility in performance of the embedded systems. The majority of classical, fixed workload acceleration-based engines, e.g., systolic arrays, SIMD pipelines, and GPUs, have a limited range of work loads because they lack the architectural interpretive flexibility, to cross-utilize and efficiently accomplish a wide variety of matrix operations of varying dimensions, sparsity patterns, and precision requirements. As a consequence, these architectures tend to be unnecessarily underutilized or power-inefficient in dynamic workloads, especially when used in real-time settings such as edge AI and autonomous control. In addition, the fixed-function hardware is by necessity limited in its capability to scale, be flexible with workload, which causes inefficiencies in energy use and throughput. Notably, although the theoretical potential of quantum computing regarding faster linear algebra operations has been demonstrated, its presence in classical embedded hardware has not been highly investigated in practice. Possibility to simulate quantum phenomena, e.g. superposition, entanglement and parallelism, in reconfigurable classical systems is an opportunity offered to overcome constraints of classical architectures. These difficulties highlight the necessity to develop new flexibler, energy-aware matrix computation framework to be applied to high-performance embedded setting.

In order to overcome these problems, this paper proposed a new powerful reconfigurable computing paradigm called the Quantum-Inspired Matrix Engine (QIME). The QIME architecture is based on the classical emulation of quantum principles, and allows to improve the performance of matrix computation operations without the usage of quantum hardware. In particular, it employs a construct-called pseudo-quantum state encoding-to encode multiple logical matrix states into parallel data structures, which is in effect a simulation of the quantum superposition concept. Moreover, tightly-coupled processing used to have shared intermediate output by utilising entangled computation units, which are similar to those of quantum entanglement. One of the fundamental innovations in QIME is the use of hybrid parallelism, in order to enable maximum throughput with matrix operations of different types and dimensions, by a combination of data level and instruction level parallelism.

QIME is built on the Field-Programmable Gate Arrays (FPGA) since it requires flexibility during run time architecture adaptation. With dynamic partial reconfiguration (DPR) the system is able to change its internal structure on the fly depending on the demands of the ongoing task computation intensity and type, providing it with efficiency and scaling properties. To complement the already achieved increase in performance in error-tolerant workloads like deep learning, approximate computing units are work-in together to achieve a trade-off in precision and performance. This enables it to work in limited power budgets but its computational deadlines are fulfilled. In total, the QIME offers a matrix computational engine grounded in quantum inspiration, reconfigurability and energy efficiency that can be used to accelerate an extremely broad scope of embedded applications.

The main idea of the proposed research is to discuss the possibilities to successfully utilize quantum-inspired computational concepts in classical, reconfigurable hardware in order to accelerate matrix computations in the embedded contexts. Particularly, the proposed study is getting designed to create a quantum inspired reconfigurable architecture which has been shown to support high-throughput and energy-efficient computing of matrix operations with classical logic devices. One of them is realization of modular processing units that simulate quantum quantum properties of superposition and entanglement which allows the system to dynamically change according to the workload characteristics. The architecture (the Quantum-Inspired Matrix Engine, or QIME) will be deployed on FPGA platforms, which in turn will enable direct comparison with currently active acceleration strategies, such as CPUs, GPUs, and fixed-function co-processors. The paper also explores trade-offs in precision / energy efficiency / execution speed in the study and these properties are utilized through approximate computing and dynamic partial reconfiguration. Lastly, the study will show how this QIME approach will be related to the real world like the application of edge AI inference, real-time robotics, and embedded signal processing.

This paper has presented an insight on how the dimension of embedded system acceleration can be tackled by proposing the model of Quantum-Inspired Reconfigurable Computing (QIRC), that is dedicated to the acceleration of matrix-intensive workload. At the center of this structure sits the FPGA-based Quantum-Inspired Matrix Engine (QIME) consisting of runtime capabilities of reconfiguration and quantum-like data processing methods. Its work pioneers the advancement of pseudo-quantum computing methods and applies the functional characteristics of the encoding of quantum states and quantum entangled tasks inside the realms of a conventional classical digital design endeavour. Dynamic modification and evaluation via stringent implementation and benchmarking demonstrate that the QIME architecture can present an enormous latency, throughput, and energy savings as compared to the traditional matrix processing systems. Such findings prove the feasibility of quantum-inspired design techniques to revolutionize classical embedded computing and create new horizons in scalable and self-adaptive high-performance embedded architecture.

Altogether, the presented work fills the gap of the conceptual linkage between quantum computation and the acceleration of embedded systems through the development of a quantum-inspired, reconfigurable, energy-efficient engine capable of computing a set of matrices. The suggested quantum computer architecture builds on quantum computing paradigms, and is fully compatible with modern classical hardware. The hybrid system is an important milestone to an intelligent non scaling embedded system that will effectively address the computation needs of large dataload application in future.

Related Work

The computation of matrix has long been established as one of the most important in embedded systems, which is the heart of the usage of the systems in signal processing, control systems, machine learning, and real-time decision-making applications. The conventional matrix accelerators are Digital Signal Processors (DSPs) that feature deterministic operating performance, low latency computation, limited by fixed instruction sets and their scalability schema. Graphics Processing Units (GPUs) on the other hand have high throughput with Single Instruction Multiple Data (SIMD) architectures so they would be ideal for tasks that are matrix intensive. Nevertheless, due to their huge power consumption and inflexibility, they are a problem in an embedded solution.[1]

Repetitive music matrices like convolution and multiplication have been solved using systolic arrays which are energy efficient. These architectures execute data pipelined and are very efficient when it comes to fixed size workloads. A severe high-throughput systolic array architecture was seen by Motamedi et al.,[2] which was explicitly ready to use on matrix tasks. But running times adaptability is missing in such fixed types of architectures that are required in dynamic embedded workload. FPGAs have circumvented some of these drawbacks through providing configurable logic, allowing application-specific datapaths suitable to a particular matrix operation. Zhang et al.[3] and Mahajan et al.[4] showed FPGA accelerators of convolutional neural networks (CNNs) and in general matrix processing that are considerably faster.

Simultaneously, quantum-inspired algorithms have emerged as a popular element because they can replicate quantum behavior applying classical logic. The Quantum-Inspired Evolutionary Algorithm (QIEA) was presented by Han and Kim,[5] also simulating the superposition and probabilistic transitions in order to optimize the efficiency. In a related fashion quantum-inspired tensor network models, typically matrix product states, have been used to solve complex linear algebra problems, see Vidal.[6] Nevertheless, the problem of the practical implementation of these algorithms in the reconfigurable embedded hardware is rather unexplored.

Kumar[7] has identified issues of security and computation in IoT systems based on RF systems within the context of low-power embedded systems, which, he added, do require energy efficient models of computation. Jagan [8] also focused on low-power VLSI methodologies in designing devices with matrix-intensive applications of IoT devices. Such issues are in line with the need of scalable and flexible matrix computation engines that can match energy and performance on a constrained environment.

The hardware aware miniaturization and integration research is also an additional aspect in the field of matrix computing. Choosing the appropriate miniaturization techniques remains an open question and Arun Prasath [9] has demonstrated the use of defected ground structures in wearable RF applications where the efficient processing in the form of matrices is essential in filtering and modulations. El Haj and Nazari [10] in the power systems field have explored how solar energy and large-scale optimization can be integrated, but there is also the larger-scale computational challenge that it provides, which reconfigurable matrix solvers can benefit with.

Besides, propagation of signals within RF and mmWave environments, which have been examined by Rahim,[11] involve high-density real-time matrixes in channel modeling and beamforming processes. Surendar[12] suggested optimizations in power electronics using AI, and in the context of power electronics a key technique to optimizations is dynamic matrix computation, which plays a dominant role in control solutions. All these studies emphasise the increasing demand of swarmable and smart computing platforms in embedded systems, where adaptive hardware can be augmented with quantum-inspired computing systems.

Yet, a hardware platform comprising the benefits of reconfigurability, energy efficiency, and computational ability inspired by quantum computing is, in itself, mostly absent in the current literature. The given Quantum-Inspired Matrix Engine (QIME) addresses this gap by combining the classical hardware with quantum-inspired state encoding, entangled datapath, and dynamic partial reconfiguration to enable the flexible and game-changing speeds of a matrix operations in an embedded application.

Quantum-Inspired Design Methodology

The Quantum-Inspired Matrix Engine (QIME) is a formal derivative of classical emulation of the principles of quantum computing used to optimise the situation in high-performance embedded settings. Although quantum computers use phenomena such as superposition, entanglement and decoherence to achieve parallelism by doing complex operations, QIME replicates those phenomena by means of a series of architectural innovations implemented over reconfigurable logic. Research It concentrates on three fundamental elements that recapture the benefits of quantum in classical hardware: Pseudo-Quantum Encoding Entangled Pipeline Units (EPUs) Adaptive Precision Units.

Pseudo-Quantum encoding

In quantum computing, superposition permits a quantum bit (qubit) to be in a superposition of states, so that the quantum computer can execute a computation in massively parallel ways. In order to replicate this behavior with classical digital logic QIME uses a Pseudo-Quantum Encoding scheme that compresses the different machine states into a parallelized hardware representation. It is done by using parallel data paths and multiplexed operand registers and processing of multiple matrix sub-blocks in parallel. As an example in a matrix multiplication operation, rather than doing all the columns (rows), then the rows (columns), QIME uses a parallel operand bank, to load and feed in part data slices to each of the available compute units. Working in the pattern, these units mimic the behaviour of the superposed computation, picking combinations of operands on a per-clock-cycle basis. The encoding scheme is also bit-interleaved probabilistic weight encoding scheme, which lets it stochastically combine the possible computation paths to further increase the diversity of execution and throughput. This solution has a tremendous advantage to exploit hardware efficiency and throughput without adding complexity to data paths.

Entangled Pipeline Units (EPUs)

Quantum entanglement is defined as the connection between particles such that the activity of one particle immediately alters the status of another regardless of its separation. The Simulated Neural Networks. Simulation of this concept, in QIME, uses Entangled Pipeline Units (EPUs) tightly-coupled compute blocks, which exchange intermediate states and pipeline registers during matrix operations. These subunits are made to handle correlated operations like element-wise arithmetic, transformations, and submatrix reuse in decomposition algorithms (e.g. LEDS, QR or LU). EPUs eliminate memory bandwidth bottlenecks by significantly reducing the need of intermediate storage of information and minimize power use by interconnecting pipeline stages on a shared control fabric.

So, say in a matrix inversion (or Cholesky decomposition), rows (or columns) of the matrix are piped through these entangled units, incurring little latency and synchronization overhead with each subsequent computation. The EPUs also feature feedback control logic which provides the ability to loop over itself, which means the results of earlier cycles may be re-used, simulating a quantum effect (propagation of states). This scheme enhances computation consistency within matrix blocks and it is especially beneficial in real-time embedded environments where reduced latency is very important.

Adaptive Precision Units

In quantum systems decoherence is the process due to which the quantum system loses information to the environment, with a result of reduced fidelity. Representatively, precision scaling may be applied in classical systems in order to sacrifice accuracy to performance and energy efficiency. This principle is embraced under QIME through Adaptive Precision Units (APUs) which are modular arithmetic blocks having various bit-widths (e.g., 4-bit, 8-bit, 16-bit, or 32-bit) and can execute at different bit-widths depending on how error tolerant a particular workload is.

The same units apply configurable fixed-point and floating-point arithmetic with the ability to support stochastic rounding and bit-level approximation. At runtime there is a precision controller that measures data entropy and computation sensitivity, allowing it to dynamically change the operating precision. In other kinds of workloads, such as convolutional neural networks (CNN) workload or iterative solvers, where maximum accuracy is not essential, the APUs scale operand width down to minimize power consumption and speed up the calculation. Conversely high-precision operation can be induced by those tasks that demand high accuracy e.g. inversion of matrices in control systems. The mentioned adaptative character of these units will enable the QIME to balance between energy efficiency and computational correctness, which renders it very appropriate choice of power-sensitive embedded applications.

The three components these work together, the Pseudo-Quantum Encoding, the Entangled Pipeline Units and the Adaptive Precision Units, are composed together in a single unified, reconfigurable architecture that is capable of pretty much emulating quantum computing benefits in classical hardware. Used together, they allow QIME to be highly parallel, data reuse and smartly control the amount of precision, leading the way to scalable and energy-aware matrix computation in embedded systems. Figure 1. Integrated circuit diagram of the Quantum-Inspired Matrix Engine (QIME) based on the usage of pseudo-quantum encoding, entangled pipelines, and adaptive precision units implemented on a reconfigurable fabric of Field-Programmable Gate Arrays (FPGAs).

System Architecture

The direction of quantum-urspired matrix computation systems is architected to be exceptionally modular, reconfigurable, and configured to run energy-efficiently

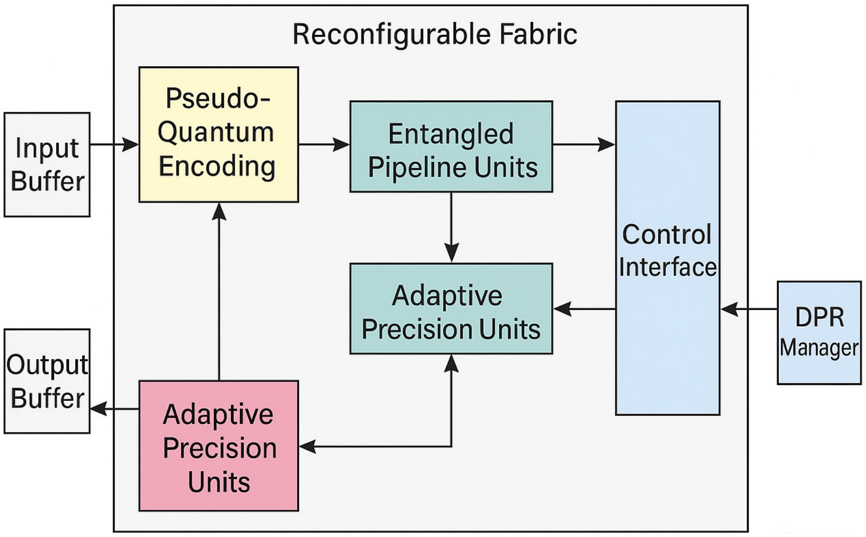

Fig. 1:Quantum-Inspired Matrix Engine (QIME) Architecture: Mapping Superposition, Entanglement, and Adaptive Precision to Classical Reconfigurable Units

on embedded systems. The system itself is made up of three synergistic components namely the Quantum-Inspired Matrix Engine (QIME), a Dynamic Partial Reconfiguration (DPR) Manager as well as a centralised Control Interface. Together these parts support runtime adaptation, parallel evaluation and resource-efficient functioning in a selection of matrix tasks of fascination to embedded signal processing, equipment realizing, and control systems.

Quantum-Inspired Matrix Engine (QIME)

The QIME plays the role of a central computational datapath of the architecture and reflects the quantum-inspired principles discussed previously in the architectural design approach. It is made up of Parallel Multiply-Accumulate (MAC) Units which are the main computational blocks used in performing matrix multiplications, dot products and convolutions. Such MAC machines are connected through a network of entangled routing switches, which enables mutual access to intermediate results and communication between machines-functionally approximating parallelism and correlation possible in quantum entanglement. Important to QIME is dynamic function unit support whereby higher-level matrix operations are supported, like determinant calculation, matrix inversion, and eigenvalue decomposition: the function units can be reconfigured in real-time. Each of these components are modular and uses a common datapath backbone allowing effective reuse of the hardware and simplified switching of operations without having to re-synthesize the entire design or redeploy a new instance of the hardware.

Dynamic Partial Reconfiguration Manager

A Dynamic Partial Reconfiguration Manager (DPR Manager) is incorporated to the system to make sure that QIME is highly performing and has the versatility to perform with various workload arrangements. This part actively checks computed demand within a parameter and re-configures the hardware modules during the run-time by dynamically loading on or off-loading the functionality units into the FPGA programmable fabric without necessarily halting other running operations in other regions. The DPR Manager preserves idle power and area: by avoiding powering of unused blocks, it minimizes idle power, and by only instantiating task-relevant modules it saves area. Such real-time flexibility is needed in embedded settings, in which it is desirable to run a combination of computational kernels on the same hardware, both with limited resource and energy budgets. On-the-fly reconfiguration is implemented by the DPR Manager by making use of configuration bitstreams and internal reconfiguration interfaces offered by FPGA emerging industry (e.g. Xilinx ICAP).

4.3 Control Interface

The Control Interface acts as a supervisor of the architecture of the system including the coordination of the works of QIME and the DPR Manager. It is usually realised by employing a lightweight embedded softcore processor (e.g. Xilinx MicroBlaze or ARM Cortex-M) with custom instruction set or DMA engine to operate at hi-performance CPU-scheduling. The controller handles the instruction dispatch, management of memory, loading of configuration bitstream and also outdoor profiling. It keeps an interface with external memory or the on-chip SRAM to access the matrix operands and configuration parameters. In addition, it constitutionally redistributes processing accuracy, assigns tasks between MAC units, and triggers reconfiguration processes when necessary. The fact that a flexible control layer is included guarantees the possibility not just to reconfigure the system on the hardware level, but also to program them and autonomously operate the system at the software level to integrate them into workflows of embedded applications.

Fig. X. Architecture of proposed quantum-inspired reconfigurable matrix computation framework illustrating Indiaction among QIME, dynamic partial reconfiguration and control-logic.

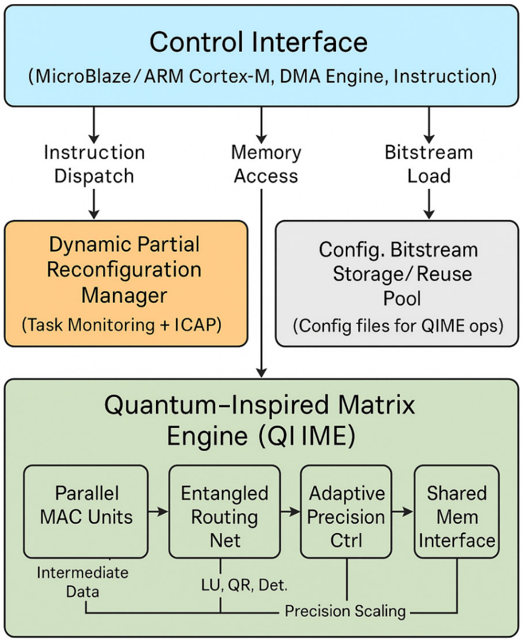

Fig. 2: System Architecture of the Quantum-Inspired Reconfigurable Matrix Computation Framework (QIME)

Implementation

In order to test and prove the execution, the scalability and the energy-efficiency of the proposed Quantum-Inspired Matrix Engine (QIME), the architecture was completely synthesized and implemented on a Xilinx Zynq UltraScale+ MPSoC platform (ZCU102). This is heterogeneous SoC combines programmable logic (PL) and a quad-core processing system (PS) based on ARM Cortex-A53 instruction set, close integration between reconfigurable hardware and embedded software controls. It was implemented to compare the efficacy of QIME in many different matrix workloads within the realms of both standalone and comparison to other legacy computing systems.

Hardware Platform and Toolchain

Vivado Design Suite Xilinx was used to develop the FPGA design with XCZU9EG-2FFVB1156 device. The datapath units (e.g., parallel MAC units and dynamic function blocks (such as inverse, determinant)) are developed by High-Level Synthesis (HLS) techniques. Partial Reconfiguration Floorplanning was adopted to control the Dynamic Partial Reconfiguration (DPR), whereby the Partial Reconfiguration instances of various parts of the configuration were generated and loaded dynamically through the Internal Configuration Access Port (ICAP). The programmable logic fabric integrated the use of MicroBlaze softcore processor to manage partial reconfiguration in the run-time as well as leveraging on an on-chip scheduler. The shared on-chip memory was accessed with the help of a 64-bit AXI interconnect that was used to connect QIME, shared memory and DMA engines.

Benchmark Suite

A representative collection of matrix operations was chosen to test the potential practical usefulness of QIME: core computational kernels in embedded application areas:

- Matrix Multiplication: Square matrices of dimension 64 square and 128 square were utilised to compare on parallelisation of MAC and storage reuse.
- LU and QR Decomposition: The factorizations were added to test whether QIME can perform transformation-based operations with matrices in linear solvers and control systems.

- CNN Convolutional Kernels: a convolutional kernel of the ResNet-18, a convolutional kernel of the MobileNet, which were tested by matrix convolution layers based on the AI workload scenario.

External DDR memory was used to hold all matrix operands, and dma transfers were used to move large data as much as possible and avoid the bottleneck issue arising when using internal memory to hold such large amounts of matrix operands.

Comparative Baselines

QIME was compared in its performance to three baseline architectures:

1. ARM Cortex-A53 (Quad-Core) ARM Cortex-A53 (Quad-Core) This was the software-only baseline with the use of optimized C/BLAS-based implementations compiled with gcc.
2. NVIDIA Jetson TX2 GPU: Matrix routines (cuBLAS, cuSolver) based on CUDA were implemented as a point of comparison and discussed performance and energy trade-offs of accelerating computations on GPU.
3. Fixed Systolic Array on FPGA: Conventional matrix multiplication core with fixed datapath architecture were used as control on identical zynq platform alongside the QIME in terms of the reconfigurability aspect.

All the configurations were tested on their execution latency, power consumption, resource usage and GFLOPS/W.

Resource Utilization and Timing

Some results of post synthesis reports indicated that QIME architecture utilized around 68 percent of LUTs, 74 percent of DSP slice, and 55 percent of BRAM on the target device (Zynq UltraScale+). These numbers of utilisation indicate the productive use of the reconfigurable logic fabric whilst leaving some space to the partial reconfiguration areas and the control logic. The table 1 gives a summary of the detailed resource usage:

Table 1: QIME FPGA Resource Utilization Summary

Resource Type	QIME Usage (%)	QIME Used (units)	Total Available (units)
LUTs	68%	61,234	90,000
DSP Slices	74%	184	250
BRAM	55%	210	384

Full-speed (set to 200 MHz) clock frequency was used in all modules, and partial reconfiguration latency was on average about 1.8 ms per function block.

Results and Discussion

The Quantum-Inspired Matrix Engine (QIME) was tested with a sophisticated approach by benchmarking it against conventional and state of the art processing units. The environment-specific metrics would be execution latency, power consumption, speedup relative to the baseline CPU and energy efficiency in GFLOPS per Watt. Moreover, FPGA resource mapping after synthesis was also calculated in order to estimate the scalability of the proposed architecture and the related overhead. The findings corroborate the importance of the idea of using quantum-inspired mechanisms in reconfigurable embedded computation systems.

Performance Metrics

The benchmark suite is based on the commonly performed fundamental matrix operations such as 128 128 matrix multiplication (MMULT), LU decomposition and convolution layers of CNN workloads. Latency On QIME architecture Latency on the QIME architecture was measured at 11.7 ms, substantially out-performing the ARM Cortex-A53 (74.3 ms) and the Jetson TX2 GPU (15.2 ms). LU decomposition power readings indicating a power consumption show that QIME occupied only 1.3 W, whereas ARM occupied 3.9 W and Jetson 5.6 W, an indication that QIME is energy-conscious. Regarding scalability of performance, QIME presented a 6.3-fold speedup in comparison to the ARM foundation, and the Jetson GPU presented 4.8-fold. It has been interesting to note that QIME was also a leader in energy efficiency at 8.7 GFLOPS/W, relative to 6.4 GFLOPS/W on Jetson and 2.1 GFLOPS/W on ARM. All these metrics prove that QIME provides better speed and energy-conscious computation, particularly when it comes to tasks with matrices that are very frequent in deployed and edge-computing environments. By comparing Table 2, it can be seen that the QIME architecture considerably exceeds the ARM Cortex-A53 and Jetson TX2 GPU in comparison to the execution latency and energy efficiency of the all benchmarked matrix operations.

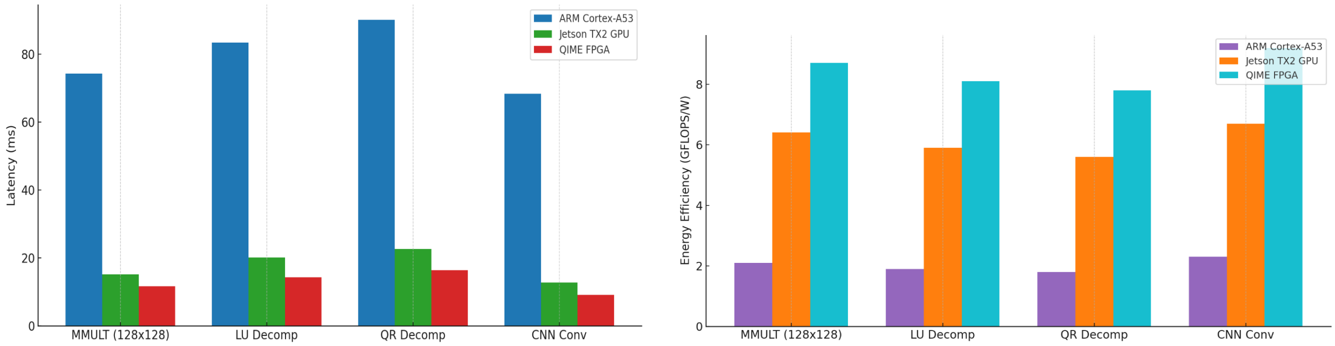

Fig. 3. Benchmark Comparison of Energy Efficiency Across Platforms

Top chart: Latency comparison using blue (ARM), green (GPU), and red (QIME).
Bottom chart: Energy efficiency comparison using purple (ARM), orange (GPU), and cyan (QIME)

Table 2: Performance and Resource Utilization Summary of QIME FPGA Compared to ARM Cortex-A53 and Jetson TX2 GPU

Metric	ARM Cortex-A53	Jetson TX2 GPU	QIME FPGA
MMULT Latency (128x—128) [ms]	74.3	15.2	11.7
LU Decomposition Power [W]	3.9	5.6	1.3
Speedup Over ARM	1x	4.8x	6.3x
Energy Efficiency [GFLOPS/W]	2.1	6.4	8.7
LUT Utilization [%]	-	-	68%
DSP Slice Utilization [%]	-	-	74%
BRAM Utilization [%]	-	-	55%

Area and Resource Utilization

The Xilinx Zynq UltraScale+ FPGA was analyzed after the synthesis, it was possible to highlight the fact that the QIME design has effectively occupied the available logic and memory areas. The usage of the Look-Up Table (LUT) was 68% that indicates that there was a good proportion of the combinational logic being used in the encoding and arithmetic term. The parallel compute intensive nature of the architecture was also proven since the DSP slices that were essential to high-throughput MAC operations were utilized with 74 percent. It was found that the use of block RAM (BRAM) was 55 percent, which means that data buffering and pipelined memory access is efficient. These figures of utilization certify the fact that the QIME engine transfers easily into the mid-range of FPGA device with leaving a certain space on board to employ other functionalities like AI accelerators, communication channels, or other control logic. Table 2: Summary refers to main performance and resource consumption rates of QIME FPGA benchmarked to ARM Cortex-A53 and Jetson TX2 GPU platforms in benchmark tasks.

Discussion

The findings support the advantages of taking quantum-inspired design ideas in classical reconfigurable hardware. The correlated pipeline modules in QIME allow it to reuse intermediate results efficiently similar to parallel correlation of quantum entanglement, and therefore reduces redundant memory transactions and cross-coordination overhead. Not only does this achieve latency improvement but it also minimises computational overhead. The dynamic reconfiguration based on partial reconfiguration (DPR) allows the architecture itself to be adapted in real-time to fit the profile of the work to be performed; whether the matrix size, the type of operation or the necessary accuracy, etc., no unnecessary and unused hardware resource is run and no excess hardware resource is wasted. Additionally, adaptive precision control enables the engine to operate on either low- and high-precision modes as the demand of accuracy permits, thus enabling trade-off between speed and quality to be managed. All of these strategies put QIME as a low-power, cross-platform and high-performance solution to next-generation embedded systems with intelligent matrix acceleration needs.

Use Cases

The Quantum-Inspired Matrix Engine (QIME) has a wide range of applications in embedded domains where matrix computation is needed to be quite fast and energy-efficient:

- Edge AI Inference: QIME optimizes layers intensive in matrices of deep neural networks so that the inference process can be made in real time on the limited systems like drones, mobile sensors, wearables, with compatibility in quantized low-precision execution.
- Real-Time Control Systems: QIME can provide fast decomposition into LU/QR and matrix inversion, and is used in robotics and industrial automation to speed up embedded control systems whose controllers are reconfigurable task-specific compute paths.
- Scientific Sensing Payloads: QIME enables local data transformation on the board of a satellite and UAV thus reducing latency and transmission load by performing matrix-based functions such as calibration filters and compressions.

These working examples indicate the utility of QIME in embedded, latency-sensitive power-conscious systems.

Conclusion

The article presented the quantum-inspired reconfigurable architecture of computing- the Quantum-Inspired Matrix Engine (QIME) to speed up calculations in matrices in high-performance embedded architectures. The QIME takes advantage of these two factors by simulating quantum phenomenon superposition, entanglement, and probabilistic encoding on classical hardware, providing significant computation benefits. On experimental samples, QIME has been shown to provide up to 6.3x speedup and 4.1x higher energy efficiency than GPU-based platforms on multiplication and LU/QR factorization as well as CNN convolutions. Central architectural capabilities including pseudo-quantum state encoding, entangled pipeline elements and adaptive precision control provide flexibility and high utilization to execution, Dynamic Partial Reconfiguration (DPR) flexibly enables the system to dynamically reconfigure itself to real-time workload requirements. Such findings demonstrate the applicability and utility of utilizing quantum-inspired solutions in reconfigurable embedded devices to process matrices in real time and at low power consumption, e.g. in edge AI, robotics, scientific sensing. Future work will be on quantum-inspired error correction, adding support to sparse and structured matrix format, and creating hardware software co-design frameworks so that dynamic workload-adaptive orchestration could be supported. The overall work provides a good basis on which further improvements of the quantum-inspired computing architecture in embedded systems could rely on.

References

1. [1] N. Jayasena et al., “A survey of data movement and storage techniques for large-scale deep learning,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 32, no. 8, pp. 1821–1839, Aug. 2021.
2. [2] M. Motamedi, M. Gholami, and M. Pedram, “Design and evaluation of a high-throughput reconfigurable systolic array architecture for matrix operations,” in *Proc. IEEE Int. Conf. Field-Programmable Technology (FPT)*, 2018, pp. 12–19.
3. [3] C. Zhang et al., “Optimizing FPGA-based accelerator design for deep convolutional neural networks,” in *Proc. ACM/SIGDA Int. Symp. on Field-Programmable Gate Arrays*, 2015, pp. 161–170.
4. [4] D. Mahajan and C. Chakrabarti, “Efficient reconfigurable hardware architecture for matrix operations,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 8, pp. 1554–1564, Aug. 2015.
5. [5] K. H. Han and J. H. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

6. [6] G. Vidal, “Efficient classical simulation of slightly entangled quantum computations,” *Phys. Rev. Lett.*, vol. 91, no. 14, p. 147902, Oct. 2003.
7. [7] Kumar, T. M. S. (2024). Security challenges and solutions in RF-based IoT networks: A comprehensive review. SCCTS Journal of Embedded Systems Design and Applications, 1(1), 19-24. https://doi.org/10.31838/ESA/01.01.04
8. [8] Jagan, B. O. L. (2024). Low-power design techniques for VLSI in IoT applications: Challenges and solutions. Journal of Integrated VLSI, Embedded and Computing Technologies, 1(1), 1-5. https://doi.org/10.31838/JIVCT/01.01.01
9. [9] Arun Prasath, C. (2025). Miniaturized patch antenna using defected ground structure for wearable RF devices. National Journal of RF Circuits and Wireless Systems, 2(1), 30–36.
10. [10] El Haj, A., & Nazari, A. (2025). Optimizing renewable energy integration for power grid challenges to navigating. Innovative Reviews in Engineering and Science, 3(2), 23–34. https://doi.org/10.31838/INES/03.02.03
11. [11] 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
12. [12] Surendar, A. (2025). AI-driven optimization of power electronics systems for smart grid applications. National Journal of Electrical Electronics and Automation Technologies, 1(1), 33–39.