RESEARCH ARTICLE

ECEJOURNALS.IN

Scalable Reconfigurable Architectures for Quantum-Inspired Computing: Design, Challenges, and Opportunities

Y. Charabi[1]*, Beh L. Wei[2]

[1]College of Applied Science, University of Technology and Applied Sciences, Ibri, Sultanate of Oman
[2]Faculty of Information Science and Technology University, Kebangsaan, Malaysia

Keywords:
Quantum-Inspired Computing,
Scalable Architecture,
Reconfigurable Hardware,
FPGA, CGRA,
Hardware-Software Co-Design,
Parallelism, Optimization

Author's Email:
charbi.y@gmail.com,
beh.lee@ftsm.ukm.my

DOI : 10.31838/RCC/03.02.02

Received : 13.11.2025

Revised : 18.01.2026

Accepted : 20.04.2026

Abstract

This paper focuses on exploring the scalable reconfigurable hardware for quantum-inspired computing (QIC), a computing paradigm which simulates quantum properties on classical hardware to solve complex problems related to optimization, simulation and machine learning. This mainly aims at reducing the gap between the complexity of the computations required by QIC algorithms and the processing power of standard hardware through the use of reprogrammable systems like the Field Programmable Gate Arrays (FPGAs) or Coarse-Grained Reconfigurable Arrays (CGRAs). The subject of methodology is the architecture study, design-space exploration and determining the parameters which are thought to be critical such as parallelism, pipelining performance and hardware-software co-design co-integration. The principles of modularity, optimizing data locality and partial runtime reconfiguration are focused so that to increase the scalability and flexibilities. The issue of resource constraints, routing congestion, synchronization delay and energy efficiency is well addressed. Experimental findings show that the flexibility of reconfigurable logic leads to the utilization of the workloads of QIC, especially when working with iterative and matrix-related computing. The boost in performance is further enhanced by use of approximate computing, in-memory operations and dynamic reconfiguration techniques. The paper passes the conclusion that scalable reconfigurable architecture is an option to speed quantum-inspired algorithms and make practical usage in the area of edge computing, embedded computing, and high-performance computing. The opportunities in the future will be described in the paper such as combination of quantum and classical with hybrid integration and neuromorphic acceleration in order to improve the performance and scalability of the systems.

How to cite this article: Charabi Y, Wei BL (2026). Scalable Reconfigurable Architectures for Quantum-Inspired Computing: Design, Challenges, and Opportunities. SCCTS Transactions on Reconfigurable Computing, Vol. 3, No. 2, 2026, 11-20

Introduction

Quantum-inspired computing (QIC) has recently gained popular attention because it promises to use the concepts of quantum physics that lead to solutions to classically intractable problems (i.e., superposition, entanglement, and quantum tunneling), but without the cost of full-quantum hardware. Quantum-inspired optimisation algorithms (QICs) like quantum-inspired optimization techniques (e.g. QUBO solvers), tensor network contraction and variational approaches have been increasingly promising in applications as diverse as combinatorial optimisation, machine learning, and physical simulations.

The use of a QIC algorithm, though powerful, may cause efficiency issues when implemented on a traditional and fixed-functio processor, specifically, with regard to scalability, flexibility, and power requirements. The reason is that QIC algorithms generally require large memory bandwidth, parallelism, and dot-matrix looping, inefficient with the traditional architecture. A convincing alternative is to use reconfigurable computing platforms that satisfy those requirements inherent in their adaptability, fine-grained parallelism, and on-the-fly architectural optimization. Nevertheless the current studies where a particular attention is paid to the static hardware acceleration or rather the algorithmic efficiency is typically driven by a single perspective, without a unified picture of what a scalable, reconfigurable architecture can look like and serve the changing needs of QIC out in the field.

The current paper fills it with an in-depth examination of scalable reconfigurable architecture in QIC workloads. It describes design principles, issues, and optimization techniques with an understanding of exciting new avenues of opportunities in hybrid quantum-classical processing.

Such platforms need to be elaborated in the recent works as the enabling technology to bridge the gap between near term quantum algorithms and practical deployment platforms, by highlighting the potential of reconfigurable logic as a bridge to be built between the two.[1]

Related Work

Recent advances in quantum-inspired computing (QIC) have led to the development of a broad variety of algorithmic strategies, which exploit the concepts of quantum mechanics, through quantum annealing, superposition, and entanglement-based logic, to describe the explanation of the hard problems using classical hardware. In practice, methods such as Quadratic Unconstrained Binary Optimization (QUBO) and Ising models have been run on GPUs and application-specific integrated circuits (ASICs) to mitigate the bottlenecks of computation in fields, including machine learning and logistics.[1, 2] Whereas fixed-function accelerators are ideal in sustained throughput in certain limited applications, they are unable to shift and respond to the new and dynamic nature of QIC workloads. These accelerators may have the limitation of predetermined data paths, lesser memory hierarchy control and lack compatibility to algorithmic reconfiguration.

What has researchers turning to reconfigurable computing platforms is to deal with this. FPGA and CGRAs are modular and scalable by nature and therefore can run QIC algorithms in real-time on these platforms or tailor hardware to the algorithms. As an example, Chatterjee et al.[3] proposed a QIC accelerator on FPGA optimized toward QUBO issues. It was capable of large gains in speed but proved only modestly scalable because it was ill-adapted to reuse resources and built on assumptions that spatial areas could be fixed. With the FPGAs being central in quantum simulations, Wang et al.[4] emphasized on the FPGAs contraction in tensor networks. The advantages of parallelism were demonstrated in their implementation but not with the inclusion of the support of runtime partial reconfiguration and the dynamic workload management.

Furthermore, some mapping attempts of variational quantum eigensolvers (VQEs) and other hybrid algorithms onto reconfigurable hardware [5] hit the problems of routing congestion, inadequacy of tools, and complexity. Such attempts sometimes to fall short of generalisation to additional QIC problem classes, or make use of the complete gamut of hardware-software co-design freedom.

The present issues are:

- There is no cohesive architectural design plan that is scalable and reusable as applicable across QIC applications
- Poor support to partial reconfiguration resulting in inadequate usage of FPGA logic resources
- Poor design space exploration tool to map quantum- motivated workflows onto diverse reconfigurable substrates

This paper fills in these gaps by combining a modular architectural skeleton, a pipelined data flow, as well as the ability to dynamically reconfigure. It complements other activities and presents specific directions to scalable and energy-efficient reconfigurable hardware toward general-purpose QIC.

Quantum-Inspired Algorithms and Their Hardware Demands

Quantum-inspired computing (QIC) is a family of algorithms that simulate a certain aspect of quantum mechanics on classical computers, including superposition, correlations inspired by entanglement, and tunneling mechanics. Such algorithms have potential performance benefits in solving high-dimensional optimization and simulation problems avoiding physical limits of existing quantum technologies.

Overview of Quantum-Inspired Algorithms

Quantum annealing, quantum approximate optimization algorithm (QAOA)-inspired routines and tensor network methods are among the best-known QIC algorithms:

- Quantum Annealing Quantum Annealing concentrates on solving complex energy landscapes by emulating the adiabatic transition of a quantum system. These dynamics have been known to be approximated by classical implementations, including simulated annealing with quantum tunneling heuristics.
- QAOA-Inspired Routines approximate variations in quantum gate-model systems. These routines are parameterized classical circuits, over cost and mixer gates, and over which one can use either gradient-based or evolutionary optimization.
- Tensor Networks (e.g. matrix product states, tree tensor networks) are frequently used to simulate entangled quantum systems (e.g. quantum spin chains). They are very dependent on operations of linear algebra including tensor contraction, decomposition and reshaping, and as such, they are easy to accelerate in a computationally intensive way.

Computational Patterns and Resource Utilization

QIC algorithms are data and iterative incorporate computational patterns. Among the key features is:

- Sparse and dense matrix: common in QUBO solvers and in tensor contraction.
- Repetitive control-flow but data dependency: examples are variational algorithms and annealing schedules.
- Large memory bandwidth requirements: particularly large scale simulation and gradient based optimization.
- Fine-grained parallelism: in tensor contractions, and implicitly vectored operations.

These trends cause non-homogenous usage of resources with some parts of the computation being essential memory-bound and some parts enjoying high compute density. Also the algorithmic structure is normally parametric and adjustable such that resource allocation is able to be customized on the fly.

Suitability for Parallel and Pipelined Hardware

QIC algorithms can be efficiently mapped to reconfigurable hardware: it naturally lends itself to parallel and modular iteration:

- Enjoyable parallelism can come at the level of concurrent independence between of computation (e.g. parallel tensor contraction, concurrent QAOA evaluations).
- Pipelining Pipelining can be used when the computation contains multiple stages, like when updating layers in annealing, or repeated matrix multiplications in tensor networks, and can be more efficient reducing throughput and latency.
- Deep performance of FPGAs and CGRAs: hardware-loop unrolling and spatial replication of compute units allow deep performance optimization.

These characteristics render QIC algorithms eligible to hardware acceleration, especially those based on modular (including dynamically reconfigurable) energy aware execution models. Each type of QIC algorithm, as seen in Figure 1, will have different computational

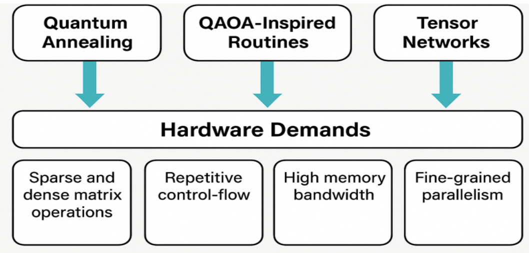

Fig. 1: Quantum-Inspired Algorithms and Their Hardware Demands

properties, including, but not limited to, iteration schemes, matrix-intensive activity, and others, which will translate almost directly into hardware requirements such as parallelism, pipelining, memory bandwidth, and resource utilization. Insight to these mappings is critical to the development of scalable reconfigurable architectures suitable to quantum-inspired workloads.

Reconfigurable Computing Platforms for QIC

The potential of reconfigurable computing platform to be specifically employed to perform quantum-inspired computing (QIC) algorithm acceleration lies in the nature of flexibility and adaptability of the platform. These architectures allow hardware design to be tailored at the architecture level in order to enable specific needs of the QIC workloads, including high parallelism, iterative execution and dynamic dataflow patterns. The paper investigates two most important reconfigurable paradigms namely, Field Programmable Gate Arrays (FPGAs) and Coarse-Grained Reconfigurable Arrays (CGRAs), and related reconfiguration methods support and development frameworks.

Field Programmable Gate Arrays (FPGAs)

FPGAs have been popular in QIC due to their small-grained configurability, and support of application-specific hardware datapaths. They also comprise configurable logic blocks (CLBs), programmable routing, as well as on-chip memory capabilities, which makes them very suitable in the performance of operations, especially pipelined execution of matrices multiplications, tensor contractions and QUBO evaluations.

QIC algorithms tend to be facilitated by an FPGA in terms of:

- Execution paths of low latency
- Bit level precision control
- Dynamic reprogramming of repeated optimization steps

But disadvantages are the routing bottlenecks, timing closure problems, and the lack of portability of a design to cross-architecture designs. However, their adaptability in the level of instructions makes them quantum-inspired workloads matchless in flexibility.

Coarse-Grained Reconfigurable Arrays (CGRAs)

CGrAs are intermediate reconfigurability of FPGAs and fixed-function ASICs features. They are built using an array of programmable processing elements (PEs) tied together by a configurable network in a fashion optimized to word-level processing, not gate-level logic.

CGRAs have the following benefits to QIC:

- Facilitate specialized data-path of tensor and linear algebra computations
- Parallellize the algorithmic block level
- Be more energy efficient than the FPGAs on structured work loads

Each of the FPGAs and CGRAs have its own unique advantages to quantum-inspired computing workload accelerations. Although FPGAs provide fine-grain configurability and are quite suitable to perform dynamic and adaptive algorithms, CGRAs are more efficient at executing regular, highly-parallel QIC routines. Table 1 provides the comparative features of these platforms and highlights the key trade-offs in architectural and performance attributes that may affect the appropriateness of these platforms to different QIC deployment implementations.

Runtime Reconfiguration and Partial Reconfiguration

The major benefit of reconfigurable platforms is the fact that they offer the capability to support runtime adaptation. Partial reconfiguration (PR) helps to update part of the hardware leaving the rest functional. It is particularly practical in the case of QIC when the algorithm stages (e.g., mixing and cost in QAOA) could be loaded and unloaded dynamically according to the workload phase.

Advantages of runtime reconfiguration are:

- Higher resources utilization
- Dynamic work loads adaptation
- Energy saving

Even through the above, PR has overheads as a result of bitstream management, reconfiguration time and complexity because of their designs.

High-Level Synthesis (HLS) and Hardware Description Languages (HDLs)

FPGAs and CGRAs have historically been programmed using hardware description languages (HDLs), most commonly Verilog, or VHDL that provide detailed access to architecture structure, but can be time-consuming to implement and usually demand expert skills. High-Level Synthesis (HLS) has given quantum-inspired computing (QIC) researchers and system designers the opportunity to describe complex algorithms in a high-level language (e.g., C/C++ or OpenCL) which automatically translates such to a synthesizable representation (RTL).

There are some benefits that come with HLS in the design of QIC systems:

- Faster prototyping and fast explorations of design space,
- Automated optimization, e.g. pipelining and loop unrolling,
- Improved portability of heterogeneous device platforms.

Nonetheless, even in the face of such benefits, the existing HLS tools continue to suffer limitations with regards to maximizing control of fine grained resource and low level parallelism which usually results in non optimal performance on reconfigurable platforms. To address this a hybrid co-design method is commonly used where the most demanding parts of the compute are accomplished manually in HDL to be highly efficient, whereas the control logic and orchestration at the system level is accomplished in HLS.

This development flow is shown in Figure 2: HDLHLS Hybrid Co-Design Workflow for Reconfigurable Architectures that shows how the automation provided by HLS can be combined with the low-level HDL refinement to be able to provide high performance and scalable QIC deployment.

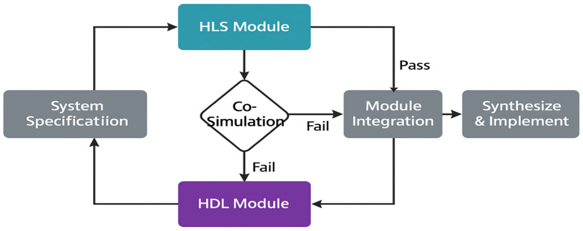

Fig. 2: HDL-HLS Hybrid Co-Design Workflow for Reconfigurable Architectures

Table 1. Comparison of FPGAs and CGRAs for Quantum-Inspired Computing (QIC)

Aspect	Field Programmable Gate Arrays (FPGAs)	Coarse-Grained Reconfigurable Arrays (CGRAs)
Granularity	Fine-grained (bit-level logic and routing)	Coarse-grained (word-level programmable processing elements)
Reconfigurability	High (full and partial reconfiguration supported)	Moderate (functional unit and interconnect-level reconfiguration)
Parallelism Support	Excellent (custom pipelining, deep loop unrolling)	Good (block-level parallelism, moderate control flow flexibility)
Resource Efficiency	Moderate-Low (may waste resources due to granularity)	High (better area and power efficiency for structured computations)
Performance on QIC Workloads	High for irregular, iterative workloads like QAOA, simulated annealing	High for structured workloads like tensor networks, matrix operations
Energy Efficiency	Lower (due to fine-grained switching activity)	Higher (optimized data path and compute unit utilization)
Development Complexity	High (requires expertise in HDL or HLS for low-level design)	Moderate (simpler programming models, better compiler support)
Runtime Reconfiguration	Supported (especially with partial reconfiguration techniques)	Limited (typically not used for dynamic phase-switching)
Toolchain Maturity	Mature (broad support from vendors like Xilinx, Intel)	Evolving (growing support in academic and custom SoC toolchains)
Suitability for QIC Algorithms	Best for dynamic, adaptive, or mixed-precision QIC routines	Best for structured, parallelizable QIC algorithms

As a recap, QIC requires architectural flexibility on computing platforms, which is found in reconfigurable computing platforms. There is a necessity to use a hybrid approach to an effective utilization of the computational capabilities of quantum inspired algorithms based on utilizing FPGA, CGRAs, and runtime reconfiguration and a mixed approach comprised of HLS and HDL. These two blocks, when taken together, can be seen in figure 3 as providing architectural flexibility, enabling effective and scalable hardware implementation to support a wide variety of QIC implementations.

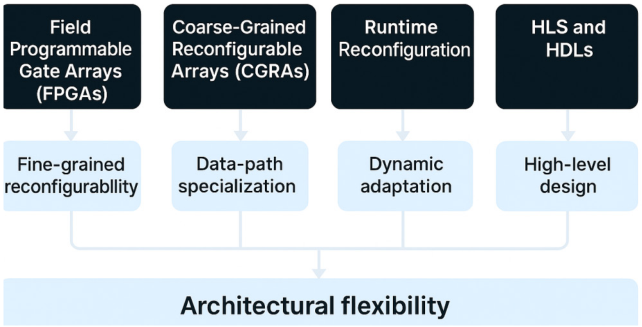
Field Programmable Gate Arrays (FPGAs)
Coarse-Grained Reconfigurable Arrays (CGRAs)
Runtime Reconfiguration
HLS and HDLs
Fine-grained reconfigurabllity
Data-path specialization
Dynamic adaptation
High-level design
Architectural flexibility

Fig. 3: Reconfigurable Computing for Quantum-Inspired Computing (QIC)

Design Principles for Scalable Architectures

Size-scale reconfigurable quantum-informed computing (QIC) needs to be structured to enable management of complicated, information-extreme workloads with favorableness and energy effectiveness. The major design principles are given below.

Modular Processing elements (PEs) and Scalable Interconnects:

The modular PEs allow parallelism and replication and they facilitate the core QIC actions, which include the contraction of tensors and the updating of matrices. Scalable interconnects (e.g. mesh, bus) provide low latency communications and wide throughput among computational units.

Tiling, Memory Hierarchy and Data Locality:

It is imperative that the memory is used efficiently because QIC algorithms are memory-bound. Strategies to increase data locality, reduce the number of external accesses, and to achieve bandwidth effectiveness by the use of tiling principles, hierarchical memory architectures (registers, scratchpads, DRAM).

Iterative Computation specialization: Pipeline-Friendly Design:

QIC workloads are commonly iterative algorithms. Deep pipelining, loop unrolling, and operation chain enables it to maintain throughput. The usage and effectiveness are enhanced through the use of instruction- and task-level parallelism.

Clock Domain Management, and synchronization:

Large scale architectures need accurate synchronization. Clock domain crossing (CDC) multi-clock domain designs and GALS ensure data coherence, and safe inter-domain communication.

Collectively, these architectural techniques will guarantee that reconfigurable platforms can be scaled up efficiently and beyond, and can accommodate dynamic as well as data-intensive tendencies of quantum-inspired workloads. Figure 4 presents these scalable principles on design, and the various contributions of modular compute units, optimized memory hierarchy, pipelined execution and synchronized clock domains to flexible and high-performance QIC implementations on hardware.

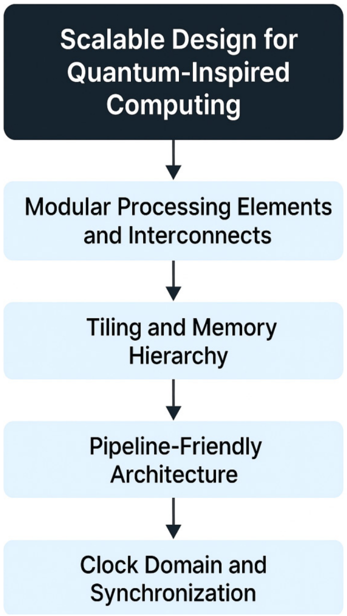
Scalable Design for Quantum-Inspired Computing
Modular Processing Elements and Interconnects
Tiling and Memory Hierarchy
Pipeline-Friendly Architecture
Clock Domain and Synchronization

Fig. 4: Scalable Design Principles for Quantum-Inspired Computing Architectures

Challenges in Implementation

Although there will be reconfigurable platforms of quantum-inspired computing (QIC), this progress is prohibited by the range of challenges against meaningful practical usage.

Hardware Resource constraints:

QIC algorithms consume a lot of logic, memory and routing resources. FPGAs commonly suffer gridlocks in routing and a scarcity of on-chip memory, such that the FPGAs have restrictions in performance because of overly much off-chip information passing.

Communication Overheads; Synchronization Overheads:

Stage-to-stage or clock domain-wide inter-PE communication and synchronization imposes stage-to-stage communication latency and bandwidth occupation that has become serious in parallel and highly pipelined architectures.

Issues of Power and Thermal:

Although more energy-efficient than the general-purpose processors, the reconfigurable devices are prone to hotspots and dynamic power-hungry that is most apparent under intensive / fine-grained workloads without appropriate management schemes.

See Programming and Toolchain Complexity

Creation of QIC accelerators brings rapid HDL/HLS learning curves, lack of debugging facilities, and lack of standardization, increasing portability and efficiency of the design.

They are complex problems that need to be solved holistically, using a combination of architectural co-design activity, compiler optimizations and thermal/ power-aware scheduling to enable scalable and efficient use of QIC deployment. All these threats with related architectural level and software level mitigation techniques and countermeasures have been compiled in Table 2 that can be taken as a guideline when it comes to designing efficient and reliable QIC hardware platforms.

Optimization Strategies

The optimization of the reconfigurable architectures in quantum-inspired computing (QIC) needs a multi-level solution that considers trade-offs between system performance; flexibility and energy efficiency.

Hardware–Software Co-Design:

Flexibleness can be achieved when incorporating software control and hardware acceleration. Hardware computing (e.g., QAOA scheduling) can be software controlled with compute-intensive workloads assigned to reconfigurable hardware, increasing flexibility and performance.

Approximate Computing:

Low-precision arithmetic or error-tolerant logic can be used selectively in non-critical paths to minimize resource usage and power consumption at the cost of little accuracy (example applications would be early

Table 2: Challenges in Reconfigurable QIC Implementation and Mitigation Strategies

Challenge	Description	Mitigation Strategies
Hardware Resource Constraints	Limited logic, memory, and routing capacity lead to congestion and fragmentation	Resource- aware partitioning, memory tiling, partial reconfiguration, floorplanning
Communication and Synchronization Overhead	Latency and contention from inter-PE data exchange and multi-clock domains	Hierarchical interconnects, clock domain crossing (CDC) techniques, asynchronous protocols
Power Consumption and Thermal Constraints	High switching activity causes localized heating and energy inefficiency	DVFS, thermal-aware scheduling, workload balancing, GALS architecture
Programming and Toolchain Complexity	Complex HDL/HLS workflows and non-unified tool support	HLS-accelerated design, IP reuse, hardware abstraction layers, open-source compiler support

stages of optimization, where a non-critical iteration is not performed with high-precision arithmetic), or to implement cost/power/area tradeoffs.

(Dynamic Voltage and Frequency Scaling - DVFS):

DVFS varies the system workload in accordance with the degree of workload. By reducing voltage/ frequency when idling or performing lightweight operations the energy is saved, whereas in peak phases the performance can be increased by raising the voltage.

Machine Learning for Design Space Exploration.

ML models can be used to automate tuning of architecture to find the best possible design parameters based on a given performance or energy target (e.g., when a desired count of PE, memory depth needs to be found). This speeds up the process and facilitates thoughtful decisions in design.

These strategies in combination increase the scalability, adaptability and efficiency of QIC reconfigurable platforms. With the help of hardware-software co-design, approximation methods, adaptivity at runtime, and design optimization using ML, the systems will be able to satisfy different requirements of quantum-inspired workloads. These optimization strategies are summarized in figure 5 which show how the strategies can increase the performance and energy efficiency of reconfigurable QIC architectures as well.

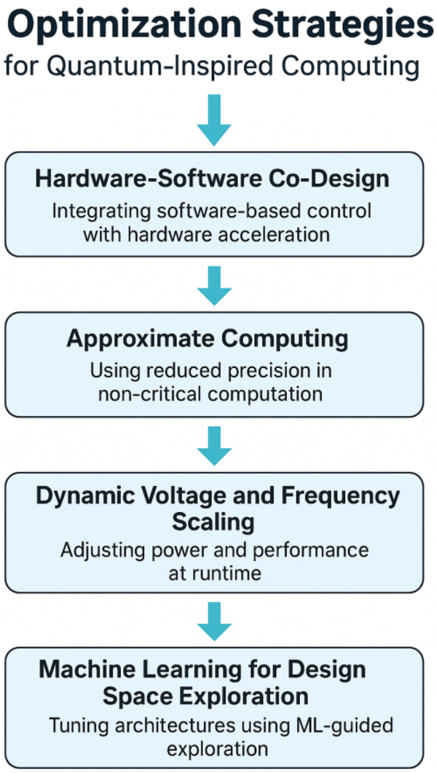
Optimization Strategies
for Quantum-Inspired Computing
Hardware-Software Co-Design
Integrating software-based control with hardware acceleration
Approximate Computing
Using reduced precision in non-critical computation
Dynamic Voltage and Frequency Scaling
Adjusting power and performance at runtime
Machine Learning for Design Space Exploration
Tuning architectures using ML-guided exploration

Fig. 5: Optimization Strategies for Quantum-Inspired Computing Architectures

Future Opportunities

With the evolution of quantum-inspired computing (QIC) it is clear that emerging technologies and architectural advancements can provide considerable growth in the longevity, accessibility and scalability of reconfigurable QIC systems. The following section presents some of the most significant future developments with transformative potentials.

Integration with Quantum Co-Processors

A potentially interesting hybrid scheme is to use classical reconfigurable hardware with quantum processing units (QPUs). With that type of set up, QIC algorithms may be divisible between a classical accelerator (e.g., FPGA or CGRA) for deterministic workloads, and a quantum co-processor to handle entanglement or probabilistic inference. Real-time hybrid execution can be supported, and data transfer overhead reduced, when the interconnects are low-latency and well-integrated via shared memory. This co-design strategy opens the door to (near-term) quantum advantage through quantumclassical heterogeneity.

Neuromorphic and In-Memory Computing Paradigms

The recently developed classes of hardware look promising, including neuromorphic computing and processing-in-memory (PIM) architectures or other display structured devices. Event-driven parallelism and low-power design in the neuromorphic chips make it suitable to parallel and iterative type of operations in QIC including optimization and annealing. Likewise, in-memory computing reduces data transfer by overlapping logic with memory arrays, and is advantageous to tensor-intensive QIC routines. Embedding these types of paradigms with reconfigurable logic may open up further efficiency gains in computation.

Cloud-Based FPGA Acceleration

FPGAs can be accessed in cloud platforms, e.g., Amazon EC2 F1, Microsoft Azure, or Xilinx Alveo, and accelerators QIC can be deployed scalably and on demand. This democratizes access to the reconfigurable infrastructure to both research and industry application scenarios, enables parameter sweeps at large scale as well as real-time inference at scale. Deployment and experimentation can further be simplified by integration with containerized development environments and orchestration frameworks (e.g. Kubernetes).

Standardization and Open-Source IP Libraries

The absence of common IP-cores, toolchains and software-frames is a major drawback towards the popularity of QIC hardware solutions. The next steps will be facilitated with the help of open-source projects that will offer hardware blocks that can be used multiple times and have been proven by third parties (e.g., QUBO solvers, tensor engines). Toolchains and interoperability will be facilitated by standardized APIs and allow the custom design time-to-market to decrease across the platforms. The ecosystems of collaborations with QIC IP developments would push innovation and accessibility.

These future possibilities when combined show a cloud-native heteronymous and standardized quantum-inspired computing (QIC) ecosystem- an ecosystem that can make use of the hybrid architecture, the next-generation memory models, and the open innovation, which can lead to real scalable energy-efficient acceleration of QIC applications in the real world.

Conclusion

The prospect of quantum-inspired computing (QIC) is presented by the scalable reconfigurable architectures that allow flexibility and high performance execution of quantum-based algorithms. With the potential of FPGA and CGRA platform, with the assist of runtime reconfiguration, high-level synthesis and modular design, high-computational workloads such as optimization, simulation and variational applications can be accomplished in a more energy-efficient and adaptive manner using QIC systems.

The article has examined some of the major architectural features and optimization tactics as well as algorithm-hardware mapping algorithms that are needed to bring the concept of QIC in a feasible form. It has also pointed out the challenges of implementation including resource limitation, routing and congestion and complexity of programming and how these challenges can be addressed through co-design, approximate computing, and dynamic resource management. The potential of heterogeneous and standardized reconfigurable platforms in HPC illustrates further opportunities in future, due to the possibility of extending them to quantum co-processors, neuromorphic modules, cloud-based acceleration, and more.

Future interdisciplinary investigations at the interface of quantum-inspired algorithm design, programmable hardware breakthrough, and automation tool chains will be essential in realizing the potential of the future intelligent computing platforms optimized to QIC.

Acknowledgments

The authors appreciate the contribution of their institutes, colleagues, research and funding bodies that permited them to make this work successful. We would like to offer a special gratitude to technical advisors and infrastructure teams who provided information and resource during the development and evaluation process of the study.

References

1. Chatterjee, A., Chen, R., & Kaul, H. (2024). A quantum-inspired optimization accelerator using reconfigurable architectures. *IEEE Transactions on Computers, 73*(2), 305–318. https://doi.org/10.1109/TC.2023.3284150
2. Neukart, F., Compostella, G., Seidel, C., von Dollen, D., Yarkoni, S., & Parney, B. (2017). Traffic flow optimization using a quantum annealer. *Frontiers in ICT, 4*, 29–35. https://doi.org/10.3389/fict.2017.00029
3. Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). Parameterized quantum circuits as machine learning models. *Quantum Science and Technology, 4*(4), 043001. https://doi.org/10.1088/2058-9565/ab4eb5
4. Wang, Y., Tan, M., & Chen, Y. (2023). High-performance tensor network contractions on FPGAs. In *Proceedings of*

the IEEE International Conference on Field-Programmable Logic and Applications (FPL) (pp. 1–4). IEEE. https://doi.org/10.1109/FPL58074.2023.10223126

5. Tannu, S. S., & Qureshi, M. K. (2019). Mitigating measurement errors in quantum computers by exploiting state-dependent bias. *ACM SIGARCH Computer Architecture News, 47*(2), 297–310. https://doi.org/10.1145/3307650.3322253
6. Marinescu, A. (2024). Cultural Narratives and Their Influence on Tourist Satisfaction in Historical Destinations. Journal of Tourism, Culture, and Management Studies, 1(1), 44-54.
7. Dumitrescu, M., & Stanescu, I. (2024). The role of leadership in developing: Innovation and Quality Culture. National Journal of Quality, Innovation, and Business Excellence, 1(1), 37-46.
8. Kowalski, T., & Nowak, M. (2024). The Impact of Digital Transformation on Quality Assurance in Healthcare Systems. National Journal of Quality, Innovation, and Business Excellence, 1(2), 1-12.
9. Poornimadarshini, S. (2025). Mathematical modeling of rotor dynamics in high-speed electric motors for aerospace applications. Journal of Applied Mathematical Models in Engineering, 1(1), 33–43.