SCCTS Transactions on Reconfigurable Computing, ISSN: 3049-1533 Vol. 2, No. 3, 2025 (pp. 18-26)
RESEARCH ARTICLE ECEJOURNALS.IN

# Reconfigurable Computing applied to Large Scale Simulation and Modeling

**Ekaterina Ivanovna Kozlova[1], Nikolai Viktorovich Smirnov[2]***

[1,2]Faculty of Innovative Technologies, Tomsk State University, 634050 Tomsk, Russia

**Keywords:**
Computational Efficiency;
Large Scale Simulation;
Modeling Techniques;
Parallel Computing;
Reconfigurable Hardware;
Simulation Optimization

**Corresponding Author Email:**
smirnov.vik.tor@mail.tsu.ru

**DOI:** 10.31838/RCC/02.03.03

**Received :** 19.06.25
**Revised :** 19.09.25
**Accepted :** 27.10.25

## Abstract

Over the past decade, reconfigurable computing has found its way as a powerful paradigm for accelerating complex computational tasks in diverse domains. Due to growing simulation and modeling loads in both scale and complexity, reconfigurable architectures must be accordingly adapted to meet the demand for efficient execution of these workloads. In this work, we elaborate on the challenges and potentials of employing reconfigurable computing (RC) for large scale simulation and modeling, using key technology advances, architectural questions as well as prominent application areas to make our point. Reconfigurable hardware (e.g. field programmable gate array), in conjunction with high performance computing (HPC) infrastructures, has enabled new approaches to approximately solve computationally intensive simulations and models. Researchers and engineers can potentially obtain dramatic speedups and energy efficiency as reconfigurable platforms offer the flexibility and parallelism necessary to leverage their power. Yet restructuring of reconfigurable computing for large scale simulation and modeling is not free of difficulties. In order to fully realize the potential of this approach there are issues to be addressed including programming complexity, scalability, and integration with existing software ecosystems. In this article we explore such challenges towards reconfigurable computing for simulation and modeling applications and explore brewing in what's possible with reconfigurable computing.

**How to cite this article:** Kozlova EI, Smirnov NV (2025). Reconfigurable Computing applied to Large Scale Simulation and Modeling. SCCTS Transactions on Reconfigurable Computing, Vol. 2, No. 3, 2025, 18-26

## Introduction

In this exploration we will see the transformative potential of reconfigurable computing on many computing related fields such as climate modeling, molecular dynamics, risk analysis and more when used on real world problems. Readers will be exposed to the current state of the art and future directions of this very rapidly evolving field with the simple task of understanding how reconfigurable computing is changing the landscape of large scale simulation and modeling.[1-4]

## Reconfigurable Computing Elementary Topics

Computer architecture paradigm shift reconfigurable computing provides a middle ground between the flexibility of general purpose processors and the efficiency of application specific integrated circuits (ASICs). Locally reconfigurable computing is predicated on the ability to dynamically reconfigure hardware to implement suitable computational structures for a particular task (Figure 1).[5-6]

## Field Programmable Gate Arrays (FPGAs)

FPGA (Field Programmable Gate Array) is the most common platform for reconfigurable computing. An FPGAs consists of an array of programmable logic blocks interconnected by a reconfigurable routing fabric. These devices can be programmed to implement complex digital circuits, and because these devices implement digital circuits, they can be used to implement custom hardware accelerators of

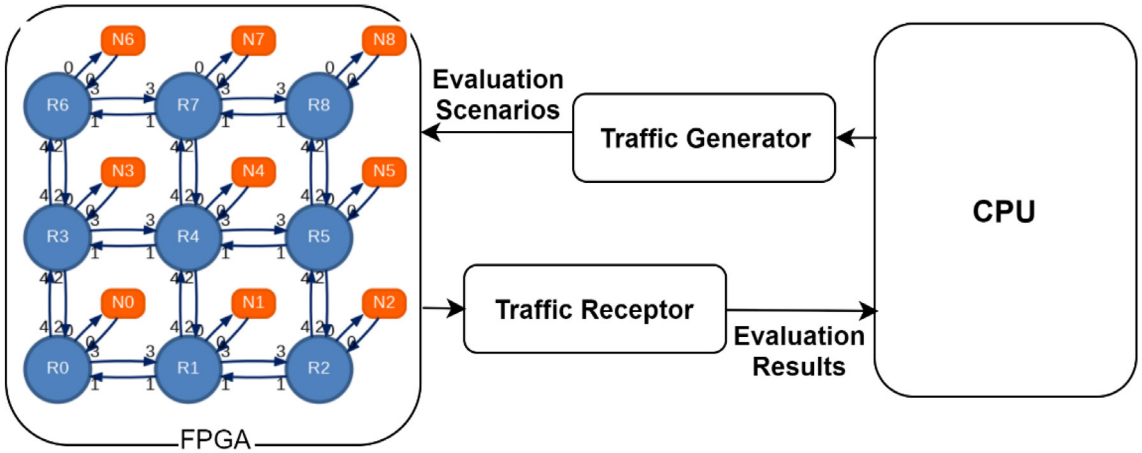


**Fig. 1: Reconfigurable Computing Elementary Topics**

custom configured hardware for various algorithms or applications.[7-8]

**FPGAs offer several advantages for large-scale simulation and modeling tasks:**

**Parallelism:** Its ability to perform multiple processing elements concurrently.

**Customization:** Mapping hardware architecture to suit the algorithm specific needs.

**Adaptability:** Support of changing the hardware dynamically for different computational phases/ processes

**Energy efficiency:** Harnessing the hardware to a single task leads to much power savings than normal general purpose processors.

## High-Level Synthesis (HLS)

Programming an FPGA was traditionally a hardware description language (HDL) affair, so solving for quadratic speed depends on a bunch of expertise in VHDL or Verilog. But, with the development of high level synthesis (HLS) tools, reconfigurable computing has made the 'write once, run everywhere' paradigm available to software programmers. It enables designers to specify algorithms in high level languages like C or C++, and to automatically generate its corresponding hardware description (Table 1).

He asserts that HLS tools have significantly reduced the development time and complexity in designing FPGA based designs in order to obtain faster iteration and explore different architectural options. It has been especially useful for exploiting reconfigurable computing for large scale simulation and modeling tasks where very quick prototyping and optimization of hardware accelerators is the key.[9-11]

**Table 1: Reconfigurable Computing for Large Scale Simulation**

| Feature | Explanation |
|---|---|
| Parallel Processing | Parallel processing enables simultaneous execution of tasks, speeding up simulations by distributing workloads across multiple cores or processors. |
| Adaptability | Adaptability allows the system to reconfigure its hardware resources on-the-fly, tailoring the computing power to specific tasks within the simulation. |
| Real-Time Execution | Real-time execution supports immediate processing of data, making it suitable for applications where speed and quick decision-making are critical. |
| Scalability | Scalability ensures that the system can handle growing datasets and more complex models as the scope of simulations expands. |
| Optimization | Optimization improves the overall performance of simulations by utilizing efficient hardware configurations and computational strategies. |
| Resource Efficiency | Resource efficiency minimizes power consumption and maximizes the use of available computing resources, making it cost-effective for large-scale simulations. |

## Reconfigurable Computing Architectures

Several architectural approaches have emerged for integrating reconfigurable computing into high-performance computing systems:

**FPGA-based accelerator cards:** Standalone FPGA boards, usually connected via PCIe interfaces, that form part of existing computer systems.

**Hybrid CPU-FPGA platforms:** Systems integrated closely on the same chip or package that include traditional processors and reconfigurable fabric.

**FPGA-based supercomputers:** Large scale systems constructed fully from interconnected nodes of FPGA.

**Cloud-based FPGA resources:** Cloud service providers provide virtualized FPGA instances, on demand access to reconfigurable computing resources.

They differ from each other in terms of performance and scalability but with different trade offs in ease of integration with an existing software ecosystem. Architecture is chosen based on the specific requirements of any given simulation or modeling task.[12-14]

## Applications to Reconfigurable Hardware

To fully exploit the capabilities of reconfigurable computing for massive simulation and modeling algorithm specifics need to adapt to the advantages of FPGA platform. However, this is typically a matter of rethinking computational approaches to maximize parallelism and minimize data movement.

### Parallelization Strategies

FPGAs are one of the prime advantages in terms of massively parallel architectures. When adapting algorithms for reconfigurable hardware, designers must identify opportunities for parallelism at various levels:

**Data-level parallelism:** Doing vector or SIMD like operations on multiple data elements in a single pass.

**Task-level parallelism:** Converting independent computational tasks into parallel execution of many complex pipelines.

**Pipeline parallelism:** Pipelines of deep processing with different stages of comutation overlapping.

The combination of these approaches is often what forms effective parallelization strategies, specifically targeting the algorithm characteristics and available FPGA resources.[15-18]

### Memory Hierarchy Optimization

The performance of FPGA based designs is heavily influenced by memory access patterns and data locality. Adapting algorithms for reconfigurable hardware often requires careful consideration of the memory hierarchy (Table 2):

**On-chip memory:** For often used data and intermediate results, using FPGA block RAM (BRAM) resources.

**External memory:** Access pattern optimization to off chip DRAM to minimize latency and maximize bandwidth utilization by employing appropriate access patterns.

**Caching strategies:** Exploiting data that is reused and local in cache

Through the use of memory hierarchy, designers can minimize data movement and maximize the computational efficiency of their FPGA based accelerators.[19-23]

## Fixed-Point Arithmetic

However, floating point arithmetic in many large scale simulation and modeling tasks can be resource intensive on FPGAs. Adapting algorithms to use fixed-point arithmetic can lead to significant improvements in performance and resource utilization:

**Precision analysis:** Finding out just how much precision is needed for each computational stage (Figure 2).

**Table 2: Reconfigurable Computing in Large Scale Simulation and Modeling**

| Application | Benefit |
|---|---|
| Climate Modeling | Reconfigurable computing accelerates climate simulations by handling complex, data-heavy models and performing iterative calculations efficiently. |
| Fluid Dynamics Simulation | In fluid dynamics simulation, reconfigurable computing allows for fast execution of highly parallel computations, improving the speed and accuracy of simulations. |
| Material Science | In material science, it enables simulations of molecular interactions and properties at large scales, helping to develop new materials more efficiently. |
| AI Training | Reconfigurable systems improve AI training models by handling large datasets and iterative processes, allowing faster convergence and more complex models. |
| Energy System Simulation | Energy system simulations benefit from the reconfiguration capabilities, enabling the analysis of large energy grids and optimization for efficiency. |
| Autonomous Vehicle Modeling | Autonomous vehicle modeling leverages reconfigurable computing to simulate vehicle behavior, traffic conditions, and sensor data processing in real-time for enhanced testing. |

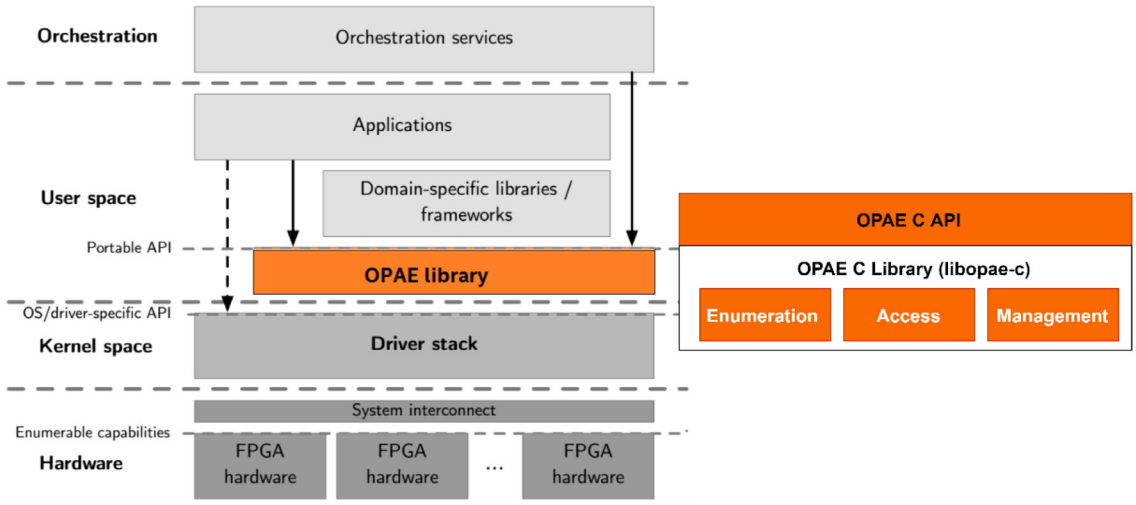


**Fig. 2. Fixed-Point Arithmetic**

**Dynamic range optimization:** As described by Chicken, we allow the fixed point representation to accommodate the full range of expected values.

## Reconfigurable Computing: Programming Models

Since large-scale simulation and modeling application starts to migrate to reconfigurable computing platforms, we face an imperative of building programming modelling that can bridge the gap between software development and hardware design. Several approaches have emerged to address this challenge:

## OpenCL for FPGAs

For FPGA based systems, another adoption of the Open Computing Language (OpenCL) framework was designed for heterogeneous computing with GPUs. OpenCL for FPGAs allows developers to write portable, high-level code that can be compiled to hardware descriptions:

**Kernel-based programming:** Mapping computations from the express as parallel kernels to FPGA resources

**Host-device model:** Getting the control flow (CPU) running and the computational kernels (FPGA) to execute for separating the two.

**Memory model:** Abstracting different kinds of memory such as global, local and private memory

Due to its portability and its familiarity to the software developer's OpenCL has become popular in reconfigurable computing community. Nevertheless, the development of hardware on so fine and arcane a structure as an FPGA commonly necessitates intrinsics (i.e., cooptimizations) in addition to basic coding techniques, and careful thought for the underlying hardware architecture.

## DSLs: Domain Specific Languages

Domain-specific languages tailored for particular classes of simulation and modeling problems have emerged as a powerful approach for programming reconfigurable systems:

Abstraction of hardware details: Enabling domain experts to describe algorithms in terms they are naturally familiar with and notation.

Automated optimization: To generate efficient FPGA implementation, we introduce the incorporation of domain knowledge.

Rapid prototyping: it is important for allowing a quick exploration of diverse algorithmic and architectural options.

Among other DSLs for reconfigurable computing are Spatial for parallel patterns, Halide for image processing pipelines, and Delite for machine learning applications. The aim of these languages is to try to preserve a balance between productivity and performance so that reconfigurable computing can be more accessible to domain experts.

## HLS Extensions

Building upon the success of HLS tools, researchers and industry practitioners have developed various extensions and frameworks to enhance the capabilities of C/C++-based hardware design. Pragma-based directives: Enable guidance by source code annotations to the HLS process for developers.

**Custom libraries:** This supplies optimized implementations of common functions and data structures for FPGA targets.

**Design space exploration tools:** Converting the process of exploring different architectural options and optimizations to automation.

These extensions aim to give finer grained control for generating hardware, whilst predicting benefits from high level programming languages.[24-25]

## Reconfigurable Systems: Scaling for Large Scale Simulations

Simulation and modeling tasks continue to scale in number and complexity, and scalability becomes a more pressing need for both scaling up reconfigurable computing architectures to handle massive datasets and complex computational workflows. Several approaches have emerged to address the challenges of scaling reconfigurable systems:

### Multi-FPGA Systems

Connecting multiple FPGAs to form larger, more powerful computing systems has become a popular approach for tackling large-scale simulations:

Inter-FPGA communication: Protocols and interfaces for data exchange between FPGAs brought to an efficient state

Workload partitioning: Exploiting parallelism in large simulations; dividing across FPGAs (Figure 3)

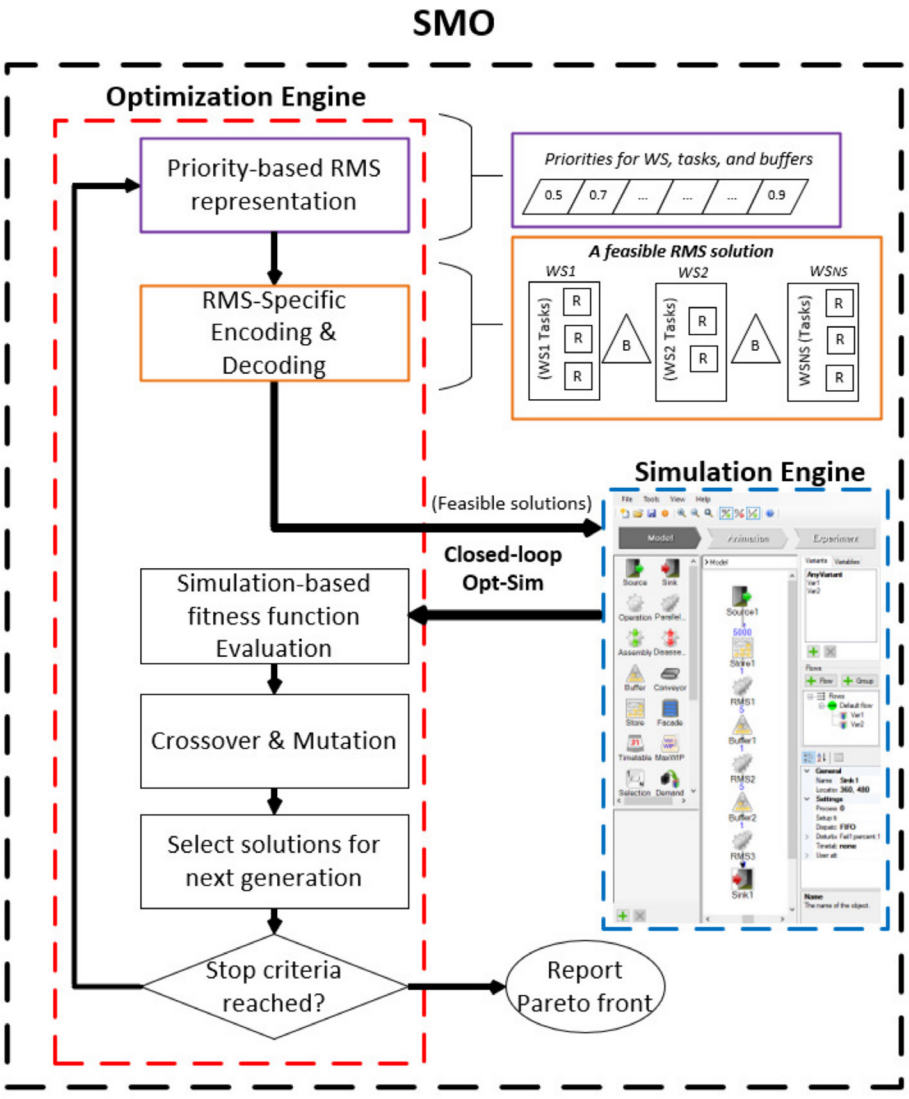


**Fig. 3: Reconfigurable Systems: Scaling for Large Scale Simulations**

**Resource sharing:** Addressing wisely how FPGAs should share memory and other resources.

However, these systems can present an attractive performance and scalability, sacrificing all of this for a poorly designed system that is hard to program in and to synchronize.

## FPGA-Based Supercomputers

Several research initiatives and commercial projects have explored the concept of building supercomputers entirely from FPGA nodes:

**Scalable interconnect architectures:** FPGAs used for exposure exercises to develop the desired high bandwidth, low latency interconnect between FPGAs, when configured with large numbers, times out.

**System software and runtime environments:** Middware and operating system creation for FPGA based super computers

**Programming models for massive parallelism:** To express and manage computations across thousands of FPGA nodes developing frameworks

For some classes of simulation and modeling problems, FPGA based supercomputers provide a path to unprecedented performance and energy efficiency. But they also involve substantial hardware infrastructure investments and software ecosystem deployments.

## Cloud-Based FPGA Resources

Cloud service providers have begun offering virtualized FPGA instances, allowing users to access reconfigurable computing resources on-demand:

**Elasticity and scalability:** FPGA resource allocation based on workload requirement dynamically.

**Cost-effectiveness:** Allowing high performance reconfigurable hardware accessible on the fly without large initial investments.

**Integration with cloud ecosystems:** Using available cloud services for storing, managing, and analyzing data

With the rise of cloud platforms for reconfigurable computing, organizations without their own FPGA infrastructure can scale reconfigurable computing flexibly and cost effectively for large-scale simulation.[26-27]

## Application Areas in Large Scale Simulation and Modeling

In numerous domains of large scale simulation and modeling, reconfigurable computing has been shown to enhance time and space reconfigurability to accelerate complex computational tasks and to facilitate scientific breakthroughs. Some key application areas include:

## Climate and Weather Modeling

Climate and weather simulations are computationally intensive tasks that can benefit significantly from the parallelism and customization offered by reconfigurable computing:

**Atmospheric dynamics:** Computing atmospheric circulation models' accelerated fluid dynamics calculation

**Ocean modeling:** Efficient solver for ocean current and temperature simulations

**Data assimilation:** Accelerating the way observations can be integrated into climate models

Accelerators based on FPGA have much success to improve the performance and energy efficiency of climate modeling workloads, making it possible to perform higher resolution simulations, and more accurate future forecasts.

Molecular Dynamics Simulations of Lithium Intercalation into a novel Solvent.

Molecular dynamics simulations, which model the behavior of atoms and molecules over time, are another area where reconfigurable computing has made significant inroads:

**Force field calculations:** Efficiently computation of interatomic forces via implementation of custom hardware

**Particle integration:** It accelerates the numerical integration of particle trajectory.

**Long-range interactions:** Specialized architectures for electrostatic and van der Waal's forces

By utilizing the convenience and parallelism of FPGAs, researchers have been able to greatly speedup molecular dynamics simulations, to last longer, and/or simulate larger systems.

## Financial Risk Analysis

The finance industry has embraced reconfigurable computing for accelerating complex risk analysis and modeling tasks:

Monte Carlo simulations: Parallel random number generator and statistical analysis pipelines

Options pricing: A study of ways of developing custom hardware for fast evaluation of financial derivatives.

**High-frequency trading:** Building low latency architectures with timely market analysis and decision making.

The use of FPGA based solutions has proven the potential to reduce time and energy spent on large scale financial simulations thus allowing a more extensive risk assessment and quicker decision making.

## Machine Learning, Artificial Intelligence

As AI and machine learning models continue to grow in size and complexity, reconfigurable computing offers a promising approach for accelerating both training and inference:

Neural network acceleration: Fast matrix multiplication with custom architectures

Sparse matrix operations: Specializing in the hardware handling sparse data structures ubiquitous within many of the AI models.

**Quantization and reduced precision:** Using flexibility of FPGAs to fit efficient low precision arithmetic

Edge computing and IoT applications powered by Reconfigurable computing platforms promise improved performance and energy efficiency while being adaptive to the power constraints.

## Future Directions and The Challenges

While reconfigurable computing has made significant strides in adapting to large-scale simulation and modeling tasks, several challenges remain to be addressed:

Despite advancements in high-level synthesis and domain-specific languages, programming reconfigurable systems remains more complex than traditional software development:

**Abstraction vs. performance:** Our goal is to balance the need for high level abstractions while allowing fine tuning of the hardware implementations.

**Debugging and verification:** Tools and methodologies for the efficient debugging and verification of FPGA based designs.

**Portability:** It then discusses the creation of programming models that target various FPGA architectures and vendors while maintaining performance.

This will be crucial in which will help with broader adoption of reconfigurable computing into the simulation and modeling community.

## Integration With Existing Software Ecosystems

Extensive software ecosystems and libraries are needed for many large scale simulation and modeling workflows. Integrating reconfigurable computing solutions into these existing frameworks presents several challenges:

**API compatibility:** Achieving interfaces that enable FPGA accelerated components to seamlessly share with existing software libraries.

**Data format standardization:** How to enable efficient data exchange between a CPU based and FPGA based component of a simulation

**Legacy code support:** We focus on the creation of tools and methodologies for speeding up legacy simulation codes on reconfigurable platforms.

In order for reconfigurable computing platforms to be more widely used in scientific and engineering domains, the interoperability between these platforms and existing software ecosystems will need to be improved.

## Solver and System Level

As reconfigurable computing systems scale to tackle larger simulation and modeling problems, new challenges emerge at the system level:

**Power and cooling:** Consecrating ourselves in developing efficient power delivery and thermal management solutions for large scale FPGA based systems

**Fault tolerance:** Mechanisms for reconfigurable systems detecting and mitigating hardware faults
Resource management: Intelligent scheduling and allocation algorithms for shared FPGA resurces in multi user environments

These system level issues are necessary to be addressed in order to build and operate large scale reconfigurable computing infrastructures for simulation and modeling applications.

## Architectures and Emerging Technologies

The field of reconfigurable computing continues to evolve, with new technologies and architectural approaches on the horizon:

3D-stacked FPGAs: Vertically Stacked FPGA dies for increased logic density and reduced interconnect delays - High power consumption

Exploring potential of vertically stacked FPGA dies for increased logic density and reduce interconnect delays

**In-memory computing:** Reconfigurable logic integration with emerging memory technologies for massively data driven computing architectures

**Quantum-classical hybrid systems:** Study of the use of reconfigurable computing for interfacing with and complements quantum processors

Such emerging technologies and architectures may enable us to use reconfigurable computing to meet new demands for large scale simulation and modeling.

## Conclusion

Reconfigurable computing appears to be well suited to the large scale simulation and modelling tasks, and this offers significant potential for advancing the art of computational science and engineering. The combination of the flexibility, parallelism, and energy efficiency of FPGA based platforms affords researchers and practitioners new ways to solve increasingly complex problems with unmatched performance and scalability. However, full realization of the power of reconfigurable computing in this domain requires addressing various challenges in the programming complexity, integration into software ecosystems, and system level design. New programming models, tools and architectural approaches to bridge this gap continue to emerge as the world of hardware design moves closer to (but has not yet made) complete acceptance of the world of software development. Future reconfigurable computing in large scale simulation and modelling seems promising with efforts in FPGAs technology, high level synthesis tools and domain specific languages that enable wider adoption and more accessible development workflow. As these technologies mature and new application areas emerge, reconfigurable computing is poised to increasingly magnify its impact in helping to push the bounds of computational modeling and simulation in the domains of science and engineering.

## References:


1. Chang, K. C., & Fan, C. P. (2019, May). Cost-Efficient Adaboost-based Face Detection with FPGA Hardware Accelerator. In *2019 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)* (pp. 1-2). IEEE.
2. Lee, J., He, J., & Wang, K. (2020, July). Neural networks and FPGA hardware accelerators for millimeter-wave radio-over-fiber systems. In *2020 22nd International Conference on Transparent Optical Networks (ICTON)* (pp. 1-4). IEEE.
3. Liu, Y., & Sarris, C. D. (2022). Efficient propagation modeling for communication channels with reconfigurable intelligent surfaces. *IEEE Antennas and Wireless Propagation Letters*, *21*(10), 2120-2124.
4. Budhu, J., & Grbic, A. (2020). Perfectly reflecting metasurface reflectarrays: Mutual coupling modeling between unique elements through homogenization. *IEEE Transactions on Antennas and Propagation*, *69*(1), 122-134.
5. Yu, L., Zhang, S., Wu, N., & Yu, C. (2022). FPGA-based hardware-in-the-loop simulation of user selection algorithms for cooperative transmission technology over LOS channel on geosynchronous satellites. *IEEE Access*, *10*, 6071-6083.
6. Nagaraju, V. S., Babu, P. A., Ratna, V. R., & Mariserla, R. (2020, December). Design and implementation of low power 32-bit comparator. In *Proceedings of the International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS 2020), Palai, India* (pp. 459-468).
7. Lian, Z., Su, Y., Wang, Y., & Jiang, L. (2021). A non-stationary 3-D wideband channel model for intelligent reflecting surface-assisted HAP-MIMO communication systems. *IEEE Transactions on Vehicular Technology*, *71*(2), 1109-1123.
8. Shaik, S., Hirao, H., & Kumar, D. (2007). Reactivity of high-valent iron–oxo species in enzymes and synthetic reagents: a tale of many states. *Accounts of chemical research*, *40*(7), 532-542.
9. Li, Z., Guo, S., Sun, Q., & Chan, G. K. L. (2019). Electronic landscape of the P-cluster of nitrogenase as revealed through many-electron quantum wavefunction simulations. *Nature chemistry*, *11*(11), 1026-1033.
10. Sharma, S., Sivalingam, K., Neese, F., & Chan, G. K. L. (2014). Low-energy spectrum of iron–sulfur clusters directly from many-particle quantum mechanics. *Nature chemistry*, *6*(10), 927-933.
11. Chaloun, T., Tabarani, F., & Wipf, S. T. (2018, April). A modular phased array transceiver with RF-MEMS SPDT switches in a 0.25 µm SiGe BiCMOS technology. In *12th European Conference on Antennas and Propagation (EuCAP 2018)* (p. 256). Stevenage UK: IET.
12. Xu, L., Zhou, X., Tao, Y., Yu, X., Yu, M., & Khan, F. (2021). AF relaying secrecy performance prediction for 6G mobile communication networks in industry 5.0. *IEEE Transactions on Industrial Informatics*, *18*(8), 5485-5493.
13. Zhong, R., Liu, Y., Mu, X., Chen, Y., Wang, X., & Hanzo, L. (2022). Hybrid reinforcement learning for STAR-RISs: A coupled phase-shift model based beamformer. *IEEE Journal on Selected Areas in Communications*, *40*(9), 2556-2569.
14. Björnson, E., Özdogan, Ö., & Larsson, E. G. (2019). Intelligent reflecting surface versus decode-and-forward: How large surfaces are needed to beat relaying?. *IEEE wireless communications letters*, *9*(2), 244-248.

15. Mohammed, Z. H., Chankaew, K., Vallabhuni, R. R., Sonawane, V. R., Ambala, S., & Markkandan, S. (2023). Blockchain-enabled bioacoustics signal authentication for cloud-based electronic medical records. *Measurement: Sensors*, *26*, 100706.
16. Xu, L., Zhou, X., Tao, Y., Yu, X., Yu, M., & Khan, F. (2021). AF relaying secrecy performance prediction for 6G mobile communication networks in industry 5.0. *IEEE Transactions on Industrial Informatics*, *18*(8), 5485-5493.
17. Perez-Palomino, G., Baine, P., Dickie, R., Bain, M., Encinar, J. A., Cahill, R., ... & Toso, G. (2013). Design and experimental validation of liquid crystal-based reconfigurable reflectarray elements with improved bandwidth in F-band. *IEEE Transactions on Antennas and Propagation*, *61*(4), 1704-1713.
18. Bildik, S., Dieter, S., Fritzsch, C., Menzel, W., & Jakoby, R. (2014). Reconfigurable folded reflectarray antenna based upon liquid crystal technology. *IEEE Transactions on antennas and propagation*, *63*(1), 122-132.
19. Tang, W., Chen, M. Z., Chen, X., Dai, J. Y., Han, Y., Di Renzo, M., ... & Cui, T. J. (2020). Wireless communications with reconfigurable intelligent surface: Path loss modeling and experimental measurement. *IEEE transactions on wireless communications*, *20*(1), 421-439.
20. Wu, Q., & Zhang, R. (2019). Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming. *IEEE transactions on wireless communications*, *18*(11), 5394-5409.
21. Huang, C., Zappone, A., Alexandropoulos, G. C., Debbah, M., & Yuen, C. (2019). Reconfigurable intelligent surfaces for energy efficiency in wireless communication. *IEEE transactions on wireless communications*, *18*(8), 4157-4170.
22. Tang, W., Chen, X., Chen, M. Z., Dai, J. Y., Han, Y., Di Renzo, M., ... & Cui, T. J. (2022). Path loss modeling and measurements for reconfigurable intelligent surfaces in the millimeter-wave frequency band. *IEEE Transactions on Communications*, *70*(9), 6259-6276.
23. Ellingson, S. W. (2021, September). Path loss in reconfigurable intelligent surface-enabled channels. In *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (pp. 829-835). IEEE.
24. Renjith, P. N., Bharati, R., Thiyagu, T. M., Vallabhuni, R. R., Mouleswararao, B., & Narayanan, L. (2023). Smart filtering for user discovery and availing balance storage space continuity with faster big data service. *Measurement: Sensors*, *26*, 100707.
25. Özdogan, Ö., Björnson, E., & Larsson, E. G. (2019). Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling. *IEEE Wireless Communications Letters*, *9*(5), 581-585.
26. Björnson, E., Özdogan, Ö., & Larsson, E. G. (2019). Intelligent reflecting surface versus decode-and-forward: How large surfaces are needed to beat relaying?. *IEEE wireless communications letters*, *9*(2), 244-248.
27. Zhong, R., Liu, Y., Mu, X., Chen, Y., Wang, X., & Hanzo, L. (2022). Hybrid reinforcement learning for STAR-RISs: A coupled phase-shift model based beamformer. *IEEE Journal on Selected Areas in Communications*, *40*(9), 2556-2569.
28. Wei, L., & Lau, W. C. (2023). Modelling the power of RFID antennas by enabling connectivity beyond limits. *National Journal of Antennas and Propagation, 5*(2), 43–48.
29. Rajput, A., Kumawat, R., Sharma, J., & Srinivasulu, A. (2024). Design of Novel High-Speed Energy-Efficient Robust 4:2 Compressor. Journal of VLSI Circuits and Systems, 6(2), 53–64. https://doi.org/10.31838/jvcs/06.02.06
30. Puri, A., & Lakhwani, K. (2013). Enhanced approach for handwritten text recognition using neural network. *International Journal of Communication and Computer Technologies, 1*(2), 79-82. https://doi.org/10.31838/IJCCTS/01.02.02
31. 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
32. Prasath, C. A. (2024). Optimization of FPGA architectures for real-time signal processing in medical devices. *Journal of Integrated VLSI, Embedded and Computing Technologies, 1*(1), 11-15. https://doi.org/10.31838/JIVCT/01.01.03
33. Thoi, N. T. (2025). Nanoparticle applications revolutionizing chemical processes. *Innovative Reviews in Engineering and Science, 2*(1), 13-21. https://doi.org/10.31838/INES/02.01.02
34. Rahim, R. (2024). Scalable architectures for real-time data processing in IoT-enabled wireless sensor networks. *Journal of Wireless Sensor Networks and IoT, 1*(1), 44-49. https://doi.org/10.31838/WSNIOT/01.01.07
35. Abdullah, D. (2024). Leveraging FPGA-based design for high-performance embedded computing. *SCCTS Journal of Embedded Systems Design and Applications, 1*(1), 37-42. https://doi.org/10.31838/ESA/01.01.07
36. Van, C., Trinh, M. H., & Shimada, T. (2025). Graphene innovations in flexible and wearable nanoelectronics. Progress in Electronics and Communication Engineering, 2(2), 10–20. https://doi.org/10.31838/PECE/02.02.02