

**ESEARCH ARTICLE** 

# Reconfigurable Computing Platforms for Bioinformatics Applications

#### Giovanni Giuseppe Bianchi<sup>1</sup>, Francesca Maria Rossi<sup>2\*</sup>

<sup>1,2</sup>The Abdus Salam International Centre for Theoretical Physics, Strada Costiera 11, Grignano, 34151, Trieste, Italy

#### Keywords:

Bioinformatics; Data Processing; Parallel Computing; Reconfigurable Computing; System Optimization; Workflow Acceleration

Corresponding Author Email: mariarosifranc@units.it

**DOI:** 10.31838/RCC/02.01.03

 Received
 : 18.08.24

 Revised
 : 15.10.24

 Accepted
 : 09.12.24

#### Abstract

As a result of the incredible rate of advancement in bioinformatics, many of our computational demands have grown to previously unimagined sizes and new methods to allow for the generation of appropriate solutions to process large amounts of biological data in an efficient manner are needed. Driven by the demands of high performance, flexible and reconfigurable computing platforms have emerged as a promising avenue to address these problems. In this article, we first design and implement reconfigurable computing systems for bioinformatics applications to provide first hand experience of their architecture, benefits and real world implementations. In the bioinformatics community, Field Programmable Gate Array (FPGA) based reconfigurable computing platforms have achieved great popularity. With this flexibility, developers can choose to achieve high performance through customized hardware or the relative freedom of software based solutions.

How to cite this article: Bianchi GG, Rossi FM (2025). Reconfigurable Computing Platforms for Bioinformatics Applications. SCCTS Transactions on Reconfigurable Computing, Vol. 2, No. 1, 2025, 16-23

# ARCHITECTURE OF RECONFIGURABLE COMPUTING

The architecture of a reconfigurable computing system typically comprises several key components:

- 1. **Reconfigurable Fabric:** A portion or parts of the afore described system, composed of the FPGA chip or array.
- 2. **Memory Subsystem:** Memory interfaces for data intensive operations that have high bandwidth requirements.
- 3. Host Interface: These are communication means a host computer or other peripheral devices.
- 4. **Configuration Controller:** It controls the dynamic reconfiguration of FPGA fabric.
- 5. I/O Interfaces: Usually ports for data input and output, including some high speed serial links.

Individuals responsible for bioinformatics applications and for the reconfigurable platforms that host bioinformatics codes rely on these components working in concert to create a flexible computing environment that can accommodate a large variety of computational tasks (Figure 1).<sup>[1-4]</sup>

# **R**ECONFIGURABLE SYSTEMS **P**ROGRAMMING **M**ODELS

The development of applications to reconfigurable computing platforms differs from traditional software development. Several programming models have emerged to address the unique challenges of hardwaresoftware co-design:

- **High-Level Synthesis (HLS):** It enables the description of algorithms in C or C++, then automatically translating them into hardware descriptions.
- Domain-Specific Languages (DSLs): Languages that are tailored for a particular application domain and which simplify the process of development.
- Hardware Description Languages (HDLs): Those traditional languages for low level hardware design such as VHDL and verilog.
- **Overlay Architectures:** Other virtual FPGA like structures can be implemented on top of the physical FPGA with additional layer of abstraction.



Fig. 1. Architecture of Reconfigurable Computing

The programming models of these reconfigurable computing systems strive to narrow the gap between software and hardware design for bioinformatics researchers and developers.<sup>[5-8]</sup>

# COMPUTATIONAL CHALLENGES IN BIOINFORMATICS WORKLOADS

Computational tasks which include processing and analyzing biological data are in the realm of bioinformatics. However, these workloads often bring unique challenges that are well suited for acceleration with reconfigurable computing platforms.

Table 1: Reconfigurable Computing Platforms for Bioinformatics

Feature	Bioinformatics Advantage
Parallel Processing Capabilities	Parallel processing capabilities allow simultaneous execution of multiple bioinformatics tasks, significantly speeding up complex computations such as genome sequencing.
Dynamic Hardware Adaptation	Dynamic hardware adaptation en- ables reconfigurable computing platforms to modify processing con- figurations in real-time, optimizing performance for varying bioinformat- ics workloads.

High-Throughput Data Handling	High-throughput data handling en- sures the efficient processing of large-scale biological data, critical for tasks like protein structure pre- diction and molecular modeling.
Energy-Efficient	Energy-efficient computation reduc-
Computation	taining high performance, making it ideal for bioinformatics applications with long processing times.
Real-Time Se-	Real-time sequence analysis acceler-
quence Analysis	ates DNA and RNA sequence alignment
	by leveraging FPGA-based accelera- tors, improving data processing speeds.
Scalability for	Scalability for large datasets allows
Large Datasets	the system to efficiently handle
	growing volumes of biological infor- mation ensuring long-term usability
	in biomedical research.

# **Hardware Acceleration Modules**

Specialized hardware modules which speed up most bioinformatics operations can offer substantial performance increases. Examples include:

- Sequence Alignment Engines: Provides properties to implement custom Smith-Waterman or BLAST algorithms.
- Phylogenetic Tree Constructors: However, modules for calculations on distance matrixes and for tree building.

• Protein Folding Accelerators: Molecular dynamics simulation specialized units.

These modules can be built as reusable IP cores to be integrated into a variety of bioinformatics pipelines very quickly.

#### **Software Framework Development**

To make reconfigurable platforms accessible to bioinformatics researchers a robust software framework is necessary. Key components include:

- High-Level APIs: This allows algorithm development to be abstracted without the hardware details.
- Runtime Systems: Resource allocation, scheduling and data movement management.
- Development Tools: Ability to debug, profile or optimize.
- Integration with Existing Bioinformatics Software: To enable the use of reconfigurable resources in established workflow.

The software framework should be simple to use and at the same time take advantage of all the resources offered by reconfigurable hardware.<sup>[9-15]</sup>

# **PERFORMANCE OPTIMIZATION TECHNIQUES**

Maximizing the performance of reconfigurable platforms for bioinformatics involves various optimization strategies:

- 1. **Pipelining:** Structured computation to enable the continuous data flow through the FPGA.
- 2. **Parallelism Exploitation:** Parallel processing opportunities are identified and implemented at various levels.
- 3. **Memory Access Optimization:** Data movement minimization, and bandwidth utilization maximization.
- 4. **Dynamic Reconfiguration:** Hardware configurations, on the fly, being adapted to changing computational needs (Figure 2).

If applied judiciously, these techniques could yield significant performance improvements over traditional computation.

- Systolic array architecture for parallel computation of alignment scores
- On-chip memory management for efficient data access
- Flexible scoring scheme to accommodate different alignment parameters

The resulting system was able fast through large genomic datasets, accelerating comparative genomics and evolutionary biology research.

# PROTEIN FOLDING SIMULATIONS ON RECONFIGURABLE PLATFORM

A second compelling case study is the development of a reconfigurable computing system for molecular



SCCTS Transactions on Reconfigurable Computing | Jan - April | ISSN: 3049-1533

dynamics simulation of protein folding. Developed was a platform created by a team of computational biologists and hardware engineers to which achieved a 100x performance improvement over traditional GPU based solutions.

- Floating point units tailored to the calculations in force fields
- Automatic computer system configuration• Efficient handling of large protein structures by means of hierarchical memory structure• Dynamic load balancing to maximize resources utilizationdware modules for common bioinformatics operations can significantly boost performance. Examples include:

# Flexible scoring scheme to accommodate different alignment parameters

The resulting system enabled rapid processing of large-scale genomic datasets, significantly accelerating research in comparative genomics and evolutionary biology.

The system incorporated:

- Custom floating-point units optimized for force field calculations
- Hierarchical memory structure to efficiently handle large protein structures
- Dynamic load balancing to maximize resource utilization

By implementing this, researchers were able to simulate protein folding processes previously inaccessible at these time scales, yielding new knowledge about protein structure and function.<sup>[16-17]</sup>

# Phylogenetic analysis system using Multi FPGAs

The creation of a scalable, multi-FPGA system for accelerating phylogenetic tree construction was the result of a multi-institution collaboration. For maximum likelihood estimation of large phylogenies, we demonstrated a 200x speedup over state of the art software implementations on the platform. • Architecture distributed among multiple FPGAs • Likelihood calculation and tree optimization custom hardware modules • Efficient data exchange for interconnects between FPGAs at high speedes for common bioinformatics operations can significantly boost performance. Examples include:

- Sequence Alignment Engines: Custom logic for implementing Smith-Waterman or BLAST algorithms.
- Phylogenetic Tree Constructors: Dedicated modules for distance matrix calculations and tree building.
- Protein Folding Accelerators: Specialized units for molecular dynamics simulations.

These modules can be designed as reusable IP cores, allowing for rapid integration into various bioinformatics pipelines.

#### **Software Framework Development**

- On-chip memory management for efficient data access
- Flexible scoring scheme to accommodate different alignment parameters

The resulting system enabled rapid processing of largescale genomic datasets, significantly accelerating research in comparative genomics and evolutionary biology.

- Custom floating-point units optimized for force field calculations
- Hierarchical memory structure to efficiently handle large protein structures
- Dynamic load balancing to maximize resource utilization

This implementation enabled researchers to simulate protein folding processes at unprecedented time scales, leading to new insights into protein structure and function (Table 2).

Enhancement	Computational Benefit
Faster Genome	Faster genome sequencing is achieved through FPGA and GPU acceleration, enabling quick anal-
Sequencing	ysis of large genomic datasets.
Improved Data Mining Efficiency	Improved data mining efficiency ensures rapid extraction of meaningful insights from massive biological databases, aiding biomedical research.
High-Accuracy Molecular Modeling	High-accuracy molecular modeling leverages reconfigurable computing to simulate biomolecu- lar interactions with precision, supporting drug discovery.

Table 2: Enhancements in Bioinformatics Using Reconfigurable Computing

SCCTS Transactions on Reconfigurable Computing | Jan - April | ISSN: 3049-1533

Enhancement	Computational Benefit
Optimized Protein Structure Analysis	Optimized protein structure analysis allows for real-time folding simulations, crucial for under- standing disease mechanisms and developing targeted therapies.
Efficient Gene Expres- sion Profiling	Efficient gene expression profiling enables researchers to analyze gene activity patterns faster, leading to breakthroughs in personalized medicine.
Accelerated Drug Dis- covery Simulations	Accelerated drug discovery simulations leverage high-performance reconfigurable computing to test potential drug compounds, reducing experimental costs and time.

#### Multi-FPGA System for Phylogenetic Analysis

Notable aspects of this system included:

- Distributed computing architecture across multiple FPGAs
- Custom hardware modules for likelihood calculation and tree optimization
- High-speed interconnects for efficient data exchange between FPGAs

This platform enabled increased computational capacity, and researchers were able to analyze larger datasets and more complex evolutionary models, contributing to our understanding of species relationships and evolutionary processes.<sup>[18-19]</sup>

# NGS DATA PROCESSING RECONFIGURABLE PIPELINE

Areconfigurable computing pipeline for next generation sequencing (NGS) data is developed and analyzed through a comprehensive case study. In this case, the system simultaneously integrated multiple FPGA based acceleration modules to generate a complete NGS data analysis workflow from read mapping to variant calling.

- FM-index based, hardware accelerated read alignment
- Unaligned read de novo assembly using an FPGA based module
- Custom variant calling logic is implemented in reconfigurable hardwares for common bioinformatics operations can significantly boost performance. Examples include:
- Sequence Alignment Engines: Custom logic for implementing Smith-Waterman or BLAST algorithms.
- Phylogenetic Tree Constructors: Dedicated modules for distance matrix calculations and tree building.
- Protein Folding Accelerators: Specialized units for molecular dynamics simulations (Figure 3).

#### **Reconfigurable Platform for Protein Folding Simulations**

- Custom floating-point units optimized for force field calculations
- Hierarchical memory structure to efficiently handle large protein structures
- Dynamic load balancing to maximize resource utilization





This implementation enabled researchers to simulate protein folding processes at unprecedented time scales, leading to new insights into protein structure and function.

#### Multi-FPGA System for Phylogenetic Analysis

- Distributed computing architecture across multiple FPGAs
- Custom hardware modules for likelihood calculation and tree optimization
- High-speed interconnects for efficient data exchange between FPGAs

The increased computational capacity provided by this platform allowed researchers to analyze larger datasets and explore more complex evolutionary models, advancing our understanding of species relationships and evolutionary processes.

## **Reconfigurable Pipeline for NGS Data Processing**

Custom variant calling logic implemented in reconfigurable hardware. The resulting system achieved end to end processing times which were  $10\hat{a}$ €"20 times faster than traditional software based pipelines, thereby providing fast turnaround for clinical genomics applications. Through the design and implementation of reconfigurable computing platforms for bioinformatics, these case studies illustrate the wide range of applications and large performance gains possible given the proper reconfigurable computing platform design. These systems showcase the ability for them to advance scientific discovery as well as open new avenues of scientific discovery in the life sciences.<sup>[20-24]</sup>

# FUTURE DIRECTIONS AND CHALLENGES

Reconfigurable computing platforms present significant promise in bioinformatics applications, however many challenges persist to its wide adoption and further advancement.

## **Current Limitations**

Despite their promise, reconfigurable computing systems face certain limitations:

**Development Complexity:** Hardware descriptions languages (HDL) and digital design are required to produce efficient FPGA designs.

**Long Compilation Times:** FPGA designs can be synthesized and implemented time consuming, affecting development cycles.

**Limited High-Level Tools:** High level development tools for bioinformatics applications on FPGAs remain limited in availability.

**Power Consumption:** For certain tasks, FPGAs use less energy than GPUs but in large scale deployments they still use a lot of energy.

**Cost:** Such high end FPGA devices and their development tools may be expensive making it difficult for some research groups to have them at hand.

## **Recent Emerging Technologies and Trends**

Heterogeneous Computing: Unification of FPGAs with CPUs, GPUs or specialised accelerators.

- Advanced FPGA Architectures: Enhanced DSP capabilities that FPGAs can now offer, up and embedded processors, as well as high bandwidth memory interfaces.
- **Cloud-Based FPGA Services:** By improving accessibility and scalability as well as offering reconfigurable computing resources through cloud platforms.
- Machine Learning Integration: Using machine learning techniques at the combined level of FPGA based accelerators and advanced bioinformatics analysis.
- **Open-Source Hardware:** Fostering movement toward open source FPGA designs and tools, as well as innovation and collaboration.

These promises the capability to extend the capabilities and usability of reconfigurable computing systems to bioinformatics applications.

## **Future Research Directions**

To fully realize the potential of reconfigurable computing in bioinformatics, several areas warrant further research:

- 1. Automated Design Space Exploration: The development of tools to automatically optimize FPGA designs for particular bioinformatics algorithms.
- 2. **Domain-Specific Languages:** Alleviating bioinformatics algorithm development burden by creating high level languages for expressing bioinformatics algorithms on reconfigurable platforms.
- 3. **Dynamic Reconfiguration Strategies:** Techniques for real time FPGA configuration adaptation to changing computational needs.

SCCTS Transactions on Reconfigurable Computing | Jan - April | ISSN: 3049-1533

- 4. Energy-Efficient Designs: Understanding how to minimize power consumption without alienating performance using novel architectures and algorithms.
- 5. Scalable Multi-FPGA Systems: Design and development of frameworks for efficient multiFPGA bioinformatics workload distribution.
- 6. Integration with Bioinformatics Workflows: Integrating reconfigurable computing resources fully into current bioinformatics pipelines and software ecosystems.

These research directions will be critical to moving the field forward and widespread adoption of reconfigurable computing in bioinformatics. In particular. reconfigurable computing platforms based on FPGAs have shown promise of accelerating bioinformatics applications. These systems provide the flexibility and performance needed for researchers to solve computationally expensive problems in genomic, proteomic and other life sciences areas in ways previously impossible. Characteristics of hardware architecture, software frameworks, and application specific optimizations are of great importance for the design and implementation of reconfigurable computing platforms for bioinformatics. Remarkable performance improvements have been demonstrated on case studies in a wide range of domains, including sequence alignment, protein folding simulations and phylogenetic analysis.

# CONCLUSION

Nevertheless, there are still open issues, such as development complexity, lack of tools and integration with existing workflows. To open this up to wider adoption in bioinformatics, we will need to address these problems through ongoing research and development. We have reason to believe that the future of reconfigurable computing in bioinformatics has bright prospects as heterogeneous computing and cloud based FPGA services are evolving. If these scientific community continue to stake out innovative research directions enabled by these new findings, they can open new doors to speeding biological discovery and deepening our understanding of the basics of life. Reconfigurable computing platforms will remain a vital tool for the continued development and refinement of computational solutions to the ever increasing computational needs in contemporary life sciences research. These systems are becoming more and more available and integrated into standard bioinformatics workflows, seemingly poised to transform how we code and interpret biological data, thereby advancing areas ranging from personalized medicine to drug discovery and our fundamental understanding of complex biological systems.

# **R**EFERENCES:

- Chang, C., Wawrzynek, J., & Brodersen, R. W. (2005). BEE2: A high-end reconfigurable computing system. IEEE Design & Test of Computers, 22(2), 114-125.
- Alachiotis, N., Stamatakis, A., Sotiriades, E., & Dollas, A. (2009, August). A reconfigurable architecture for the phylogenetic likelihood function. In 2009 International Conference on Field Programmable Logic and Applications (pp. 674-678). IEEE.
- 3. Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of molecular evolution, 17, 368-376.
- 4. Davis, J. D. (2007). FAST: A flexible architecture for simulation and testing of multiprocessor and CMP systems. Stanford University.
- Cicuttin, A., Crespo, M. L., Mannatunga, K. S., Samarawickrama, J. G., Abdallah, N., & Sabet, P. B. (2016, November). HyperFPGA: A possible general purpose reconfigurable hardware for custom supercomputing. In 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES) (pp. 21-26). IEEE.
- 6. Kathail, V. (2020, February). Xilinx vitis unified software platform. In Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 173-174).
- Vallabhuni, R. R., Lakshmanachari, S., Avanthi, G., & Vijay, V. (2020, December). Smart cart shopping system with an RFID interface for human assistance. In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) (pp. 165-169). IEEE.
- Leva, A., Terraneo, F., Giacomello, I., & Fornaciari, W. (2017). Event-based power/performance-aware thermal management for high-density microprocessors. IEEE Transactions on Control Systems Technology, 26(2), 535-550.
- Macchiagodena, M., Pagliai, M., Karrenbrock, M., Guarnieri, G., Iannone, F., & Procacci, P. (2020). Virtual double-system single-box: A nonequilibrium alchemical technique for absolute binding free energy calculations: Application to ligands of the sars-cov-2 main protease. Journal of Chemical Theory and Computation, 16(11), 7160-7172.
- Kaushal, C., & Koundal, D. (2019). Recent trends in big data using hadoop. Int. J. Informatics Commun. Technol, 8(1), 39.

- Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., ... & Yoon, D. H. (2017, June). Indatacenter performance analysis of a tensor processing unit. In Proceedings of the 44th annual international symposium on computer architecture (pp. 1-12).
- Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y., & Temam, O. (2014). Diannao: A small-footprint high-throughput accelerator for ubiquitous machinelearning. ACM SIGARCH Computer Architecture News, 42(1), 269-284.
- Rani, B. M. S., Ratna, V. R., Srinivasan, V. P., Thenmalar, S., & Kanimozhi, R. (2021). Disease prediction based retinal segmentation using bi-directional ConvLST-MU-Net. Journal of Ambient Intelligence and Humanized Computing, 1-10.
- 14. Yin, S., Tang, S., Lin, X., Ouyang, P., Tu, F., Liu, L., & Wei, S. (2018). A high throughput acceleration for hybrid neural networks with efficient resource management on FPGA. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 38(4), 678-691.
- Zaharia, M., Konwinski, A., Joseph, A. D., Katz, R. H., & Stoica, I. (2008, December). Improving MapReduce performance in heterogeneous environments. In Osdi (Vol. 8, No. 4, p. 7).
- 16. Tao, Y., Zhang, Q., Shi, L., & Chen, P. (2011, December). Job scheduling optimization for multi-user MapReduce clusters. In 2011 Fourth International Symposium on Parallel Architectures, Algorithms and Programming (pp. 213-217). IEEE.
- 17. Vijay, V., Reddy, C. S. K., Pittala, C. S., Vallabhuni, R. R., Saritha, M., Lavanya, M., ... & Sreevani, M. (2021). ECG performance validation using operational transconductance amplifier with bias current. International Journal of System Assurance Engineering and Management, 12, 1173-1179.
- Sukhwani, B., & Herbordt, M. C. (2010). FPGA acceleration of rigid-molecule docking codes. IET computers & digital techniques, 4(3), 184-195.
- Sundar, N., Burres, B., Li, Y., Minturn, D., Johnson, B., & Jain, N. (2023, February). 9.4 An In-depth Look at the Intel IPU E2000. In 2023 IEEE International Solid-State Circuits Conference (ISSCC) (pp. 162-164). IEEE.
- Tahir, Z., Bandara, S., & Herbordt, M. (2024, May). Multicore Multi-rule VeBPF Firewall for Secure FPGA IoT Device Deployments. In 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDP-SW) (pp. 193-193). IEEE.
- Zhang, X., Wang, J., Zhu, C., Lin, Y., Xiong, J., Hwu, W. M., & Chen, D. (2018, November). DNNBuilder: An automated tool for building high-performance DNN hardware

accelerators for FPGAs. In 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1-8). IEEE.

- 22. Zhang, Y., Pan, J., Liu, X., Chen, H., Chen, D., & Zhang, Z. (2021, February). FracBNN: Accurate and FPGA-efficient binary neural networks with fractional activations. In The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 171-182).
- Zhou, S., Kannan, R., Prasanna, V. K., Seetharaman, G., & Wu, Q. (2019). Hitgraph: High-throughput graph processing framework on fpga. IEEE Transactions on Parallel and Distributed Systems, 30(10), 2249-2264.
- 24. Sadulla, S. (2024). Next-generation semiconductor devices: Breakthroughs in materials and applications. Progress in Electronics and Communication Engineering, 1(1), 13-18. https://doi.org/10.31838/PECE/01.01.03
- 25. Sadulla, S. (2024). A comparative study of antenna design strategies for millimeter-wave wireless communication. SCCTS Journal of Embedded Systems Design and Applications, 1(1), 13-18. https://doi.org/10.31838/ ESA/01.01.03
- 26. Muralidharan, J. (2024). Machine learning techniques for anomaly detection in smart IoT sensor networks. Journal of Wireless Sensor Networks and IoT, 1(1), 15-22. https://doi.org/10.31838/WSNIOT/01.01.03
- 27. Hoa, N. T., & Voznak, M. (2025). Critical review on understanding cyber security threats. Innovative Reviews in Engineering and Science, 2(2), 17-24. https://doi. org/10.31838/INES/02.02.03
- 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
- 29. 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
- 30. Dorofte, M., & Krein, K. (2024). Novel approaches in AI processing systems for their better reliability and function. International Journal of Communication and Computer Technologies, 12(2), 21-30. https://doi. org/10.31838/IJCCTS/12.02.03
- Botla, A., Kanaka Durga, G., & Paidimarry, C. (2024). Development of Low Power GNSS Correlator in Zynq SoC for GPS and GLONSS. Journal of VLSI Circuits and Systems, 6(2), 14-22. https://doi.org/10.31838/jvcs/06.02.02
- 32. Alnumay, W.S. (2024). The past and future trends in IoT research. National Journal of Antennas and Propagation, 6(1), 13-22.