SCCTS Transactions on Reconfigurable Computing, ISSN: 3049-1533 Vol. 2, No. 2, 2025 (pp. 1-7)
RESEARCH ARTICLE ECEJOURNALS.IN

Runtime Reconfiguration Techniques for Efficient Operation of FPGA Based Systems in Real Time Environments

Choi Min-Young[1], Jang Hyun-Soo[2], Jeon Hye-Jin[3]*

[1-3]Department of Electronic Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea

Keywords:
Efficient Operation;
FPGA Systems;
Real-Time Processing;
Runtime Reconfiguration;
System Optimization;
Reconfigurable Computing

Corresponding Author Email:
jinhyej@seoultech.ac.kr

DOI: 10.31838/RCC/02.02.01

Received : 19.12.24
Revised : 13.03.25
Accepted : 02.05.25

Abstract

Now, field-programmable gate array (FPGA) technology is rapidly developing, and runtime reconfiguration is being established as a critical capability for real time systems. This article discusses current state of the art techniques in implementing efficient runtime reconfiguration within FPGAs systems, while still meeting strict real time requirements and minimizing power consumption and resource utilization. However, the ability to dynamically reconfigure hardware resources on-the-fly on FPGAs as they are applied from automotive control systems to aerospace and telecommunications, opens up exciting new possibilities as FPGAs are found. Then, we explore power implications, pay close attention to innovative methods for partial reconfiguration, and discuss building flexible runtime systems that can change with changes in application demands. At the end of this comprehensive guide, readers will be fully acquainted with state-of-the-art reconfiguration techniques, important design considerations and future trends of adaptive FPGA based systems. If you're an embedded systems engineer, FPGA designer, or reconfigurable computing researcher, this article will be useful to you in allowing you to take advantage of runtime reconfiguration in your next project. Understanding of the FPGA Architecture and the reconfiguration capabilities.

How to cite this article: Min-Young C, Hyun-Soo J, Hye-Jin J (2025). Runtime Reconfiguration Techniques for Efficient Operation of FPGA Based Systems in Real Time Environments. SCCTS Transactions on Reconfigurable Computing, Vol. 2, No. 2, 2025, 1-7

Introduction

The landscape of field-programmable gate array (FPGA) technology is rapidly evolving, with runtime reconfiguration emerging as a game-changing capability for real-time systems. This article explores cutting-edge techniques for implementing efficient runtime reconfiguration in FPGA-based platforms, with a focus on meeting stringent real-time constraints while optimizing power consumption and resource utilization. As FPGAs continue to find applications in domains ranging from automotive control systems to aerospace and telecommunications, the ability to dynamically reconfigure hardware resources on-the-fly opens up exciting new possibilities. We'll examine innovative approaches for managing partial reconfiguration, analyze power implications, and discuss strategies for building flexible runtime systems that can adapt to changing application demands (Figure 1).[1-6]

By the end of this comprehensive guide, readers will gain a thorough understanding of state-of-the-art reconfiguration techniques, key design considerations, and emerging trends shaping the future of adaptive FPGA-based systems. Whether you're an embedded systems engineer, FPGA designer, or researcher in reconfigurable computing, this article provides valuable insights to help you leverage the full potential of runtime reconfiguration in your next project.[7-12]

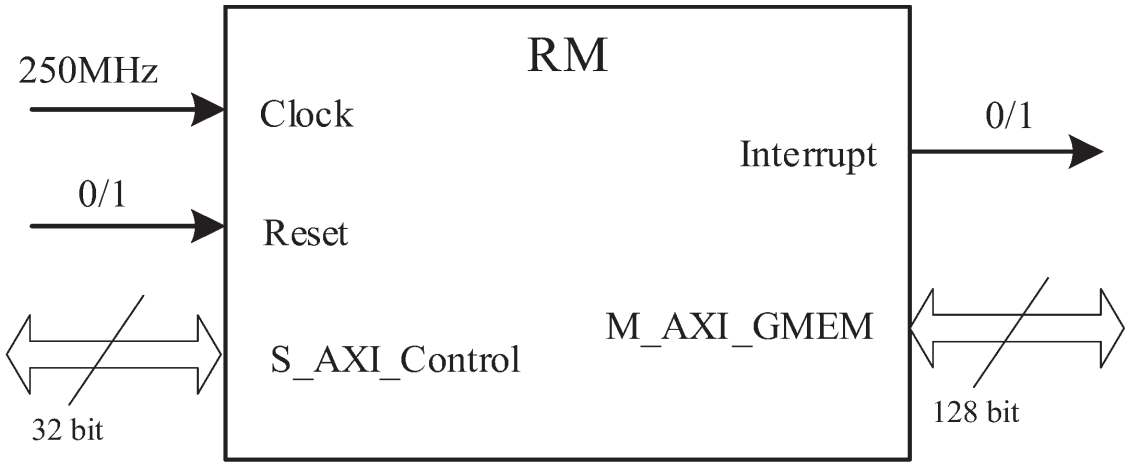

Fig. 1: Inter-Module Communication Protocols

Inter-Module Communication Protocols

Implement handshaking mechanisms to manage reconfiguration events. Use message passing interfaces for loosely-coupled modules. Consider shared memory architectures for high-bandwidth data exchange. Develop error detection and recovery mechanisms for reliable communication. Implement standardized I/O interfaces (e.g. PCIe, Ethernet). Design bridge modules to handle protocol translations. Consider using soft processors for complex control and communication tasks. Implement DMA controllers for efficient data transfer. Memory Management in Reconfigurable Systems.[13-18]

Bitstream Storage Strategies

Use external memory (e.g. Flash, DDR) for storing large bitstream libraries. Implement on-chip caching of frequently used configurations. Consider using compression techniques to reduce storage requirements. Develop strategies for managing partial bitstreams. Data Management for Reconfigurable Modules. Implement mechanisms for state preservation during module swapping. Use shared memory regions for inter-module data exchange. Memory Virtualization Techniques. Implement memory management units (MMUs) for address translation. Use memory overlays to maximize utilization of limited on-chip memory. Develop paging mechanisms suitable for reconfigurable architectures. Consider implementing garbage collection for dynamic memory allocation. Security Considerations. Implement encryption for stored bitstreams and runtime data. Use secure boot mechanisms to ensure system integrity. Develop access control mechanisms for reconfigurable regions. Consider using physical unclonable functions (PUFs) for hardware-based security. Performance Analysis and Optimization. Metrics for Evaluating Reconfigurable Systems. Reconfiguration overhead (time and energy). Resource utilization efficiency. Adaptation responsiveness. Overall system throughput and latency. Profiling and Monitoring Techniques. Implement on-chip performance counters. Use logic analyzers for detailed timing analysis. Develop software profiling tools for high-level performance assessment. Consider using machine learning techniques for predictive performance modeling. Optimization Strategies. Pipeline reconfiguration processes to overlap with computation. Implement speculative preloading of configurations. Use partial reconfiguration to minimize reconfiguration overhead. Explore multi-context FPGAs for rapid switching between configurations (Table 1).[19-27]

Design Space Exploration

Develop automated tools for design space exploration. Use heuristic algorithms to navigate complex optimization landscapes. Consider multi-objective optimization techniques to balance competing goals. Implement runtime adaptation mechanisms based on performance feedback. Case Studies: Real-World Applications. Adaptive Signal Processing Systems. Dynamic adaptation of filter coefficients. Real-time switching between different modulation schemes. Adaptive beamforming for phased array antennas. Cognitive radio systems with spectrum sensing capabilities. Automotive Control Systems. Adaptive cruise control systems. Dynamic reconfiguration of engine control units. Flexible infotainment systems. Advanced driver assistance features with real-time

Table 1: Runtime Reconfiguration for FPGA Systems

Approach	Objective
Dynamic Partial Reconfiguration	Dynamic partial reconfiguration allows FPGA components to be modified while the system is running, improving system flexibility and efficiency.
Adaptive Resource Allocation	Adaptive resource allocation ensures that FPGA resources are distributed based on real-time workload demands, optimizing performance and minimizing waste.
Task Migration	Task migration involves shifting workloads between different hardware components, balancing resource utilization and reducing processing bottlenecks.
Reconfiguration Overhead Minimization	Reconfiguration overhead minimization techniques aim to reduce the time and energy spent during reconfiguration, improving overall system performance in real-time environments.
On-the-Fly Hardware Modifications	On-the-fly hardware modifications enable the FPGA system to adapt its hardware structure dynamically to meet the evolving needs of real-time applications.
Self-Healing Mechanisms	Self-healing mechanisms allow FPGA-based systems to automatically recover from faults by reconfiguring the hardware to bypass faulty areas, maintaining reliable operation.

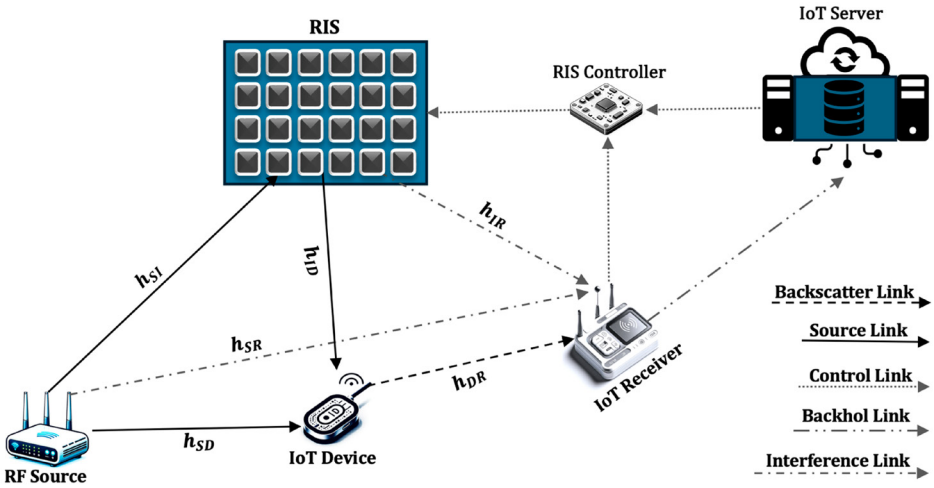

Fig. 2. Design Space Exploration

adaptation. Aerospace and Defense Applications. Software-defined radios for satellite communication. Adaptive navigation systems. Real-time image processing for reconnaissance. Fault-tolerant avionics systems with dynamic hardware redundancy (Figure 2).[28-34]

High-Performance Computing

Adaptive accelerators for scientific simulations. Real-time reconfiguration for financial modeling. Dynamic load balancing in heterogeneous computing clusters. Energy-efficient data center applications with workload-specific optimizations. Future Directions and Challenges. Current Limitations. Large designs have high reconfiguration overhead. Little tool support for partial reconfiguration workflows. Design and verification of reconfigurable systems is highly complex. High reconfiguration overhead for large designs. Limited tool support for partial reconfiguration workflows. Complexity in designing and verifying reconfigurable systems. Power consumption concerns during frequent reconfigurations. Emerging Technologies. 3D-stacked FPGAs with dedicated reconfiguration layers. Non-volatile configuration memory for instant-on capabilities. Silicon photonics for ultra-fast reconfiguration. Neuromorphic architectures with inherent adaptability. Research Opportunities.[35-37] Automated design tools for reconfigurable systems. Practical runtime compilation techniques for hardware generation at just in time. Autonomous reconfiguration capabilities in self adaptive systems. Employing machine learning for designing reconfiguration strategies. Standardized interfaces for reconfigurable modules. Reconfigurable system performance can be evaluated by using. Common benchmarks Multi vendor reconfigurable platform interoperability standards. Mission-critical reconfigurable systems safety &

security certifications, several challenges remain. High reconfiguration overhead for large designs. Limited tool support for partial reconfiguration workflows. Complexity in designing and verifying reconfigurable systems. Power consumption concerns during frequent reconfigurations. Emerging Technologies. Several promising technologies may address current limitations. 3D-stacked FPGAs with dedicated reconfiguration layers. Non-volatile configuration memory for instant-on capabilities. Silicon photonics for ultra-fast reconfiguration. Neuromorphic architectures with inherent adaptability.[38-41]

Research Opportunities

Exciting avenues for future research include. Automated design tools for reconfigurable systems. Runtime compilation techniques for just-in-time hardware generation. Self-adaptive systems with autonomous reconfiguration capabilities. Integration of machine learning for optimizing reconfiguration strategies. Standardization Efforts. Developing industry standards will be crucial for wider adoption. Standardized interfaces for reconfigurable modules. Common benchmarks for evaluating reconfigurable system performance. Interoperability standards for multi-vendor reconfigurable platforms. Safety and security certifications for mission-critical reconfigurable systems. The field of runtime reconfiguration for FPGA based systems faces challenges which can be addressed and technologies emerging in the field can be explored to add to continued innovation and growth.[23-25]

Table 2: Optimizations Using Runtime Reconfiguration in FPGA Systems

Optimization	Benefit
Reduced Latency	Reduced latency ensures fast response times by minimizing the delays caused by reconfiguration processes, critical for real-time systems.
Enhanced Throughput	Enhanced throughput is achieved by optimizing the FPGA's performance in real-time, allowing for more data processing without compromising on speed.
Energy Savings	Energy savings are realized by adjusting FPGA configurations to match the workload, avoiding unnecessary power consumption during idle or low-demand periods.
Scalable Design	Scalable design allows FPGA systems to be easily expanded or adapted to handle larger datasets and more complex applications in dynamic environments.
Fault Tolerance	Fault tolerance ensures continued operation despite hardware failures, achieved by utilizing reconfigurable components to adapt to faulty areas or components.
Resource Utilization Efficiency	Resource utilization efficiency improves by dynamically reconfiguring the FPGA to allocate resources only when needed, optimizing the system's resource allocation and performance.

A powerful way to create flexible, efficient and adaptive hardware platforms for applications in real-time environments is through runtime reconfiguration techniques for FPGA based systems. These techniques provide new possibilities for a number of applications, from automotive control systems, to aerospace and high performance computing by enabling dynamic modification of hardware functionality. Flex runtime system design. Finding reconfiguration specific to optimize power consumption. Partial re configuration workflow implementation. Real time scheduling for reconfigurable architectures. Dynamic modules communication infrastructures. Memory management strategies are These performance analysis and optimization techniques.[26-27]

Despite significant progress, several challenges remain. High reconfiguration overhead for large designs. Limited tool support for partial reconfiguration workflows. Complexity in designing and verifying reconfigurable systems. Power consumption concerns during frequent reconfigurations. Emerging Technologies. Several promising technologies may address current limitations. 3D-stacked FPGAs with dedicated reconfiguration layers. Non-volatile configuration memory for instant-on capabilities. Silicon photonics for ultra-fast reconfiguration. Neuromorphic architectures with inherent adaptability. Exciting avenues for future research include. Automated design tools for reconfigurable systems. Runtime compilation techniques for just-in-time hardware generation. Self-adaptive systems with autonomous reconfiguration capabilities. Integration of machine learning for optimizing reconfiguration

strategies. Standardization Efforts. Developing industry standards will be crucial for wider adoption. Standardized interfaces for reconfigurable modules. Common benchmarks for evaluating reconfigurable system performance. Interoperability standards for multi-vendor reconfigurable platforms. Safety and security certifications for mission-critical reconfigurable systems. By addressing these challenges and exploring emerging technologies, the field of runtime reconfiguration for FPGA-based systems is poised for continued innovation and growth.

Conclusion

Designing flexible runtime systems. Optimizing power consumption during reconfiguration. Implementing partial reconfiguration workflows. Real-time scheduling for reconfigurable architectures. Communication infrastructures for dynamic modules. Memory management strategies. Performance analysis and optimization techniques. We've also looked at real world case studies, and discuss future directions into the field by highlighting the current challenges and exciting future avenues of innovation. The ability of FPGA based systems to dynamically adapt to changing requirements and environments provides the future of FPGA based systems. Research and development in areas such as automated design tools, self-adaptive architectures and emerging reconfiguration technologies are continuing to release even more powerful, flexible reconfigurable systems in the near future.

References:

1. Pittala, C. S., Parameswaran, V., Srikanth, M., Vijay, V., Siva Nagaraju, V., Venkateswarlu, S. C., ... & Vallabhuni, R. R. (2021). Realization and comparative analysis of thermometer code based 4-bit encoder using 18 nm FinFET technology for analog to digital converters. In *Soft Computing and Signal Processing: Proceedings of 3rd ICSCSP 2020, Volume 1* (pp. 557-566). Singapore: Springer Singapore.
2. Vallabhuni, R. R., Koteswaramma, K. C., & Sadgurbabu, B. (2020, October). Comparative validation of SRAM cells designed using 18nm FinFET for memory storing applications. In *Proceedings of the 2nd International Conference on IoT, Social, Mobile, Analytics & Cloud in Computational Vision & Bio-Engineering (ISMAC-CVB 2020)*.
3. Vallabhuni, R. R., Karthik, A., Kumar, C. V. S., Varun, B., Veerendra, P., & Nayak, S. (2020, December). Comparative Analysis of 8-Bit Manchester Carry Chain Adder Using FinFET at 18nm Technology. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 1579-1583). IEEE.
4. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5).
5. DiLuoffo, V., Michalson, W. R., & Sunar, B. (2018). Robot Operating System 2: The need for a holistic security approach to robotic architectures. *International Journal of Advanced Robotic Systems*, *15*(3), 1729881418770011.
6. Colaprete, A., Andrews, D., Bluethmann, W., Elphic, R. C., Bussey, B., Trimble, J., ... & Captain, J. E. (2019, December). An overview of the volatiles investigating polar exploration rover (viper) mission. In *AGU fall meeting abstracts* (Vol. 2019, pp. P34B-03).
7. Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM*, *34*(4), 30-44.
8. Grossmann, A., & Morlet, J. (1984). Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, *15*(4), 723-736.
9. Krishnan, S., Wan, Z., Bhardwaj, K., Whatmough, P., Faust, A., Neuman, S., ... & Reddi, V. J. (2021). Autopilot: Automating soc design space exploration for swap constrained autonomous uavs. *arXiv preprint arXiv:2102.02988*.
10. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5).
11. DiLuoffo, V., Michalson, W. R., & Sunar, B. (2018). Robot Operating System 2: The need for a holistic security approach to robotic architectures. *International Journal of Advanced Robotic Systems*, *15*(3), 1729881418770011.
12. Colaprete, A., Andrews, D., Bluethmann, W., Elphic, R. C., Bussey, B., Trimble, J., ... & Captain, J. E. (2019, December). An overview of the volatiles investigating polar exploration rover (viper) mission. In *AGU fall meeting abstracts* (Vol. 2019, pp. P34B-03).
13. Karunaratne, M., Mohite, A. K., Mitra, T., & Peh, L. S. (2017, June). HyCUBE: A CGRA with reconfigurable single-cycle multi-hop interconnect. In *Proceedings of the 54th Annual Design Automation Conference 2017* (pp. 1-6).
14. Sarwar, M. H., Shah, M. A., Umair, M., & Faraz, S. H. (2019, September). Network of ECUs Software Update in Future vehicles. In *2019 25th International Conference on Automation and Computing (ICAC)* (pp. 1-5). IEEE.
15. Mirfakhraie, T., Vitor, G., & Grogan, K. (2018, July). Applicable protocol for updating firmware of automotive hvac electronic control units (ecus) over the air. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social*

Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 21-26). IEEE.

16. Xie, Y., Zeng, G., Kurachi, R., Xiao, F., & Takada, H. (2021). Optimizing extensibility of CAN FD for automotive cyber-physical systems. *IEEE Transactions on Intelligent Transportation Systems*, *22*(12), 7875-7886.
17. Selesnick, I. W., Baraniuk, R. G., & Kingsbury, N. C. (2005). The dual-tree complex wavelet transform. *IEEE signal processing magazine*, *22*(6), 123-151.
18. Krishnan, S., Wan, Z., Bhardwaj, K., Whatmough, P., Faust, A., Wei, G. Y., ... & Reddi, V. J. (2020). The sky is not the limit: A visual performance model for cyber-physical co-design in autonomous machines. *IEEE Computer Architecture Letters*, *19*(1), 38-42.
19. Kepner, J., Aaltonen, P., Bader, D., Buluç, A., Franchetti, F., Gilbert, J., ... & Moreira, J. (2016, September). Mathematical foundations of the GraphBLAS. In *2016 IEEE High Performance Extreme Computing Conference (HPEC)* (pp. 1-9). IEEE.
20. Ojika, D., Gordon-Ross, A., Lam, H., & Patel, B. (2019). FAAM: FPGA-as-a-microservice-a case study for data compression. In *EPJ Web of Conferences* (Vol. 214, p. 07029). EDP Sciences.
21. Lienen, C., Platzner, M., & Rinner, B. (2020, December). Reconros: Flexible hardware acceleration for ros2 applications. In *2020 International Conference on Field-Programmable Technology (ICFPT)* (pp. 268-276). IEEE.
22. Liu, S., Pittman, R. N., & Forin, A. (2010, February). Minimizing partial reconfiguration overhead with fully streaming DMA engines and intelligent ICAP controller. In *FPGA* (p. 292).
23. Ojika, D., Gordon-Ross, A., Lam, H., & Patel, B. (2019). FAAM: FPGA-as-a-microservice-a case study for data compression. In *EPJ Web of Conferences* (Vol. 214, p. 07029). EDP Sciences.
24. Kim, W., Gupta, M. S., Wei, G. Y., & Brooks, D. (2008, February). System level analysis of fast, per-core DVFS using on-chip switching regulators. In *2008 IEEE 14th International Symposium on High Performance Computer Architecture* (pp. 123-134). IEEE.
25. Lienen, C., Platzner, M., & Rinner, B. (2020, December). Reconros: Flexible hardware acceleration for ros2 applications. In *2020 International Conference on Field-Programmable Technology (ICFPT)* (pp. 268-276). IEEE.
26. Liu, S., Pittman, R. N., & Forin, A. (2010, February). Minimizing partial reconfiguration overhead with fully streaming DMA engines and intelligent ICAP controller. In *FPGA* (p. 292).
27. Suleiman, A., Zhang, Z., Carlone, L., Karaman, S., & Sze, V. (2019). Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones. *IEEE Journal of Solid-State Circuits*, *54*(4), 1106-1119.
28. Hussein, A. S., Anwar, A., Fahmy, Y., Mostafa, H., Salama, K. N., & Kafafy, M. (2021). Implementation of a dpu-based intelligent thermal imaging hardware accelerator on fpga. *Electronics*, *11*(1), 105.
29. Lu, Z., Wang, J., Li, Z., Chen, S., & Wu, F. (2021). A resource-efficient pipelined architecture for real-time semi-global stereo matching. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(2), 660-673.
30. Lu, Z., Wang, J., Li, Z., Chen, S., & Wu, F. (2021). A resource-efficient pipelined architecture for real-time semi-global stereo matching. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(2), 660-673.
31. Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., ... & Kumar, V. (2018). Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, *3*(2), 965-972.
32. Alkady, G. I., El-Araby, N. A., Abdelhalim, M. B., Amer, H. H., & Madian, A. H. (2014, October). A fault-tolerant technique to detect and recover from open faults in FPGA interconnects. In *2014 14th Biennial Baltic Electronic Conference (BEC)* (pp. 69-72). IEEE.
33. Jacobs, A., Cieslewski, G., George, A. D., Gordon-Ross, A., & Lam, H. (2012). Reconfigurable fault tolerance: A comprehensive framework for reliable and adaptive FPGA-based space computing. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, *5*(4), 1-30.
34. Wang, G., Liu, S., & Sun, J. (2018). A dynamic partial reconfigurable system with combined task allocation method to improve the reliability of FPGA. *Microelectronics reliability*, *83*, 14-24.
35. Mullapudi, R. T., Mark, W. R., Shazeer, N., & Fatahalian, K. (2018). Hydranets: Specialized dynamic architectures for efficient inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8080-8089).
36. Takhirov, Z., Wang, J., Saligrama, V., & Joshi, A. (2016, August). Energy-efficient adaptive classifier design for mobile systems. In *Proceedings of the 2016 International Symposium on Low Power Electronics and Design* (pp. 52-57).
37. Vaithianathan, M., Patil, M., Ng, S. F., & Udkar, S. (2023). Comparative study of FPGA and GPU for high-performance computing and AI. *ESP International Journal of Advancements in Computational Technology (ESP-IJACT)*, *1*(1), 37-46.
38. Julian, A., Mary, G. I., Selvi, S., Rele, M., & Vaithianathan, M. (2024). Blockchain based solutions for privacy-preserving authentication and authorization in networks. *Journal of Discrete Mathematical Sciences and Cryptography*, *27*(2-B), 797-808.
39. Liu, S., & Gaudiot, J. L. (2020). Autonomous vehicles lite self-driving technologies should start small, go slow. *IEEE Spectrum*, *57*(3), 36-49.

40. Danne, K., & Platzner, M. (2005, May). Periodic real-time scheduling for FPGA computers. In *Third International Workshop on Intelligent Solutions in Embedded Systems, 2005.* (pp. 117-127). IEEE.
41. Saha, S., Sarkar, A., & Chakrabarti, A. (2015). Scheduling dynamic hard real-time task sets on fully and partially reconfigurable platforms. *IEEE Embedded Systems Letters*, *7*(1), 23-26.
42. Abdullah, D. (2024). Enhancing cybersecurity in electronic communication systems: New approaches and technologies. *Progress in Electronics and Communication Engineering, 1*(1), 38–43. https://doi.org/10.31838/PECE/01.01.07
43. 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
44. 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
45. Surendar, A. (2024). Emerging trends in renewable energy technologies: An in-depth analysis. *Innovative Reviews in Engineering and Science, 1*(1), 6-10. https://doi.org/10.31838/INES/01.01.02
46. Jagan, B. O. L. (2024). Low-power design techniques for VLSI in IoT applications: Challenges and solutions. *Journal of Integrated VLSI, Embedded and Computing Technologies, 1*(1), 1-5. https://doi.org/10.31838/JIVCT/01.01.01
47. Muralidharan, J. (2023). Innovative RF design for high-efficiency wireless power amplifiers. *National Journal of RF Engineering and Wireless Communication, 1*(1), 1-9. https://doi.org/10.31838/RFMW/01.01.01
48. Sudha, M., Karthikeyan, S., Daniel, J., & Muthupandian, R. (2024). Wearable device for heart rate monitoring. *International Journal of Communication and Computer Technologies, 12*(1), 27-32. https://doi.org/10.31838/IJCCTS/12.01.03
49. Abdul, A. M., & Nelakuditi, U. R. (2021). A New Blind Zone Free PFD in Fractional-N PLL for Bluetooth Applications. Journal of VLSI Circuits and Systems, 3(1), 19–24. https://doi.org/10.31838/jvcs/03.01.04
50. Halily, R., & Shen, M. (2024). Directing techniques for high frequency antennas for use in next-generation telecommunication countries. *National Journal of Antennas and Propagation, 6*(1), 49–57