RESEARCH ARTICLE ECEJOURNALS.IN

Using FPGA-Based Embedded Systems for Accelerated Data Processing Analysis

Koopman Choset[1], Jungmayr Bindal[2]*

[1,2]Chair for Compiler Construction, TU Dresden, Germany

Keywords:
Low-Power Embedded Systems;
Embedded Systems Security;
Wireless Sensor Networks;
Autonomous Systems;
Industrial Automation

Corresponding Author Email:
Bind.jun@tu-dresden.de

DOI: 10.31838/ESA/02.01.08

Received : 28.09.24
Revised : 30.11.24
Accepted : 30.12.24

Abstract

Over the past years as the technological world changes the demand of high performance energy efficient computing is at its peak. Traditional computing architectures are usually inadequate for processing the massive amount of data required by industries in real time. FPGA-based embedded systems is a pioneering path which harnesses the power of field programmable gate arrays (FPGAs) and the power of embedded processing to realize accelerated data processing. This in-depth exploration of FPGA based embedded systems covers their advantages, challenges and change that serves as an essential entry point in the journey of FPGA based embedded systems. We will cover how these systems are revolutionizing these industries -- automotive applications, industrial automation, telecommunications, defense -- and more in rethinking a way to solve difficult computational problems. We will dissect the inner workings of FPGA embedded systems on this journey, analyze the role they play in modern computing paradigms and best practices in implementations. Whether you are an engineer with experience or a techophile at heart, this article seeks to equip you with a comprehensive knowledge of this fast moving technology so you're not just aware of it, but also ready to combine it with some elementary engineering to realize some interesting, powerful methods of data processing.

How to cite this article: Choset K, Bindal J (2025). Using FPGA-Based Embedded Systems for Accelerated Data Processing Analysis. SCCTS Journal of Embedded Systems Design and Applictions, Vol. 2, No. 1, 2025, 79-85

Embedded systems using FPGA

In the accelerated data processing, we observe the genius combination of FPGAs and embedded system. With this powerful combination, a versatile computing platform is built to handle challenging algorithms with great efficiency. In this post, we assume a general knowledge of what FPGAs are, how they operate and what are their benefits to the world of embedded system design and development. However, if you do not, you may find the first of this series of posts, which explains and provides an introduction, here. Field programmable gate array (FPGA) is a well known semiconductor device constituted of an array of configurable logic blocks that are connected to each other by interconnects which are programmable. Unlike fixed architecture processors found in traditional processors, FPGAs enable the change of them to reprogram and reconfigure after manufacturing. Such flexibility enables engineers to provide the best fit for the hardware to targeted application needs, optimizing performance and power consumption.

However, embedded systems are computer systems that are designed and built for performing specific tasks within certain larger systems or products. Today these systems are everywhere we use them, from smartphones and home appliances to industrial machinery, automotive control units etc. However, with FPGAs embedded in embedded systems, they become highly efficient, application specific solutions that optimally incorporate strengths of both FPGAs and embedded systems. This is a powerful platform that makes use of software programmability and hardware flexibility when you marry FPGAs and embedded systems together. This synergy allows developers to implement custom hardware accelerators for computationally intensive tasks and maintain the capability of running software on integrated processors. It yields a highly amenable system that is able to otherwise provide very high performance over a variety of application.[1-4]

Advantages of FPGA Based Embedded Systems

Adoption of FPGA based embedded systems not just offers benefits concurrently, but also makes them an obvious choice for developers and organizations looking for high performance, energy efficient computing solutions. So now let's look at the underlying reasons we need to move to such systems, as opposed to the traditional Moore's Law driven computing current we're in today. Unparalleled flexibility of FPE based embedded systems is by far the biggest benefit. FPGAs are reconfigurable and therefore allow engineers to reprogram hardware function after hardware deployment, which is impossible with fixed function processors. It brings flexibility to development time, lower costs, and speed of reactivity to new market demands or changing technological standards (Figure 1).

One additional compelling advantage of FPGA based systems is the exceptional performance. FPGAs have the parallel processing capabilities allowing them to execute multiple operations simultaneously, hence we achieve significantly faster data processing as fast as sequential execution on a general purpose processor. Because of this, FPGA-based embedded systems are well suited for such applications as high frequency trading, video processing, or autonomous vehicle control systems due to the real time or near real time processing associated with them. Another area where FPGA based embedded systems really stand out is energy efficiency. Based on application specific hardware accelerators, these systems achieve higher performance/watt than general purpose processor executing the same functions using software. In particular, power consumption is a critical considerations in edge computing and IoT applications, making this efficiency doubly important.[5-9]

FPGA Based Embedded Systems: Applications

Embedded systems based on FPGA have found wide use in a large number of industries and applications due to the versatility and performance of FPGA based embedded system. These systems are making their impact in many areas of interest to the consumer, and as well as participating in highly cutting edge scientific research. We will examine some of the most notable applications where FPGA based embedded systems are leading the way to innovation and efficiency. FPGA based systems are ubiquitous in automotive applications such as advanced driver assistance systems (ADAS) and autonomous vehicle development. Real time data processing for these systems derived from multiple sensors, cameras, and radar units is performed, which facilitates rapid decision making

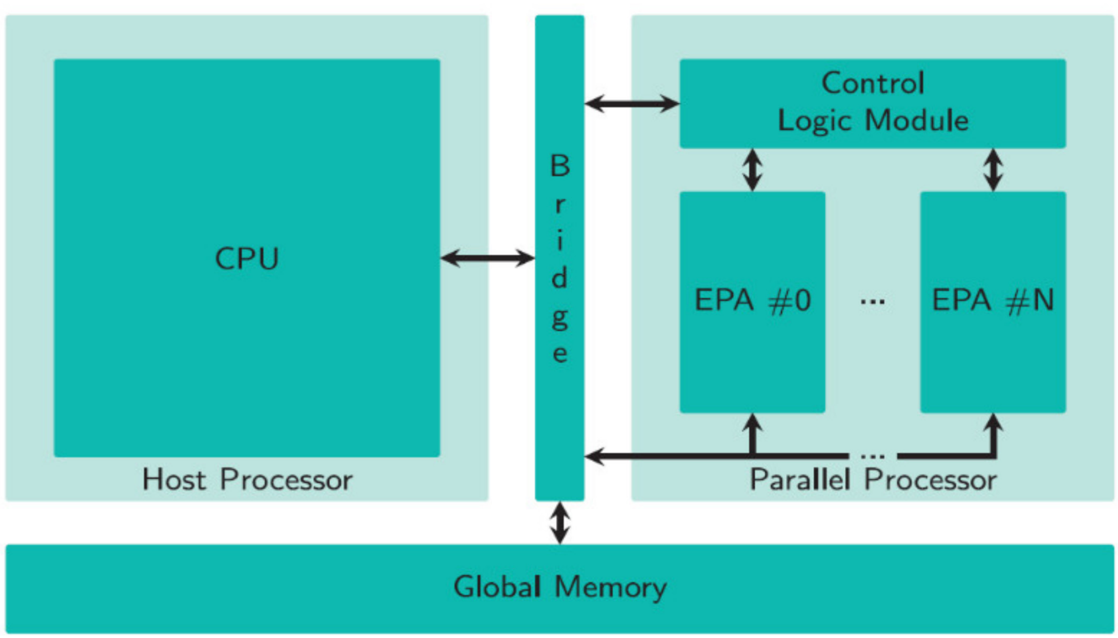

Fig. 1: Advantages of FPGA Based Embedded Systems

and increased vehicle safety. FPGAs' reconfigurability also enables the updates and improvements to these systems when technology evolves.

The industry of telecommunications employs often FPGA based embedded systems for network infrastructure and signal processing. In base stations, routers and switches, they are used for deploying complex communication protocols and dealing with high bandwidth data streams. By reconfiguring FPGAs telecom companies can adapt their hardware to new standards and technologies as new ones emerge without having to engineer equal or more advanced systems. Factors, FPGA based embedded systems are rewriting the rules of industrial automation and manufacturing processes. They serve as fast and real time control of robotic systems, precise motion control and fast image processing for quality inspection. Maintaining production efficiency and product quality requires the low latency performance of these systems.[10-13]

FPGA Based Embedded System Design Challenge

FPGA based embedded systems provide many positive attributes but present challenges unique of design and implementation to the developer. They are hurdles that need to be understood to successfully use the full potential of this technology. In this work, we consider some of the key challenges in the embedded system design using FPGA and introduce strategies to overcome them.

Hardware software co design is one of the major challenges. Unlike conventional software development, FPGA-based systems necessitate that engineers view their system as consisting of both hardware and software components. For developers that are used to just have pure software solution, this paradigm shift feels a bit daunting. For effective partitioning of tasks between hardware and software, system requirements must be carefully understood, and simultaneous optimization of performance, power consumption, and resource utilization is required (Table 1).[1]

However, the learning curve for FPGAs development tools and workflows are another major hurdle. FPGAs bring steeper learning curves compared to traditional software IDEs as well; due to this, FPGA design environments like Xilinx Vivado, Intel Quartus, etc. Because architectures will be abstracted and manipulated using HDLs such as VHDL or Verilog, engineers must become familiar with the concept of design space, timing analysis, and place and route. This transition is often time consuming and may require extra training or expertise.

Other hurdle in FPGA based system design is resource management and optimization. FPGAs have finite resources such as logic cells, memory blocks, and DSP slices. However, each of the logic cells, memory blocks, and other slices can be used independently by your code depending on the needs. It is difficult to efficiently exploit these resources to meet performance requirements for the complex design. Since processing is already in place, developers need to optimize their designs carefully to avoid resource bottlenecks, as well as meet timing constraints.[14-19]

Overcoming Design Challenges

In order to overcome the difficulties present in designing FPGA based embedded systems, developers and organizations have strategies and best practices

Table 1: FPGA Architectures for Data Processing Acceleration

Architecture	Feature
Pipeline Architecture	Pipeline architecture allows for continuous data processing, enabling higher throughput by processing multiple stages of data in parallel.
Parallel Processing	Parallel processing allows multiple computations to be performed simultaneously, maximizing data throughput and minimizing processing time.
Dataflow Architecture	Dataflow architecture optimizes the movement of data through various processing units, ensuring efficient data handling and reducing bottlenecks.
Reconfigurable Logic	Reconfigurable logic allows the FPGA to be customized for specific tasks, enabling dynamic adaptation to different data processing requirements.
Dual-Port Memory	Dual-port memory allows for simultaneous read and write operations, improving memory access and ensuring efficient data flow in processing tasks.
Embedded DSPs	Embedded DSPs (Digital Signal Processors) accelerate signal processing tasks by providing dedicated hardware for high-performance mathematical computations.

to employ. By using these approaches, teams can reduce the time needed to develop this incredibly powerful technology and most importantly maximize its benefits. High level synthesis (HLS) tools can play a one effective strategy. First, these tools enable developers to specify hardware functionality in a high level programming language such as C or C++ then automatically converts this information into Hardware Description Language (HDL) code. And this is a nice way to bridge the gap between software and hardware development so that if you're a software engineer, you can easily start to develop on FPGA. The use of HLS tools can greatly reduce development time, as well as lower the barrier to entry of FPGA based system design.

A second way is to take a modular design style approach. Complex systems become easier to reason about by 'breaking it down' into distinct, reusable components (modules) which the developer can manage in isolation, thus increasing design reusability. Also this approach is collaborative development because the other team members are working on the different modules of it simultaneously. The use of modular design principles can bring more maintainable and scalable FPGA based embedded systems. However, to cross the learning curve when it comes to designing in FPGA design tools can't be done without investing in comprehensive training and skill development. By making available to their engineering team training resource, workshops as well as hands on experience with FPGA development environment, the organizations should make resources of this kind available. A solid foundation built upon FPGA design principles and best practices will pave a path to a productive team and a great design.

FPGA-Based SoC: The Next Evolution

As FPGA technology continues to advance, a new paradigm has emerged: System on Chip (SoC) designs based on FPGA. By bridging the reconfigurability of FPGAs with the power of integrated processors, these innovative platforms provide a compelling solution to a multitude of applications. We take a look at the idea of FPGA based SoCs and how they contribute to embedded system design. As FPGA based SoC is a single chip, it integrates one or more processor cores (generally ARM based or RISC-V) along with a complete bank of FPGA fabric on the same die. Seamless communication between custom hardware accelerators implemented in the FPGA logic and the processor is enabled by this tight integration. This category's popular platforms are the Xilinx Zynq and Intel Stratix 10 SoC FPGAs.

While there are many advantages to using FPGA based SoCs. They offer the best of both worlds: the software programmability of traditional processors and hardware flexibility of FPGAs. The combination allows software to be used to implement complex algorithms and also offloads computationally intensive tasks to custom hardware accelerators. Its result is a highly efficient and versatile platform which is capable of dealing with various applications. FPGA based SoCs have the important advantage of being able to evolve with changing requirements. Developers are thus free to update the FPGA portion of the design without changing the processor software, as new algorithms or standards arise. This is especially important in fast emerging technology areas like artificial intelligence and machine learning where new techniques are being designed all the time.

AI and Machine Learning on FPGA Acceleration

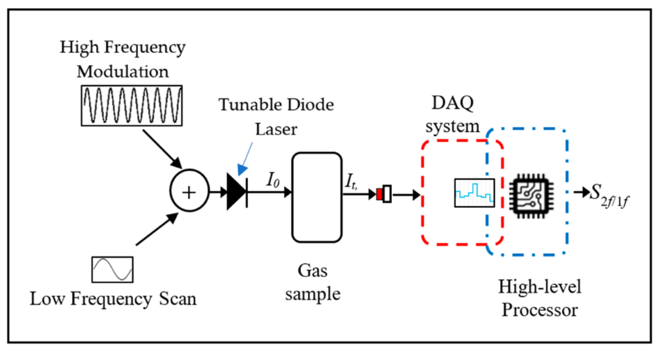

Fig. 2: AI and Machine Learning on FPGA Acceleration

With an increasingly rapid growth Artificial Intelligence and Machine Learning, it has spawned demand for high performance computing platforms that can execute intricate algorithms quickly. FPGA based embedded systems are an emerging solution for accelerating AI and ML workloads with unique performance, flexibility and energy efficiency. FPGAs have proven to be one of the key advantages of FPGAs when it comes to AI applications because they can actually implement custom hardware accelerator for specific algorithm. For instance, convolutional neural networks (CNNs) as used for image recognition are highly parallelizable to FPGAs, providing significant performance benefits compared to general purpose execution. Second, the reconfigurable nature of FPGAs permits developers to quickly adjust the hardware to emerging AI algorithms.

Polysynaptic computation at the edge also suits FPGA-based systems well because they provide good velocity in terms of power and weight wasted at the point of use. Processing data locally on FPGA accelerated embedded systems decreases the need for cloud connectivity as well as response times for time sensitive applications. Areas such as autonomous vehicles, industrial automation, as well as smart city infrastructure make use of this approach the most. Another big usage of FPGAs AI and ML is the way they can work with various data type and formats. FPGAs support different precision levels and data representation than fixed function AI accelerators, so developers can reconfigure their design to provide optimal performance for use cases. This flexibility is very important in such a fast evolving field like AI, as new techniques, as well as new optimizations emerge every day.

Low Latency Applications and Real Time Processing

FPGA based embedded systems are one of the most compelling use cases for FPGA based systems due to the reasons they can perform real time processing and processing with extremely low latency. FPGAs provide the parallel processing capabilities and the customizable hardware architecture suitable for situations where every millisecond counts. As we go on and see how these systems are revolutionizing industries that require quick processing of data and rapid decision making. FPGA systems are extensively utilized in the financial sector implementing the high frequency trading and the real time market data analysis. Financial institutions compete in fast paced markets through the ability to process market feeds and trade with ultra low latency. So, FPGAs can run complex trading algorithms directly in hardware and reduce latency to microseconds or even to nanoseconds.

Embedded systems based on FPGAs have also proven to excel in the wireless communications area, for instance to support 5G networks, whose applications include autonomous vehicles and remote surgery which defined extremely low latencies. Complex signal processing algorithms and protocol stacks, required for these cutting edge applications can be implemented directly in hardware on FPGA based systems, thereby expediting the rapid data processing necessary. FPGA based systems are used in industrial automation for real time control of manufacturing processes. These systems help to process sensor data, take decisions and control actuators with deterministic timing and ensure precise and consistent operation of machinery. FPGA based solutions have the low latency necessary to ensure product quality and maximize production efficiency (Table 2).

Energy Efficiency and Power Management

High performance computing is a demand that continues to grow, requiring the need for electric energy efficient embedded system design. In this area, the advantages of FPGA based embedded systems far exceed those of traditional computing architectures, providing a balance of performance and power consumption that is hard to achieve on other platforms. Application specific hardware accelerators are one of the key factors that enable the high

Table 2: Optimization Techniques for FPGA Data Processing

Technique	Benefit
Clock Gating	Clock gating reduces power consumption by disabling the clock signal to unused logic, minimizing energy usage during idle periods.
Memory Optimization	Memory optimization techniques ensure efficient usage of memory resources, reducing latency and improving overall system performance.
Algorithm Parallelization	Algorithm parallelization breaks complex data processing tasks into smaller sub-tasks that can be executed simultaneously on FPGA resources.
Resource Mapping	Resource mapping allocates FPGA resources effectively to avoid underutilization and maximize the efficiency of the data processing tasks.
Power Management	Power management techniques adjust the operating power levels of FPGA components to balance performance and energy consumption.
I/O Optimization	I/O optimization reduces data transfer bottlenecks by ensuring efficient management of input and output operations, improving data processing speed.

energy efficiency of FPGA based systems. FPGAs can realize a higher performance per watt than general purpose processors that perform the same functions in software, by tailoring the hardware to the specific requirements of the application. In battery powered devices as well as edge computing applications where power consumption is key, this efficiency is indeed valuable.

Complementary to this, FPGA vendors have also improved the power efficiency of their devices. Modern FPGAs have sold equipped with advanced power management features, as partial reconfiguration and power gating, so that unused parts of the chip can be powered down when not in use. The techniques presented here enable developers to design for both performance and power consumption. One advantage of energy efficiency of FPGA based systems is they can adapt to changing workloads. FPGAs operate in contrast to fixed function ASICs, and — unlike fixed function ASICs– can be reconfigured on the fly to optimize power consumption according to current processing requirements. It provides the system with the dynamic adaptability to be running at peak efficiency all the time, no matter what task the system happens to be doing.

Future Trends and Innovations

Given the rapid rate of technology advancement, FPGA based embedded systems will also have an increasing role in shaping the future of computing. Now you are ready to explore some of the emerging trends and innovations that should power these systems' development in the upcoming years. And one of the most exciting trends we're seeing right now is taking AI and machine learning capabilities and integrating them directly into FPGA hardware. Another path towards scaling the AI engine fabric to meet a target performance footprint is for FPGA vendors to develop specialized AI engines and tensor processing units that integrate seamlessly into traditional FPGA fabric. We expect this trend to expedite the adoption of FPGAs for edge AI applications to consolidate AI inference at the edge while improving performance.

Another area of innovation involves the design of high bandwidth memory interfaces for FPGAs. With increasing processing data demands, memory bandwidth often limits the performance of the system. In order to overcome this challenge, new memory technologies, such as High Bandwidth Memory (HBM) are being integrated onto FPGA platforms, which increase performance for more data intensive applications. In the world of FPGAs, software defined hardware is also becoming a popular concept. The philosophy behind this approach is to decrease the development barrier needed for FPGA development by offering higher level abstractions and tools to software engineers. With these technologies maturing, we can expect to see more spread adoption of FPGA based solution to several industries.

Conclusion

Embedded systems based on FPGAs have proven to be a powerful and versatile solution for accelerating data processing in the current fast developing technological environment. These systems provide unmatched performance, energy efficiency and flexibility across a wide variety of applications, by harnessing the flexibility of field programmable gate arrays (FPGAs) and the power of embedded processing. During this exploration, we have learned of many of the benefits of FPGA based embedded systems such as ability to implement custom hardware accelerators, suitability for real time processing and low latency applications. We also discuss ways for overcoming these hurdles and their challenges through the design and implementation. Moving forward, FPGA based embedded systems are poised to be a cornerstone to innovation across a wide range of industries. AI and machine learning and 5G communications and beyond await these versatile platforms to take on the most challenging computational challenges of tomorrow. It is no longer an option, but required, for engineers, developers and organizations who want to remain at the forefront of technology by embracing FPGA based embedded systems. These innovative platforms give us the opportunity to harness their power in data processing as we create new pave the way for applications and solutions that will help define our digital future.

References:

1. Lesén, E., Björholt, I., Ingelgård, A., & Olson, F. J. (2017). Exploration and preferential ranking of patient benefits of medical devices: a new and generic instrument for health economic assessments. *International Journal of Technology Assessment in Health Care*, *33*(4), 463-471.
2. Mao, Y., & Zhang, L. (2021). Optimization of the medical service consultation system based on the artificial intelligence of the internet of things. *IEEE Access*, *9*, 98261-98274.

3. Masterson, F., & Cormican, K. (2013). Overview of the regulation of medical devices and drugs in the European union and the United States. *Therapeutic Innovation & Regulatory Science*, *47*(6), 715-722.
4. Ma, K., Wang, X., & Wang, Y. (2013). DPPC: dynamic power partitioning and control for improved chip multiprocessor performance. *IEEE Transactions on Computers*, *63*(7), 1736-1750.
5. Vijay, V., Sreevani, M., Rekha, E. M., Moses, K., Pittala, C. S., Shaik, K. S., ... & Vallabhuni, R. R. (2022). A Review On N-Bit Ripple-Carry Adder, Carry-Select Adder And Carry-Skip Adder. *Journal of VLSI circuits and systems*, *4*(01), 27-32.
6. Megiddo, N., & Modha, D. S. (2004). Outperforming LRU with an adaptive replacement cache algorithm. *Computer*, *37*(4), 58-65.
7. Levy, M., & Huli, C. (2019, September). Design thinking in a nutshell for eliciting requirements of a business process: A case study of a design thinking workshop. In *2019 IEEE 27th international requirements engineering conference (RE)* (pp. 351-356). IEEE.
8. Chang, H. F. (2013). *A synthesis approach to manage complexity in software systems design* (Doctoral dissertation, University of Southern California).
9. Draft, S. (2013). Computer science curricula 2013. *ACM and IEEE Computer Society, Incorporated: New York, NY, USA*.
10. Babu, P. A., Nagaraju, V. S., & Vallabhuni, R. R. (2021, June). Speech emotion recognition system with librosa. In *2021 10th IEEE international conference on communication systems and network technologies (CSNT)* (pp. 421-424). IEEE.
11. Andersson, B., Baruah, S., & Jonsson, J. (2001, December). Static-priority scheduling on multiprocessors. In *Proceedings 22nd IEEE Real-Time Systems Symposium (RTSS 2001)(Cat. No. 01PR1420)* (pp. 193-202). IEEE.
12. Azevedo, A., Issenin, I., Cornea, R., Gupta, R., Dutt, N., Veidenbaum, A., & Nicolau, A. (2002, March). Profile-based dynamic voltage scheduling using program checkpoints. In *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition* (pp. 168-175). IEEE.
13. Kivity, A., Kamay, Y., Laor, D., Lublin, U., & Liguori, A. (2007, July). kvm: the Linux virtual machine monitor. In *Proceedings of the Linux symposium* (Vol. 1, No. 8, pp. 225-230).
14. Dall, C., & Nieh, J. (2014). KVM/ARM: the design and implementation of the linux ARM hypervisor. *Acm Sigplan Notices*, *49*(4), 333-348.
15. Vallabhuni, R. R., Shruthi, P., Kavya, G., & Chandana, S. S. (2020, December). 6Transistor SRAM cell designed using 18nm FinFET technology. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 1584-1589). IEEE.
16. Xi, S., Wilson, J., Lu, C., & Gill, C. (2011, October). RT-Xen: Towards real-time hypervisor scheduling in Xen. In *Proceedings of the ninth ACM international conference on Embedded software* (pp. 39-48).
17. Buerki, R., & Rueegsegger, A. K. (2013). Muen-an x86/64 separation kernel for high assurance. *University of Applied Sciences Rapperswil (HSR), Tech. Rep*.
18. Feng, L., Huang, J., Hu, J., & Reddy, A. (2021). Fastcfi: Real-time control-flow integrity using fpga without code instrumentation. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, *26*(5), 1-39.
19. Kadar, M., Fohler, G., Kuzhiyelil, D., & Gorski, P. (2021, May). Safety-aware integration of hardware-assisted program tracing in mixed-criticality systems for security monitoring. In *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)* (pp. 292-305). IEEE.
20. Kesana, S., Prem Sai, N. N., Yallaiah, O., Saikrishna, S., & Khadarvali, S. (2021). Circularly polarized fractal patch antenna with probe feed technique for Wi-Max applications. *National Journal of Antennas and Propagation, 3*(2), 15–20
21. Kumar, C. V. S. R., & Nelakuditi, U. R. (2021). Hardware/Software Co-Design Using ZYNQ SoC. Journal of VLSI Circuits and Systems, 3(1), 14–18. https://doi.org/10.31838/jvcs/03.01.03
22. Gopalakrishnan, K., Lakshmanan, R., Naveen, V., TamilKumaran, S., & Venkatesh, S. (2017). Digital signature manager. *International Journal of Communication and Computer Technologies, 5*(2), 53-57.
23. Kavitha, M. (2023). Beamforming techniques for optimizing massive MIMO and spatial multiplexing. *National Journal of RF Engineering and Wireless Communication, 1*(1), 30-38. https://doi.org/10.31838/RFMW/01.01.04
24. Kavitha, M. (2024). Energy-efficient algorithms for machine learning on embedded systems. *Journal of Integrated VLSI, Embedded and Computing Technologies, 1*(1), 16-20. https://doi.org/10.31838/JIVCT/01.01.04
25. Kavitha, M. (2024). Enhancing security and privacy in reconfigurable computing: Challenges and methods. *SCCTS Transactions on Reconfigurable Computing, 1*(1), 16-20. https://doi.org/10.31838/RCC/01.01.04
26. Abdullah, D. (2024). Recent advancements in nanoengineering for biomedical applications: A comprehensive review. *Innovative Reviews in Engineering and Science, 1*(1), 1-5. https://doi.org/10.31838/INES/01.01.01
27. Kavitha, M. (2024). Environmental monitoring using IoT-based wireless sensor networks: A case study. *Journal of Wireless Sensor Networks and IoT, 1*(1), 50-55. https://doi.org/10.31838/WSNIOT/01.01.08
28. Kavitha, M. (2024). Advances in wireless sensor networks: From theory to practical applications. *Progress in Electronics and Communication Engineering, 1*(1), 32–37. https://doi.org/10.31838/PECE/01.01.06