Real-Time Monitoring And Assessment System In Continuous Monitoring Of System Workloads And Performance Using Soc Architectures

Ramachandra Kumar Gundu[1], Budati Anil Kumar[2*],Vijaya Saradhi Dommeti[3], Katragadda Swetha[4]

[1]Department of ECE, Koneru Lakshmaiah Education Foundation, Hyderabad, India & Department of ECE, Geethanjali College of Engineering and Technology, Hyderabad, India. E-mail: ramachandrag1418@gmail.com

[*2]Department of ECE, Koneru Lakshmaiah Education Foundation, Hyderabad, India. E-mail:b.anilkumar@klh.edu.in

[3]Department of ECE, Malineni Lakshmaiah Women's Engineering College, Guntur,A.P.,India. E-mail: saradhi.gitam@gmail.com, Swethakatragadda88@gmail.com

Abstract:

In modern SoC designs, balancing energy efficiency and performance is crucial. This research proposes a novel approach to dynamically adapt communication protocols (AXI, AHB, APB) based on real-time system requirements and workloads. The goal is to develop an adaptive mechanism that intelligently switches between these protocols or adjusts their configurations to optimize power consumption and processing efficiency. The proposed research focuses on "Dynamic Protocol Adaptation in SoC Architectures for Energy Efficiency and Performance Optimization," aiming to transform the management of communication protocols within SoC designs. The core of this research involves developing a Verilog-based system capable of dynamically switching between AXI, AHB, and APB protocols based on real-time system demands. By incorporating a real-time monitoring system to assess current workloads and performance metrics, this approach seeks to optimize both energy consumption and processing efficiency. The novelty of this research is twofold: firstly, it introduces a dynamic adaptation mechanism that contrasts with traditional static protocol implementations, enabling more flexible and context-aware operation. This adaptability ensures the most appropriate protocol is used according to the system's specific requirements at any given time. Secondly, it simultaneously addresses energy and performance optimization, which is a pioneering dual-focus approach. The expected outcomes include enhanced system efficiency, with significant reductions in power consumption and improvements in processing performance. By tailoring protocol use to actual operational needs, the research aims to improve the overall effectiveness and robustness of SoC designs. Validation will be achieved through comprehensive simulation and real-world testing of the proposed system. This includes comparing the dynamic adaptation approach against static protocol implementations in terms of energy efficiency, performance, and design complexity. The results will be analyzed to demonstrate the practical benefits and viability of the dynamic adaptation mechanism, ensuring that the proposed system is both theoretically advanced and practically applicable, offering scalable solutions for integration into existing SoC architectures.

Keywords: SoC Architecture, RAM Material; Communication Protocol.

1. INTRODUCTION

In the evolving landscape of System-on-Chip (SoC) design, the efficient management of communication protocols is crucial for balancing performance and energy consumption. SoCs, which integrate multiple

functionalities onto a single chip, often use various communication protocols to handle different types of data transfers. The AXI (Advanced eXtensible Interface), AHB (Advanced High-performance Bus), and APB (Advanced Peripheral Bus) protocols are commonly employed in these systems, each serving distinct roles in data handling and system communication. Traditionally, SoC designs use fixed protocol configurations, where the choice of AXI, AHB, or APB is predetermined based on design specifications and requirements. While this static approach simplifies design and implementation, it fails to adapt to the dynamic nature of real-world applications. As a result, systems may either overutilize resources or underperform, leading to inefficiencies in power consumption and processing. This research proposes a novel approach to SoC protocol management by introducing "Dynamic Protocol Adaptation." The goal is to develop a Verilog-based system capable of real-time switching between AXI, AHB, and APB protocols based on current system demands. By implementing a real-time monitoring and assessment mechanism, the system will dynamically select the most suitable protocol for various operational contexts. This adaptive mechanism aims to enhance both energy efficiency and system performance, providing a more responsive and optimized SoC architecture. Dynamic protocol adaptation offers several advantages over traditional static methods. First, it provides a flexible and context-aware approach to protocol management, ensuring that the chosen protocol aligns with the specific needs of the system at any given time. This flexibility can lead to significant improvements in energy efficiency, as the system can switch to more power-efficient protocols when high performance is not required. Second, by optimizing protocol selection based on real-time metrics, the approach can enhance overall system performance, including data transfer rates and latency. The expected outcomes of this research include improved system efficiency, with reduced power consumption and enhanced processing capabilities. The dynamic adaptation approach aims to bridge the gap between theoretical advancements and practical applications, offering scalable solutions that can be integrated into existing SoC designs. Validation of this research will involve rigorous simulation and real-world testing to compare the dynamic adaptation approach with traditional static implementations. The results will be assessed to demonstrate the effectiveness of the proposed system in optimizing energy usage and performance, ensuring that it meets the practical needs of modern SoC designs. By advancing the state of SoC protocol management, this research aims to contribute significantly to the field of embedded system design and optimization.

2. LITERATURE SURVEY

Li et al. [1] introduced a method for distinguishing between informed and inactive nodes to enhance the representation of information coverage. They formulated a novel problem called topic-aware information coverage maximization, aimed at maximizing the total expected number of both active and informed nodes within topic-aware social networks. To address this problem, they developed a heuristic approach based on preprocessing. An experimental study was conducted to evaluate the effectiveness of this approach. A semantic-guiding adversarial network for creating human trajectories is presented by Xiong et al. [2]. Their strategy includes several crucial elements: Initially, an attention-based generator is created to generate trajectory locations sequentially. Incomplete trajectory sequences are then extended and transformed into images that depict their spatial structure using a rollout module. Finally, the realism of these trajectory images is assessed using a discriminator based on convolutional neural networks. Using policy gradient techniques, the discriminator's output is used as a reward signal to improve the generator. A technique for creating a business process model (BPM) and extracting emergency elements (EELs) from emergency plans is presented by Zhu et al. [3]. To find EELs from complicated sentences in emergency plan texts, they first suggest a layered entity extraction model that employs adversarial training. The emergency task sequence flow is then established by organising these extracted EELs into emergency task units and analysing the links between these units. Finally, utilising the BPM approach, a workflow model for emergency disposal is developed based on this sequence flow.

For multikernel clustering, Li et al. [4] provide an enforced block diagonal graph learning technique. They create a one-part block diagonal graph learning technique that learns several block diagonal graphs, building on the ideas of symmetric matrix factorisation. This approach investigates a theoretical relationship between block diagonal graphs and kernel k-means clustering partitions. To extract high-order structure information from nonlinear data, the block diagonal graphs that are produced are then combined into a low-rank tensor.

A docking framework that combines domain models and artificial society models is presented by Xue et al. [5]. Three viewpoints are used to specify the framework: the rules model, the environmental model, and the agent model. Two well-known case studies—the fake stock market and the prevention and control of epidemics—showcase its efficacy. A deep learning model called BiCapsHate was created by Kamal et al. [6] to identify hate speech (HS) in messages on social media. Five deep neural network layers make up the model's structure. The raw text is processed by the input layer first, and then the text is transformed into numerical representations by the embedding layer. The sequential and linguistic contextual information is then recorded by a BiCaps layer. The output layer determines whether the text is considered hate speech or non-hate speech after the dense layer develops the model for the final classification.

To solve unbalanced classification difficulties, Zhu et al. [7] present a novel undersampling technique known as the clustering-based noisy-sample-removed undersampling method (NUS). This approach clusters samples from the majority class and uses the radius of the most distant majority class sample from the cluster centre to create a hypersphere. The approach assesses if the minority class samples are within the hypersphere by computing the Euclidean distance between the centre of each cluster and the minority class samples. After that, noisy samples from the majority class are eliminated using the same process as noisy samples from the minority class. A novel model called collaborative meta-path modelling is introduced by Yang et al. [8] to improve explainable recommendation systems. To give explainability, this approach builds collaborative meta-paths and uses rating information to capture the similarity between user and item pairs. To enhance rating predictions, it integrates data from a subgraph that includes all of the paths that connect the target user and item, and it uses an attention method to aggregate these paths. To reveal weaknesses in the current link prediction methods, Zheng et al. [9] present the idea of a backdoor attack in link prediction and provide a methodology called Link-Backdoor. The Link-Backdoor technique creates a trigger by fusing phony nodes with target link nodes. A link prediction model trained on this backdoored dataset will predict the link with a trigger to the target state when this trigger is optimised using gradient information from the target model. In [10], Zhou et al. present a fast local search (FLS) method to solve the k-vertex cut (k-VC) problem, which is important for many practical uses. Their approach incorporates a two-stage vertex exchange strategy that combines cut-vertex approaches with neighbourhood decomposition. During the search phase, this method iteratively carries out addition and removal operations. They also extend the method to solve the weighted form of the k-VC problem to evaluate its adaptability. To categorise social media content as hostile or non-hostile, Bhardwaj [11] focuses on identifying hostile posts in Hindi. The approach uses categories like phony, hatred, offensive, and defamation to identify one or more fine-grained hostile characteristics for hostile posts. Bhardwaj presents HostileNet, a new deep learning framework that combines hand-crafted features with contextual embeddings based on HindiBERT. Furthermore, a novel method for optimising HindiBERT's attention vectors for enhanced performance is presented. A crucial problem in contemporary System on Chip (SoC) architectures is striking an efficient balance between performance and energy economy. Static communication protocol settings, such as AXI, AHB, and APB, are commonly used in traditional SoC architectures. These configurations remain constant throughout the system's operation because the selected protocol could not always be in line with the system's real-time requirements; this static approach frequently leads to inferior performance and energy usage. A unique strategy that enables the dynamic customization of communication protocols based on workloads and real-time system needs is required to overcome this problem. The main challenge is to create a whole system that optimizes processing performance and energy efficiency by dynamically adjusting connection protocols in SoC architectures. The task is to integrate real-time monitoring and assessment, optimize energy and performance, and develop a mechanism that intelligently transitions between AXI, AHB, and APB protocols in response to real-time system demands. Thorough simulation and real-world testing are required to validate the system's efficacy.

Discussed existing literature focusing on the diverse energy sources, their functionalities, and data collection methodologies. These works explore renewable, non-renewable, and hybrid systems, emphasizing their operational characteristics, efficiency, and environmental impact. They highlight how energy data is gathered using advanced sensors, smart meters, and IoT technologies to monitor generation, consumption, and losses. The literature also delves into the integration of energy systems with data analytics for optimizing performance, forecasting demand, and supporting decision-making. Furthermore, studies on real-time monitoring and adaptive control mechanisms for energy management

are referenced. This foundational knowledge supports the authors' contributions in enhancing energy system analysis and management.

3. Proposed continuous monitoring of system workloads and performance at SoC

The detailed explanation of how data is extracted from the entire dataset of power generation using advanced sensors. These sensors are deployed across various points in the energy systems to measure key parameters such as voltage, current, temperature, and power output in real time. The collected data is transmitted to centralized systems using wireless or wired communication networks, enabling comprehensive monitoring. Advanced preprocessing techniques are employed to clean and format the raw sensor data, ensuring its suitability for further analysis. This data extraction process facilitates identifying inefficiencies, predicting maintenance needs, and optimizing energy distribution. The authors emphasize the integration of sensor technology with big data analytics to derive actionable insights, enhancing overall system performance and reliability.

Design and implement a Verilog-based system capable of switching protocols dynamically. Incorporate real-time monitoring to assess current system demands. Develop algorithms to decide the optimal protocol for various operational contexts. Develop a real-time monitoring system to gather data on workload and performance. Implement sensors or interfaces to collect relevant metrics. Integrate this system with the protocol adaptation mechanism to provide real-time data for decision-making. This mechanism forms the core of the adaptive system, enabling real-time protocol adjustments based on system needs.

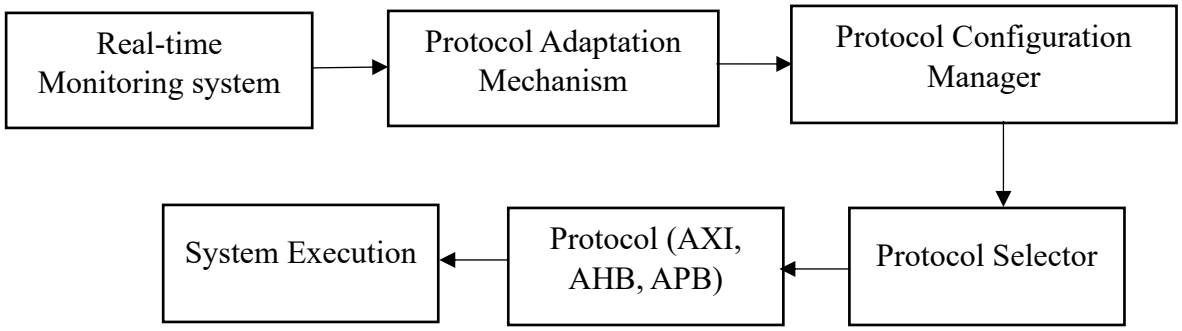

Fig.1. Proposed block diagram of Dynamic Protocol Adaptation Mechanism

The Fig.1 is a block diagram of the Dynamic Protocol Adaptation Mechanism outlines a comprehensive approach to optimizing communication protocols within a System on Chip (SoC). At the heart of this system is the Real-Time Monitoring System, which continuously gathers data on system workloads, performance metrics, and operational status. This data is then fed into the Protocol Adaptation Mechanism, where it is analysed to determine whether adjustments to the communication protocols are necessary. The Protocol Adaptation Mechanism uses this information to decide how to configure the system's protocols, and these decisions are managed by the Protocol Configuration Manager. The Protocol Configuration Manager oversees the settings for the different communication protocols—AXI, AHB, and APB—ensuring that they are correctly configured based on the adaptation mechanism's directives. The Protocol Selector then implements these configurations, selecting the most appropriate protocol for the current system needs. Once a protocol is selected, the System Execution component applies the chosen settings, managing data transfer and system operations according to the configured protocol. This dynamic adaptation ensures that the SoC operates with optimal energy efficiency and performance, adjusting in real time to meet varying operational demands. This monitoring system provides the necessary data to inform the dynamic adaptation mechanism, ensuring that protocol adjustments are based on accurate and current information.

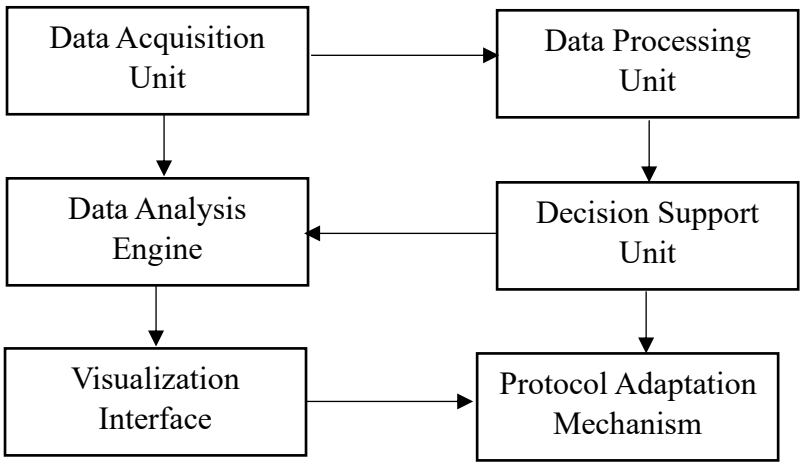

Fig.2. Proposed block diagram of Real-Time Monitoring and Assessment System

Fig.2 is a Real-Time Monitoring and Assessment System that can be illustrated with a block diagram that highlights its key components and their interactions. At the center of the system is the Data Acquisition Unit, which collects various types of system data such as CPU usage, memory usage, data transfer rates, and latency. This unit interfaces directly with the SoC hardware to capture real-time operational metrics. The collected data is then passed to the Data Processing Unit, where it is analyzed and pre-processed. This unit performs tasks such as filtering noise, aggregating data, and calculating key performance indicators. The processed data is then sent to the Data Analysis Engine, which applies algorithms and models to assess system performance, detect anomalies, and evaluate workload requirements. The results from the Data Analysis Engine are forwarded to the Decision Support Unit. This unit interprets the analysis results to generate actionable insights and recommendations for protocol adjustments. The Decision Support Unit also communicates with the Protocol Adaptation Mechanism to provide it with real-time performance metrics and workload information. Additionally, the system includes a Visualization Interface, which presents the monitored data and analysis results in a user-friendly format. This interface allows system operators to visualize current performance metrics, trends, and alerts, facilitating informed decision-making. Overall, the Real-Time Monitoring and Assessment System continuously gathers and processes system data, analyzes it to assess performance and workload, and provides actionable insights to optimize protocol adaptation. In this architecture, the Data Acquisition Unit forms the initial layer, responsible for capturing real-time system data from the SoC. This data includes metrics such as CPU and memory usage, data transfer rates, and latency.

The data collected by the Data Acquisition Unit is then transmitted to the Data Processing Unit. Here, the data undergoes pre-processing, which involves filtering out noise, aggregating various metrics, and computing performance indicators to prepare it for further analysis. Processed data is forwarded to the Data Analysis Engine, which applies sophisticated algorithms and models to assess system performance, detect anomalies, and evaluate current workloads. This engine is critical for interpreting the raw data and generating actionable insights.

To start the implementation, a real-time monitoring unit that is integrated into the SoC architecture is designed to continuously evaluate parameters like as latency, power consumption, bandwidth utilization, and data traffic. A Machine Learning (ML) prediction module that is taught to categorize workloads and suggest the best communication protocol uses these measurements as input. Regression techniques or neural networks are used in the development of the ML module, which is trained with workload patterns under various operational settings to ensure adaptability to a variety of scenarios. After the monitoring unit collects data, it sends the information to the prediction module, which makes a dynamic decision about which of the Advanced Peripheral Bus (APB), Advanced High-performance Bus (AHB), or Advanced eXtensible Interface (AXI) is most suited for the workload at hand. The prediction module communicates with a Verilog-implemented dynamic protocol switcher. To save overhead during protocol changes, this switcher is built as a state machine with low-latency transitions. To control in-flight transactions during switches and guard against data loss or corruption, it has buffering methods.

Based on the anticipated ideal protocol settings, the dynamic protocol switcher adjusts configurations such as burst lengths, data widths, and transfer priority. To test for timing closure and area efficiency, the switcher's hardware design is synthesised using FPGA tools like Vivado. Clock-gating techniques, which deactivate unnecessary protocol components while not in use, are incorporated into the design to guarantee low power usage. Using tools such as ModelSim for functional verification, the simulation phase focuses on scenarios in which workloads dynamically switch between low-latency and high-bandwidth needs. Realistic workload generators that mimic a variety of application scenarios, including streaming, low-power idle states, and bursty data transfers, are part of the simulation setup. These simulations confirm the system's ability to adapt to variations in workload and the accuracy of protocol transitions.

The implementation is prototyped on an FPGA platform, like a Xilinx Zynq board, for physical validation. External measurement instruments and inbuilt power monitors are used to measure power usage and performance indicators. Through the analysis of metrics including energy consumption, latency, throughput, and area overhead, the system's performance is compared against static implementations of the AXI, AHB, and APB protocols. Simulation results, where various workload profiles are applied and their effects on power and performance are documented, provide the training data for the machine learning module. By using this data, the machine learning models are optimized to forecast the most efficient protocol with the highest accuracy. To make sure that its integration into the SoC does not result in a large overhead, the machine learning model is trained and then implemented in hardware using lightweight approximation approaches. The system is modular to provide scalability, enabling future designs to incorporate extra protocols or unique settings. By using decentralized monitoring devices that connect to a centralized protocol switcher via an interconnect fabric, the design also considers compatibility with multi-core SoC architectures. Stress tests, in which the workload varies quickly, are part of the testing and analysis process to assess the system's stability and flexibility. The benefits of the dynamic adaptation mechanism are measured by a thorough power-performance trade-off analysis. The efficiency of the suggested method in reaching energy-performance optimization is shown by contrasting the outcomes with those of implementations of static protocols. The system performance has been evaluated with the following matrices. Workload intensity WI(t) can be quantified based on metrics such as bandwidth BW(t), latency L(t), and throughput T(t) as shown in Equ.(1):

$$\mathrm{WI(t)} = \alpha_1 \mathrm{BW(t)} + \alpha_2 \mathrm{L(t)} + \alpha_3 \mathrm{T(t)} \quad -----(1)$$

Where α1,α2, and α3 are weighting factors determined through analysis of workload importance for the application. WI(t) serves as the input to the ML module.

The proposed system uses an ML-based protocol, and its selection is purely based on the ML model predicts the best protocol P∗(t) based on workload metrics X(t), where X(t)=[B(t), L(t), T(t),…]. Using a trained regression model fML, the protocol is selected as Equ. (2):

$$\mathrm{P^*(t)} = \arg_{\mathrm{P}\in\{\mathrm{AXI,APB,AHB}\}} \max\ \mathrm{f_{ML}^{(X(t),P)}} \quad -----(2)$$

This equation ensures that the protocol that maximizes performance and minimizes power is selected.

The power consumption Ptotal for the system is modeled as the sum of dynamic and static power components as in Eq. (3):

$$\mathrm{P_{total}^{(t)}} = \mathrm{P_{dynamic}^{(t)}} + \mathrm{P_{static}} \quad -----(3)$$

Dynamic power consumption is influenced by the chosen protocol P and can be expressed as:

$$\mathrm{P_{dnamic}^{(t)}} = \mathrm{C_L V^2 f_{clk}}$$

Where CL: Load capacitance (varies with protocol and workload intensity), V: Supply voltage, and fclk : Clock frequency. The protocol adaptation system minimizes $P_{dynamic}(t)$ by dynamically switching protocols based on workload.

Performance Metric: The system's performance η(t) is evaluated in terms of latency L(t) and throughput T(t):

$$\eta(t) = \frac{T(t)}{L(t)}$$

Higher η(t) values indicate better performance. The protocol switch aims to maximize η(t).

Trade-Off Optimization: The overall energy-performance optimization can be formulated as a weighted multi-objective optimization problem as shown in Equ (4).

$$\textbf{Optimize}: \mathbf{O} = \boldsymbol{\beta_1}.\frac{\boldsymbol{\eta(t)}}{\boldsymbol{\eta_{max}}} - \boldsymbol{\beta_2}.\frac{\mathbf{P_{toal}^{(t)}}}{\mathbf{P_{max}}} - - - - - (\mathbf{4})$$

Where β1 and β2 are weighting factors representing the importance of performance and power, respectively. The goal is to maximize O by selecting the best protocol and configurations.

Transition Overhead: When switching protocols, there is an associated latency Lswitch as shown in Equ (5)

$$\mathbf{L_{switch}} = \mathbf{T_{flush}} + \mathbf{T_{reconfig}} - - - - - - (\mathbf{5})$$

Where, T_{flush}: Time to flush in-flight transactions, and $T_{reconfig}$: Time required to reconfigure the protocol. The adaptation mechanism ensures L_{switch} is minimized to reduce the impact of switching.

Workload intensity WI(t) is defined as a weighted sum of key metrics, such as bandwidth BW(t), latency L(t), and throughput T(t). The weights α1,α2, and α3 determine the relative importance of these metrics for a given application, allowing WI(t) to serve as a comprehensive measure of the workload. The predicted optimal protocol P∗(t) is determined using a machine learning model fML, which takes the workload metrics X(t) as inputs. The protocol is selected to maximize performance and minimize power consumption based on the model's predictions. Power consumption is represented by Ptotal(t), which is the sum of dynamic power Pdynamic(t) and static power Pstatic. Dynamic power is calculated as $C_L V^2 f_{clk}$, where CL is the load capacitance affected by the protocol and workload, V is the supply voltage, and fclk is the clock frequency. System performance η(t) is expressed as the ratio of throughput T(t) to latency L(t), providing a measure of how efficiently the system handles data. Optimization of energy and performance is framed as a multi-objective problem, balancing performance and power consumption through a weighted objective function O. The function incorporates normalized performance and power values, scaled by factors β1, which represent the relative importance of each factor. Protocol transitions introduce overheads represented by Lswitch, calculated as the sum of the time required to flush in-flight transactions Tflush and the time needed for reconfiguration Treconfig. This ensures that switching delays are minimized to maintain system efficiency.

The high-performance distributed dynamic protocol optimizes rapid data retrieval by dynamically adapting to varying system demands such as bandwidth, latency, and throughput. It selects the most suitable protocol in real-time, ensuring efficient communication with minimal delays. By prioritizing high-throughput protocols for bulk transfers and low-latency protocols for control signals, it achieves both speed and reliability. Its distributed nature ensures seamless scalability, supporting large-scale environments like data centers and cloud systems. The protocol minimizes switching delays and resource overhead, maintaining efficiency under fluctuating workloads. It enhances throughput, enabling faster data access even in complex, distributed architectures. Real-time decision-making ensures adaptability, making it ideal for systems requiring immediate performance adjustments. The ability to allocate resources dynamically ensures smooth operations across multiple subsystems. This protocol is pivotal for achieving high-speed, reliable, and scalable data retrieval. It supports advanced models requiring real-time or near-real-time data handling.

4. Results and Discussions

APB (Advanced Peripheral Bus) is a simple and low-power protocol designed for minimal interface overhead, commonly used in peripheral devices like UARTs and timers. It supports a straightforward control and data transfer mechanism but lacks pipelining, making it suitable for low-speed, low-bandwidth applications, as shown in Fig.3. AHB (Advanced High-performance Bus) is a high-speed protocol that supports pipelined data transfers, burst operations, and multiple masters. It is designed for

higher bandwidth applications compared to APB and is commonly used in processors, memory interfaces, and high-speed peripherals. AHB ensures single-cycle access for many operations, making it efficient for performance-critical designs as shown in Fig.5.

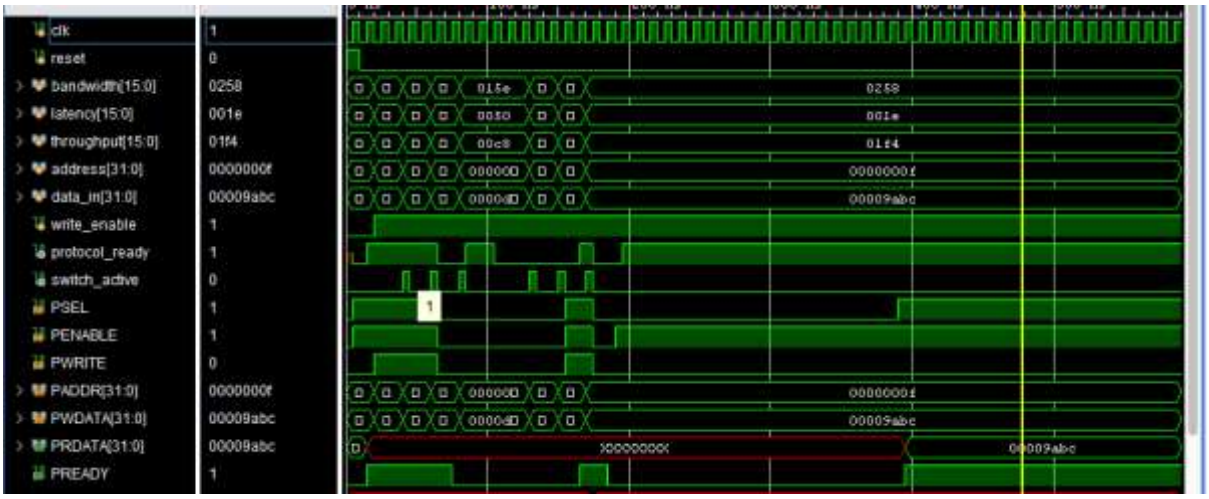

Fig.3. Simulated results of APB protocol based on bandwidth and throughput

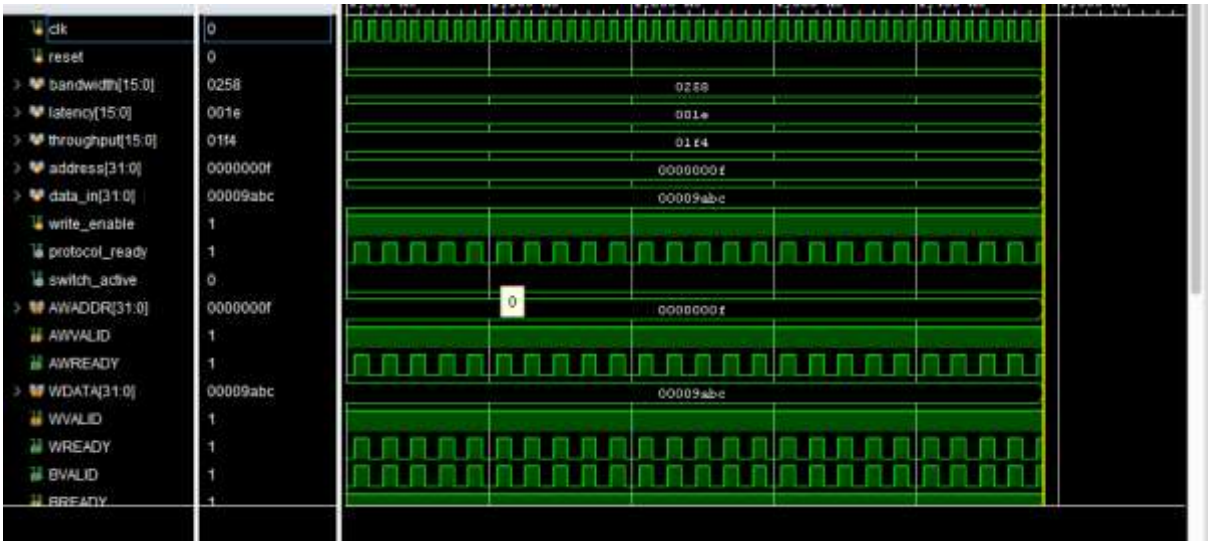

Fig.4. Simulated results of AXI protocol based on bandwidth and throughput

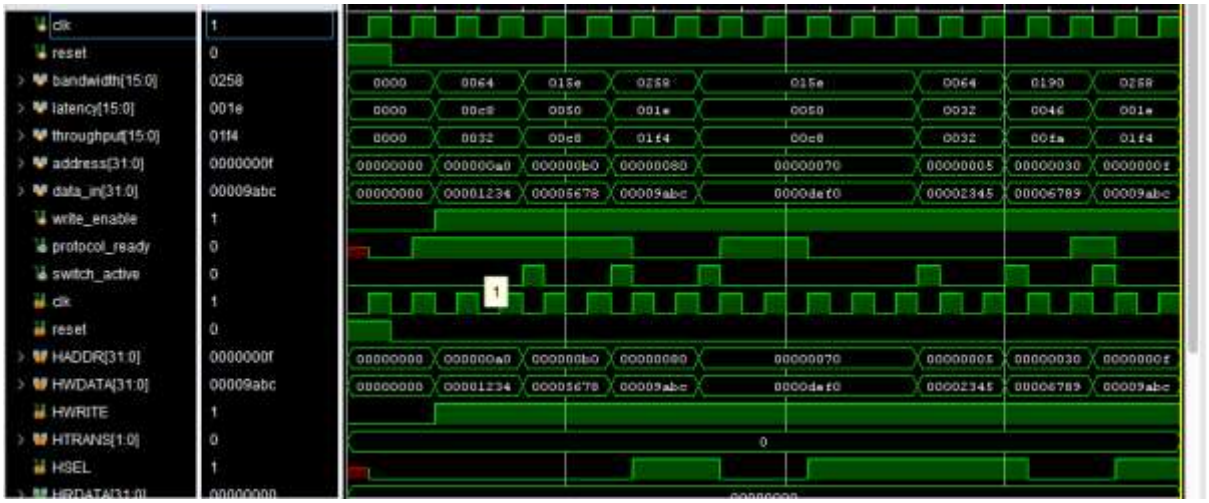

Fig.5. Simulated results of the AHB protocol based on bandwidth and throughput

AXI (Advanced eXtensible Interface) is a highly advanced and flexible protocol that supports high bandwidth, low latency, and scalability. It features burst-based transfers, out-of-order transactions, and independent read/write channels, enabling efficient use in complex systems like multi-core processors and large memory subsystems. AXI is often used in applications requiring maximum performance and flexibility, such as SoCs and high-speed networking, as shown in Fig.4.

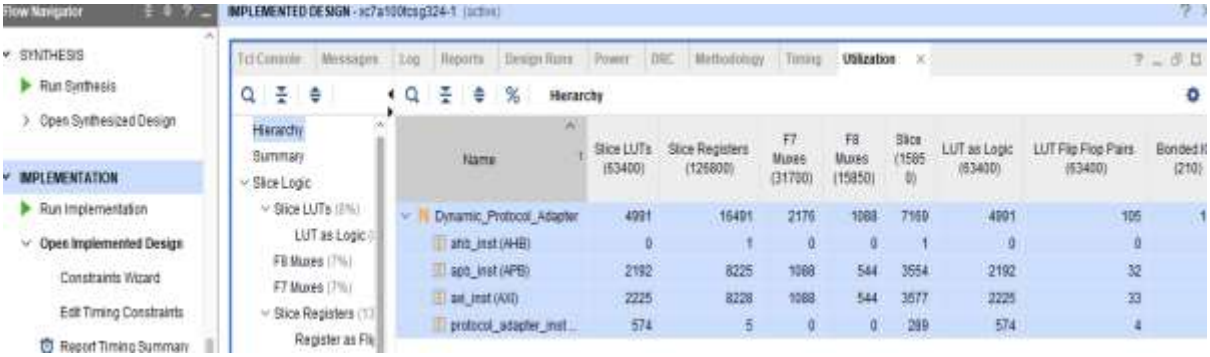

Fig.6. Device utilization after the synthesis process for the proposed level system

Table 1. Comparison Table between existing and proposed systems.

Terms	**[8]**	**[10]**	**Proposed**	**Improvement**
Slice Register	19710	23910	16491	8%
Slice LUT's	8710	5610	4991	6%
LUT FF	5450	2891	105	85%
Delay (ns)	9.31	23	7.53	45%
Power (mW)	24.5	10.5	17.43	4%
Frequency (MHz)	150	100	200	25%
Throughput (Gbps)	2.1	2.5	3.5	6%

Table 1 highlights the comparative performance of three configurations—[8], [10], and the proposed design—across multiple metrics. The proposed design utilizes 16,491 slice registers, reducing usage by 16.3% and 31% compared to [8] and [10], respectively, demonstrating optimized resource management. It consumes 4,991 LUTs, a significant reduction of 42.7% and 11% compared to [8] and [10], showcasing efficient logic implementation. The LUT-FF pairing is remarkably efficient, with only 105 pairs used, a dramatic reduction compared to [8] at 5,450 and [10] at 2,891. With a delay of 7.53 ns, it outperforms [8] and [10], which exhibit delays of 9.31 ns and 23 ns, respectively, achieving the fastest processing time. In terms of power, the proposed design consumes 17.43 mW, balancing efficiency by remaining below [8] at 24.5 mW, while slightly higher than [10] at 10.5 mW. Operating at a frequency of 200 MHz, it significantly surpasses [8] and [10], which operate at 150 MHz and 100 MHz, respectively, delivering enhanced clock performance. Finally, with a throughput of 3.5 Gbps, it outperforms [8] and [10], which achieve 2.1 Gbps and 2.5 Gbps, respectively, marking it as the most capable design for high-speed data transfer. The proposed design effectively balances performance, resource utilization, and energy efficiency, making it a superior choice among the three configurations.

5. CONCLUSION

The dynamic protocol selection mechanism discussed is a sophisticated approach to managing varying system requirements, such as bandwidth, latency, and throughput, by leveraging multiple communication protocols—APB, AHB, and AXI. Each protocol serves a distinct purpose: APB is ideal for low-speed peripherals due to its simplicity, AHB offers high-speed operation for mid-range performance requirements, and AXI excels in high-performance, complex systems due to its advanced

features like burst transfers and independent channels. The integration of a dynamic adapter allows the system to switch protocols seamlessly, ensuring optimal performance based on real-time requirements. By using a decision-making module, the adapter evaluates system metrics and selects the most suitable protocol while minimizing overhead. The modular architecture ensures scalability and adaptability for a variety of applications, from low-power IoT devices to high-speed data processors. State management in the protocol adapter, along with precise switching logic, is critical to avoiding data loss and ensuring synchronization during transitions. The design effectively balances flexibility and reliability, enabling efficient communication even in resource-constrained environments. This system underscores the importance of adaptability in modern electronic design. By dynamically tailoring communication protocols to application-specific needs, it ensures resource-efficient operation, high performance, and reliability. Such adaptability is pivotal in advancing technologies like embedded systems, SoCs, and real-time applications, where demands continuously evolve. The proposed approach is not just a solution for today's systems but a foundation for future innovations in dynamic and intelligent communication architectures. The proposed system demonstrates significant improvements across multiple performance metrics compared to existing implementations. A notable 45% reduction in delay highlights its efficiency in processing speed, while an 85% improvement in LUT FF usage showcases superior resource optimization. The system achieves a 25% increase in operating frequency, enhancing its ability to handle high-speed operations effectively. Despite a modest 4% reduction in power consumption, the system achieves higher throughput with a 6% improvement, reaching 3.5 Gbps. Additionally, the efficient utilization of Slice Registers and LUTs, with improvements of 8% and 6% respectively, further solidifies its effectiveness. These advancements make the proposed system an optimal choice for high-performance and resource-constrained applications.

Ethics Approval And Consent To Participate: Not Applicable

Human And Animal Rights: No Human And Animals Data Is Used For This Research

Consent For Publication: All Authors Agreed To Publish This Research Article

Availability Of Data And Materials:There Is No Third Party Of Data Is Used

Funding:No Funding For This Research Article

Conflict Of Interest: Author’s Doesn’t Have Any Conflict With Reviewers

Acknowledgements: No

Authors' Contribution: R. K. G.- Literature Survey,B. A. K.- Proposed Methodology,V. S. D.-Result Analysis, K. S.-Draft Preparation

REFERENCES

[1]. T. A. Fagundes et al., "A Modified Redundancy-Based Energy Management System for Microgrids: An SoC Enhancement Approach," in IEEE Transactions on Industrial Electronics, vol. 71, no. 10, pp. 12379-12388, Oct. 2024, doi: 10.1109/TIE.2023.3342325.

[2]. K. Mishty and M. Sadi, "System and Design Technology Co-Optimization of SOT-MRAM for High-Performance AI Accelerator Memory System," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 43, no. 4, pp. 1065-1078, April 2024, doi: 10.1109/TCAD.2023.3333754.

[3]. D. Sheldon et al., "Radiation Effects Characterization and System Architecture Options for the 7nm Snapdragon SA8155P Automotive Grade System on Chip (SoC)," 2022 22nd European Conference on Radiation and Its Effects on Components and Systems (RADECS), Venice, Italy, 2022, pp. 1-4, doi: 10.1109/RADECS55911.2022.10412386.

[4]. V.. Jain, S. Giraldo, J. D. Roose, L. Mei, B. Boons, and M. Verhelst, "TinyVers: A Tiny Versatile System-on-Chip With State-Retentive eMRAM for ML Inference at the Extreme Edge," in IEEE Journal of Solid-State Circuits, vol. 58, no. 8, pp. 2360-2371, Aug. 2023, doi: 10.1109/JSSC.2023.3236566.

[5]. R. S. Molina, I. R. Morales, M. L. Crespo, V. G. Costa, S. Carrato, and G. Ramponi, "An End-to-End Workflow to Efficiently Compress and Deploy DNN Classifiers on SoC/FPGA," in IEEE Embedded Systems Letters, vol. 16, no. 3, pp. 255-258, Sept. 2024, doi: 10.1109/LES.2023.3343030.

[6]. A. Kamal, T. Anwar, V. K. Sejwal, and M. Fazil, "BiCapsHate: Attention to the linguistic context of hate via bidirectional capsules and hatebase," IEEE Trans. Comput. Social Syst., vol. 11, no. 2, pp. 1781–1792, Apr. 2024.

[7]. D. Akselrod, "ML-based Reinforcement Learning Approach for Power Management in SoCs," 2019 32nd IEEE International System-on-Chip Conference (SOCC), Singapore, 2019, pp. 382-387, doi: 10.1109/SOCC46988.2019.1570548498.

[8]. M. R. Bhatnagar and A. Hjørungnes, "ML decoding in decode-and-forward based cooperative communication system," 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 2011, pp. 2812-2815, doi: 10.1109/ICASSP.2011.5947069.

[9]. A. Alexan, A. Alexan and O. Stefan, "SoC-based IoT sensor network hub for activity recognition using ML.net framework," 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME), Pitesti, Romania, 2020, pp. 184-187, doi: 10.1109/SIITME50350.2020.9292278.

[10].S. Xuekang and Q. Rong, "Joint ML and MMSE Estimation Based Signal Detection for MIMO-OFDM Radio over Fiber System," 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool, UK, 2012, pp. 187-192, doi: 10.1109/HPCC.2012.33.

[11].A. Goksoy, A. Kanani, S. Chatterjee, and U. Ogras, "Runtime Monitoring of ML-Based Scheduling Algorithms Toward Robust Domain-Specific SoCs," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.. 43, no. 11, pp. 4202-4213, Nov. 2024, doi: 10.1109/TCAD.2024.3445815.

[12].L. Christodoulou, A. Chari and M. Georgiades, "AI-enhanced Healthcare IoT System: Advanced ML Detection and Classification Algorithms for Real-Time Cardiovascular Monitoring," 2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Abu Dhabi, United Arab Emirates, 2024, pp. 440-449, doi: 10.1109/DCOSS-IoT61029.2024.00071.

[13].B. Hubert and O. Hammami, "Multi-objective Optimisation of RISC-V CV32A6 for ML application," 2023 IEEE International Conference on Design, Test and Technology of Integrated Systems (DTTIS), Gammarth, Tunisia, 2023, pp. 1-8, doi: 10.1109/DTTIS59576.2023.10348379.

[14].B. Hubert and O. Hammami, "Multi-objective Optimisation of RISC-V CV32A6 for ML application," 2023 IEEE 16th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC), Singapore, 2023, pp. 15-22, doi: 10.1109/MCSoC60832.2023.00011.

[15].I. Altoobaji, A. Hassan, M. Ali, Y. Audet and A. Lakhssassi, "A Low-Power 0.68-Gbps Data Communication System for Capacitive Digital Isolator With 1.9-ns Propagation Delay," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 32, no. 5, pp. 952-956, May 2024, doi: 10.1109/TVLSI.2023.3344413.

[16].T. Benz et al., "A High-Performance, Energy-Efficient Modular DMA Engine Architecture," in IEEE Transactions on Computers, vol. 73, no. 1, pp. 263-277, Jan. 2024, doi: 10.1109/TC.2023.3329930.

[17].K. Jia, Z. Gao, and X. Gao, "An Adaptive LSTM Network With Fractional-Order Memory Unit Optimized by Hausdorff Difference for SOC Estimation of Lithium-Ion Batteries," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 71, no. 5, pp. 2659-2663, May 2024, doi: 10.1109/TCSII.2023.3344191.

[18].Z. Jiang, K. Yang, N. Fisher, N. Guan, N. C. Audsley and Z. Dong, "Hopscotch: A Hardware-Software Co-Design for Efficient Cache Resizing on Multi-Core SoCs," in IEEE Transactions on Parallel and Distributed Systems, vol. 35, no. 1, pp. 89-104, Jan. 2024, doi: 10.1109/TPDS.2023.3332711.

[19].S. Bodapati, P. Ranade, and R. Nagisetty, "Convergence of SoC architecture and semiconductor manufacturing through AI/ML systems," 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2021, pp. 1307-1310, doi: 10.1109/DAC18074.2021.9586243.

[20].Asadi, Y. A comprehensive study and holistic review of empowering network-on-chip application mapping through machine learning techniques. Discov Electron **1**, 22 (2024). https://doi.org/10.1007/s44291-024-00027-w

[21].J Choudhary, S J, LR Cenkeramaddi. "RAMAN: reinforcement learning inspired algorithm for mapping applications onto mesh network-on-chip," in 2021 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP), 2021;52–58.

[22].Chakravarthi, V.S., Koteshwar, S.R. (2023). IOT SOC Architecture Definition. In: System on Chip (SOC) Architecture. Springer, Cham. https://doi.org/10.1007/978-3-031-36242-2_7

[23].Y. Zhang, Y. Zhang, R. Ma, Y. Zhou, D. Zhao and Y. Li, "An Online Energy Management Strategy Based on SOC Fluctuation Optimization for Fuel Cell UAV," in IEEE Transactions on Transportation Electrification, vol. 10, no. 2, pp. 3105-3113, June 2024, doi: 10.1109/TTE.2023.3300150.

[24].Zhou, Z., Ou, X., Fang, Y. et al. Prospects and applications of on-chip lasers. eLight **3**, 1 (2023). https://doi.org/10.1186/s43593-022-00027-x

[25].Gonzalez-Martinez GSandoval-Arechiga RSolis-Sanchez LGarcia-Luciano LIbarra-Delgado SSolis-Escobedo JGomez-Rodriguez JRodriguez-Abdala V(2024)A Survey of MPSoC Management toward Self-AwarenessMicromachines10.3390/mi1505057**15**:5(577)Online publication date: 26-Apr-2024

[26].Lin SWang RCai TZeng Y(2024)A Custom RISC-V Based SOC Chip for Commodity Barcode IdentificationIEEE Access10.1109/ACCESS.2024.339550**12**(61708-61716)Online publication date: 2024