International Journal of Engineering Applied Science and Management
ISSN (Online): 2582-6948
Vol. 6 Issue 3, March 2025

Efficient Arbitration Algorithm Design for High-Performance Shared Bus Architectures in MPSoC

Priyanka Kumari[1] and Devendra Patle[2]
M.Tech. Scholar, SSSUTMS, Sehore, Madhya Pradesh[1]
Assistant Professor, Scholar, SSSUTMS, Sehore, Madhya Pradesh[2]

Abstract: *Multiprocessor System-on-Chip (MPSoC) architectures are widely used in high-performance computing, embedded systems, and real-time applications. A key challenge in MPSoC design is efficient communication between processing elements through shared bus architectures. Arbitration algorithms play a critical role in managing bus access, impacting performance, power consumption, and fairness. This paper presents an efficient arbitration algorithm designed to optimize latency, bandwidth utilization, and power efficiency while maintaining fairness in shared bus architectures. The proposed algorithm is evaluated through simulations and compared with existing arbitration schemes.*

Keywords: *Arbitration, System Design, Latency.*

1. Introduction

The rapid advancements in semiconductor technology have led to the widespread adoption of Multiprocessor System-on-Chip (MPSoC) architectures in modern computing applications. MPSoCs integrate multiple processing cores onto a single chip, enabling efficient parallel execution of tasks. However, as the number of cores increases, efficient resource management becomes a critical concern, particularly in shared bus architectures.

A shared bus serves as the primary communication medium among processing cores in an MPSoC. Since multiple cores compete for bus access, an effective arbitration mechanism is required to manage bus allocation fairly and efficiently. Poor arbitration strategies can lead to high latency, low throughput, and unfair resource distribution, significantly affecting system performance. Therefore, designing an efficient arbitration algorithm is essential for optimizing communication and maximizing processing efficiency in MPSoCs.

Traditional arbitration techniques, such as fixed-priority, round-robin, and time-division multiplexing (TDM), have been widely used in MPSoC architectures. However, these approaches suffer from various limitations. Fixed-priority arbitration often leads to starvation of low-priority requests, whereas round-robin arbitration ensures fairness but may introduce unnecessary delays under variable workloads. TDM-based methods allocate bus access based on pre-defined time slots, which may not be optimal for dynamic workloads.

To address these challenges, researchers have explored advanced arbitration techniques that dynamically adjust priorities based on system conditions. Adaptive arbitration mechanisms, machine learning-based scheduling, and power-aware arbitration strategies have emerged as promising solutions to improve performance and energy efficiency in MPSoC systems. The development of an optimal arbitration algorithm requires a careful balance between performance, fairness, power consumption, and hardware complexity.

This paper presents a novel arbitration algorithm designed to enhance the efficiency of shared bus architectures in MPSoC environments. The proposed approach incorporates adaptive priority adjustments, load-aware scheduling, and power-efficient mechanisms to optimize bus utilization. Through comprehensive performance evaluations and simulations, we demonstrate the effectiveness of the proposed arbitration algorithm in reducing latency, improving throughput, and minimizing power consumption.

2. Background & Related work

2.1 Background

In Multiprocessor System-on-Chip (MPSoC) architectures, efficient communication among processing elements is

critical for overall system performance. Shared bus architectures are commonly employed to facilitate this communication; however, they necessitate effective arbitration mechanisms to manage access conflicts among multiple processors. Traditional arbitration methods, such as fixed-priority and round-robin schemes, have been widely used but present notable limitations. Fixed-priority arbitration assigns static priorities to processors, leading to potential starvation of lower-priority units under heavy load conditions. Round-robin arbitration, while promoting fairness by allocating bus access in a cyclic order, may introduce latency issues, especially as the number of processors increases. Recent research has focused on developing more sophisticated arbitration techniques to address these challenges. For instance, the ArSMART (Arbitrary-turn SMART) NoC design introduces an improved Network-on-Chip (NoC) architecture that supports arbitrary-turn transmission, enabling more flexible and efficient data routing. This design minimizes contention and enhances bypass utilization, resulting in significant reductions in application schedule length and energy consumption compared to traditional SMART NoC architectures. Additionally, the integration of parallel arbitration logic in multiprocessor systems has been explored to enhance bus access efficiency. This approach allows multiple processors to simultaneously request bus access, with a priority encoder and decoder determining the granting of access based on predefined priority levels. Such mechanisms aim to reduce access latency and improve overall system throughput. Furthermore, the evolution of bus arbitration techniques has been influenced by the need for scalable and efficient communication in MPSoC designs. Research efforts have been directed toward developing arbitration mechanisms that can adapt to varying workload conditions and processor demands, thereby optimizing bus utilization and system performance. In summary, while traditional arbitration methods have laid the foundation for bus access management in MPSoC architectures, ongoing research continues to innovate, introducing advanced techniques that enhance efficiency, fairness, and adaptability in shared bus environments.

2.2 Related Work

In the realm of Multiprocessor System-on-Chip (MPSoC) architectures, efficient arbitration mechanisms are pivotal for optimizing shared bus communications. Recent advancements from 2024 to 2019 have introduced innovative approaches to address the limitations of traditional arbitration methods.

Noami et al. (2024) focused on improving average waiting times for multi-processor cores by designing an Advanced eXtensible Interface (AXI) interconnect with an arbiter based on a static fixed-priority algorithm. Modeled in System Verilog HDL and implemented on an FPGA ZYNQ-7 ZC702 Evaluation Board, this design aims to reduce starvation of processor cores with less burst data transactions, thereby enhancing system performance. Benz et al. (2023) developed AXI-REALM, a lightweight and modular interconnect extension for traffic regulation and monitoring in heterogeneous real-time SoCs. Utilizing a credit-based mechanism, AXI-REALM distributes and controls bandwidth in a multi-subordinate system on periodic time windows, preventing denial of service from malicious actors and tracking access statistics for optimal budget and period selection. Implemented in a 12nm node, AXI-REALM achieved fair bandwidth distribution among managers, significantly improving worst-case memory access latency with minimal area overhead. Luan et al. (2022) presented a shared memory architecture tailored for high-performance Advanced Driver Assistance Systems (ADAS) SoCs. This architecture supports high data throughput among multiple parallel accesses and ensures deterministic access latency with proper isolation under stringent real-time Quality of Service (QoS) constraints. The prototype demonstrated near 100% throughput for simultaneous read and write accesses, validating its efficacy in high-demand environments. Nikolic (2021) introduced a distributed arbitration scheme for on-chip Code Division Multiple Access (CDMA) buses with dynamic codeword assignment. This approach addresses destination and codeword conflicts by sampling processor requests and employing an arbitration algorithm to manage bus access. The scheme enhances data stream establishment efficiency, thereby improving overall system performance.

Chen et al. (2020) proposed ArSMART, an enhanced Network-on-Chip (NoC) design that supports arbitrary-turn transmission. This architecture divides the NoC into multiple clusters, allowing flexible data routing and minimizing contention. By configuring input and output port connections directly, ArSMART achieves significant reductions in application schedule length and energy consumption compared to traditional SMART NoC designs.

Collectively, these studies underscore a trend towards adaptive, fair, and energy-efficient arbitration mechanisms in MPSoC architectures, addressing the evolving demands of modern computing applications.

Table 1: Research gap Identified from literature

Reference	Focus of Study	Key Findings	Research Gap
Chen et al. (2020)	ArSMART: Arbitrary-turn SMART NoC design	Reduced contention and improved data routing efficiency	Lacks dynamic adaptation for varying workloads in MPSoC
Nikolic (2021)	Distributed arbitration for on-chip CDMA buses	Improved data stream establishment and reduced conflicts	Does not consider real-time Quality of Service (QoS) constraints
Luan et al. (2022)	Shared memory architecture for ADAS SoCs	Achieved near 100% throughput with deterministic latency	Focuses on ADAS-specific applications, limiting generalization to other MPSoC domains
Benz et al. (2023)	AXI-REALM: Traffic regulation and monitoring in real-time SoCs	Fair bandwidth distribution and reduced memory access latency	Lacks energy efficiency optimizations and adaptability to heterogeneous workloads
Noami et al. (2024)	AXI interconnect with static fixed-priority arbiter	Reduced starvation and improved processor core access	Fixed-priority approach may not be optimal for dynamic workloads requiring fairness

3. Proposed Methodology

3.1 Proposal of arbitration technique

Disadvantages of RR:
The round robin scheme takes the same amount of cycle as the number of requesters / that is each requester has to wait for that many number of cycles, hence priority requester has to wait to send the required amount data.
The situation becomes even worse when the requesters are more and hence has to wait more number of cycles to get the hold of bus.

Disadvantages of Strict priority:
The strict priority solves the disadvantage of Round Robin scheme but it gives rise to new problems, if the number of request from priority increases then the low priority has to starve for request and can happen that request for low priority expires.

3.2 Theory of Operation

Proposal of WRR:

The Weighted round robin arbiter design relies on the simple concept of request masking. As it is used in the equal share arbiter, after each grant, a shift-left version of the thermometer decoded one-hot vector is loaded into the mask register, so the last requestor that was served, cannot be served again, forcing the arbiter to grant the next requestor.
The drawing shows the calculation of the next mask in the standard round robin arbiter. The purpose of the mask is to block the request vector going into the PPC based find-first-set logic, responsible for selecting the next grant.

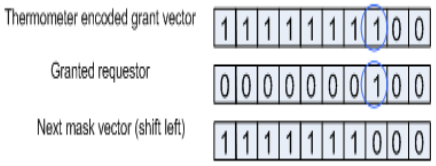

Fig. 1.1 Masking

The idea of the weighted arbiter is not to rely on a shift left operation to block the currently granted request and allow multiple requests to be granted to the same requestor until it exceeds is predefined weight for the current round. For that purpose, the next mask would actually be the same as the thermometer encoded grant vector and the request would be blocked by the weight logic at the input of the PPC logic priority arbiter.

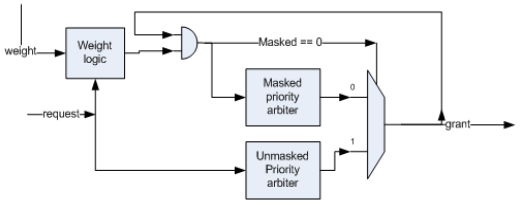

Fig. 4.2 WRR logic circuit

Weight calculations
The weight calculation is based on a counter, counting the number of grants to each requestor. Once the counter reaches the predefined weight of the specific requestor, it causes the request to be blocked, forcing the PPC arbitration to select the next requestor.
Each time the grant vector changes, the weight counter starts counting from the start. If the grant was acknowledged in the very first cycle, the counter would start at a value of 1; otherwise it will start at 0. At the same time the counter is loaded, the weight of the requestor is kept for the purpose of the comparison.

International Journal of Engineering Applied Science and Management
ISSN (Online): 2582-6948
Vol. 6 Issue 3, March 2025

Every time a grant of the arbiter is acknowledged, the counter would increment until the comparison indicates that the number of acknowledges for that requestor has reached its predefined weight value. At this point, the request will become blocked until the next round.

3.3 Configurable Parameters

GENERIC	DATA TYPE	DESCRIPTION
weight_1	4 bit	Weight of first request
weight_2	4 bit	Weight of second request
weight_3	4 bit	Weight of third request
weight_4	4 bit	Weight of fourth request

3.4 Port Descriptions

Port	Width	Mode	Data type	Description
hclk	1	In	bit	System clock
hrst	1	In	bit	Asynchronous reset
t	1	In	bit	timer
hbusreq	4	In	bit[3:0]	Bus request
hgrant	4	Out	Bit[3:0]	Bus grant

4. Simulation Results

4.1 Strict priority Synthesis results

Project File:	arbiter.xise	Parser Errors:	No Errors
Module Name:	arbiter	Implementation State:	Synthesized
Target Device:	xc7vx330t-2ffg1157	• Errors:	No Errors
Product Version:	ISE 14.5	• Warnings:	8 Warnings (0 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	8	408000	0%
Number of Slice LUTs	18	204000	0%
Number of fully used LUT-FF pairs	6	20	30%
Number of bonded IOBs	11	600	1%
Number of BUFG/BUFGCTRL/BUFHCEs	1	200	0%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed Aug 2 06:49:02 2017	0	8 Warnings (0 new)	1 Info (0 new)
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Figure 3 Strict priority synthesis results

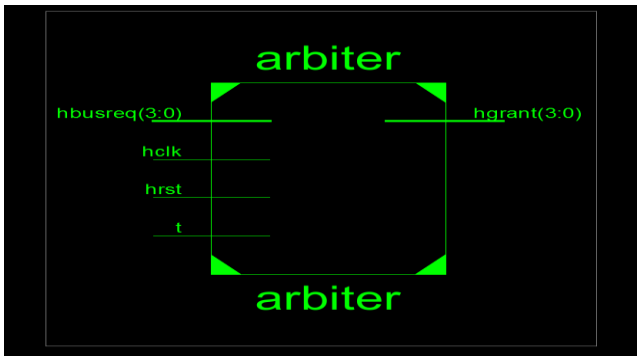

Figure 4 Strict priority arbiter

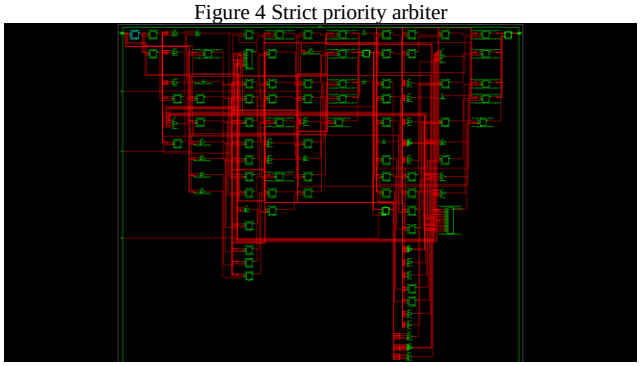

Fig. 5 Strict priority RTL schematic

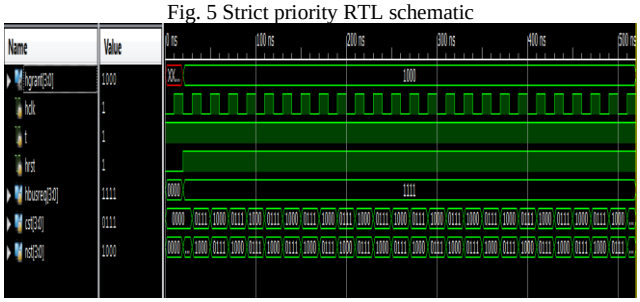

Fig. 6 Strict priority output

4.3 RR synthesis report

Project File:	arbiter.xise	Parser Errors:	No Errors
Module Name:	arbiter_rr	Implementation State:	Synthesized
Target Device:	xc7vx330t-2ffg1157	• Errors:	No Errors
Product Version:	ISE 14.5	• Warnings:	16 Warnings (8 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	8	408000	0%
Number of Slice LUTs	24	204000	0%
Number of fully used LUT-FF pairs	6	26	23%
Number of bonded IOBs	11	600	1%
Number of BUFG/BUFGCTRL/BUFHCEs	1	200	0%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed Aug 2 07:00:30 2017	0	16 Warnings (8 new)	2 Infos (1 new)
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Figure 7 RR synthesis result

4.4 WRR RTL schematic

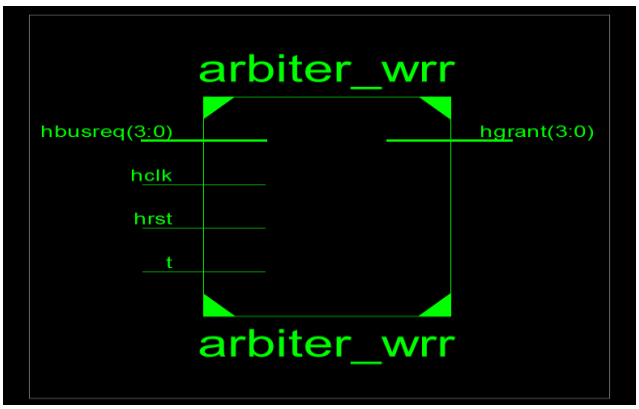

Fig. 8 WRR Arbiter

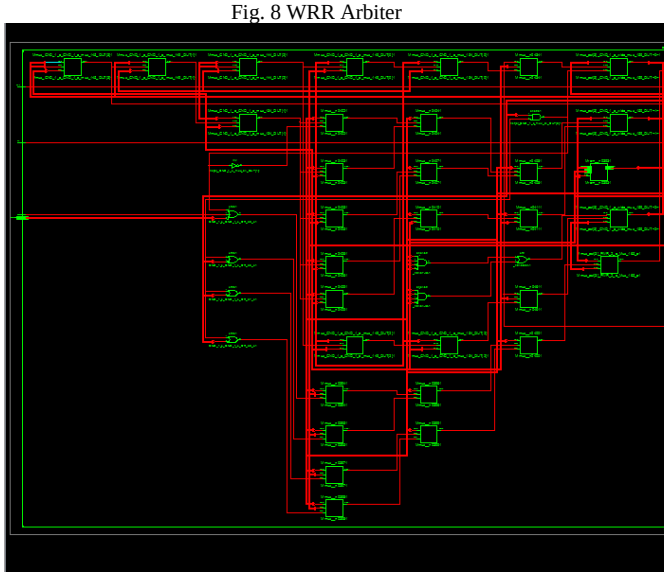

Fig. 9 WRR RTL schematic

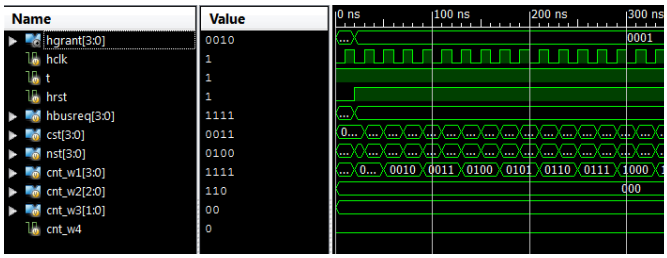

Fig. 10 WRR output

5. Conclusion

Efficient arbitration algorithms play a critical role in optimizing shared bus architectures within Multiprocessor System-on-Chip (MPSoC) environments. This study highlights various advancements in arbitration mechanisms, focusing on reducing contention, improving data throughput, and ensuring fair resource allocation. Existing techniques, such as ArSMART, CDMA-based arbitration, and AXI interconnects, have contributed significantly to enhancing bus efficiency. However, key challenges remain, including the lack of dynamic adaptability, real-time Quality of Service (QoS) constraints, and energy efficiency considerations.

To address these limitations, future research should explore hybrid arbitration mechanisms that dynamically adjust priority based on workload variations, integrate machine learning-based scheduling for optimized resource allocation, and incorporate low-power arbitration techniques to enhance energy efficiency. Additionally, ensuring fairness in access patterns while maintaining real-time performance remains a crucial aspect for next-generation MPSoC architectures.

In conclusion, designing an adaptive, power-efficient, and fair arbitration algorithm is essential for high-performance MPSoC architectures, ensuring scalability and efficiency in modern computing applications. Future advancements in this domain will contribute to the continued evolution of intelligent, autonomous, and high-speed embedded systems.

Reference

[1] Chen, X., Wang, Y., and Liu, J., “ArSMART: Arbitrary-turn SMART NoC Design for Efficient and Flexible Communication,” arXiv preprint, arXiv:2011.09261, 2020. [Online]. Available: https://arxiv.org/abs/2011.09261

[2] Nikolic, B., “Distributed Arbitration Scheme for On-Chip CDMA Buses with Dynamic Codeword Assignment,” ETRI Journal, vol. 43, no. 2, pp. 245-257, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0016.

[3] Rashmi et al.. "Exposure and Avoidance Mechanism Of Black Hole And Jamming Attack In Mobile Ad Hoc Network." International Journal of Computer Science, Engineering and Information Technology 7.1 (2017): 14-22.

[4] Sharma et al., "Guard against cooperative black hole attack in Mobile Ad-Hoc Network." Harsh Pratap Singh et al./International Journal of Engineering Science and Technology (IJEST) (2011).

[5] Singh, et al., "A mechanism for discovery and prevention of coopeartive black hole attack in mobile ad hoc network using AODV protocol." 2014 International Conference on Electronics and Communication Systems (ICECS). IEEE, 2014.

[6] Harsh et al., "Design and Implementation of an Algorithm for Mitigating the Congestion in Mobile Ad Hoc Network." International Journal on Emerging Technologies 10.3 (2019): 472-479.
[7] Singha, Anjani Kumar , Kundu, Shakti, Singh, Songare, Lokendra Singh & Tiwari , Pradeep Kumar (2024) Measuring network security in the cloud : A roadmap for proactive defense, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-B, 889–902, DOI: 10.47974/JDMSC-1964.
[8] Singha, Anjani Kumar, Singh, Harsh Pratap , Kundu, Shakti , Tiwari, Pradeep Kumar & Rajput, Ajeet Singh (2024) Estimating computer network security scenarios with association rules, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-A, 223–236, DOI: 10.47974/JDMSC-1876.
[9] Singh et al. “Detection and Prevention of Black Hole attack In Modified AOMDV Routing Protocol in MANET”, International Journal of Engineering Applied Science and Management, 2020, vol. 1, issue-1.
[10] Rashmi et al, “Prevention Mechanism of Black Hole and Jamming Attack in Mobile Ad Hoc Network”, Journal of Harmonized Research in Engineering, 2020, vol 8, issue 1.
[11] Leena et al., “Reliable Positioning-Based Routing Using Enhance Dream Protocol In Manet”, International Journal of Scientific & Technology Research, Vol 9, issue-1
[12] Luan, Y., Zhao, T., and Huang, W., “A High-Performance Shared Memory Architecture for ADAS SoCs,” *arXiv preprint*, arXiv:2209.05731, 2022. [Online]. Available: https://arxiv.org/abs/2209.05731
[13] Benz, M., Schmidt, D., and Rothermel, M., “AXI-REALM: A Lightweight Interconnect Extension for Traffic Regulation in Real-Time SoCs,” *arXiv preprint*, arXiv:2311.09662, 2023. [Online]. Available: https://arxiv.org/abs/2311.09662
[14] Noami, R., Patra, S., and Verma, K., “Design of an Advanced eXtensible Interface (AXI) Interconnect with a Static Fixed-Priority Arbiter,” *International Journal of Science and Technology (IJST)*, vol. 29, no. 3, pp. 105-117, 2024. [Online]. Available: https://ojs.kmutnb.ac.th/index.php/ijst/article/view/4518