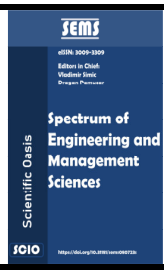
SCIENTIFIC OASIS

Spectrum of Engineering and Management Sciences

Journal homepage: www.sems-journal.org
ISSN: 3009-3309



Optimized Task Deployment in Dynamic Voltage and Frequency Scaling-Enabled Network-on-Chip Systems: Enhancing Energy Efficiency and Real-Time Responsiveness

Kainat Irfan[1], Mujeeb Ur Rehman[1,*]

1 Department of Computer Science, University of Management and Technology, Sadra Badra, Sialkot, Punjab, Pakistan

ARTICLE INFO

Article history:
Received 7 November 2024
Received in revised form 18 November 2024
Accepted 8 December 2024
Available online 8 December 2024

Keywords:
Multi-Core Design; Real-Time Operating System; Network-on-Chip; Offloading; Scheduling Efficiency.

ABSTRACT

In modern multi-design computing systems, which employ dynamic voltage and frequency scaling (DVFS) and network-on-chip (NoC) communications, the optimization of task deployment is precarious for enhancing overall system performance. It introduces a comprehensive methodology that integrates task allocation, scheduling, frequency management, redundancy handling, and diverse data routing approaches. The aim is to optimize energy intake, real-time responsiveness, and system heftiness. The system design features a primary processing element associated with three slave computing units (CUs) within a 2D mesh network, with the primary CU connected to a co-processor for dependent task scheduling. This research also proposes innovative algorithms for co-processor and real-time operating system (RTOS) scheduling to reduce latency and boost power efficiency. These methodologies aim to maximize job scheduling efficiency in RTOS environments, thus refining overall system performance.

1. Introduction

The landscape of computing has been reshaped by multicore architectures, which amalgamate numerous processors onto a solitary chip, yielding platforms characterized by diminished supply frequency, heightened data throughput, and augmented energy efficiency [1]. Recent strides in nanoscale technologies have fostered a paradigm shift towards network-on-chips (NoCs) for processor communication, supplanting traditional, non-scalable data buses [2].

The computational complexity at the processors is small in comparison to the overhead associated with inter-processor communication over NoCs [3], which includes both temporal and energy expenses [4]. The routing path and task mapping decisions determine this communication overhead [5,6]. Data transmission is required when dependent tasks are assigned to different processors, and manifold routing paths—like those found in mesh networks—introduce further complexity inside NoCs. As indicated in Figure 1, the objective of the task deployment is energy

* *Corresponding author.*
E-mail address: mujeeb.rehman.pak@gmail.com

https://doi.org/10.31181/sems21202426i

reduction under real-time limitations. To this end, we take into consideration both energy-oriented and time-oriented pathways as viable possibilities for data transmission. Therefore, as essential components of the task deployment process, inter-processor communication, task mapping, and routing path selection must be carefully considered to maximize overall system performance.

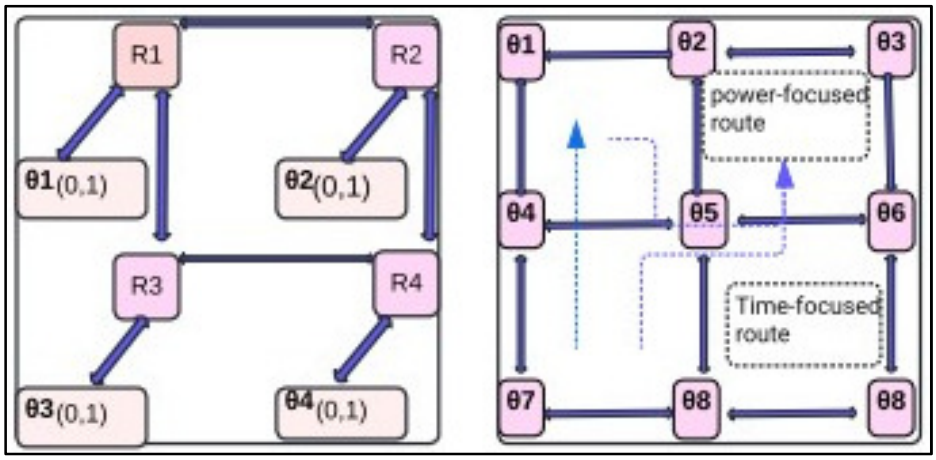


Fig. 1. NOC-based architecture

Section 2 presents a review of related literature covering power-saving techniques, offloading methodologies in RTOSs, and our specific use case. Section 3 outlines the problem statement and a proposed solution framework. Section 4 provides methodologies and techniques. The conclusion is in Section 5.

2. Literature Review

This section provides an overview of the current developments in workload distribution methods, energy conservation methods, and task assignment methodologies for multi-core systems. In an RTOS, the kernel's scheduling algorithm frequently evaluates the necessity for a context switch, selects the subsequent execution task, and manages context storage, allowing application functions to operate without synchronization, resource usage concerns, or function call ordering [7,8]. However, this approach may result in less predictable scheduling and increased system overhead, particularly in control applications.

Decentralized task distribution techniques have proven instrumental in enhancing the scheduling performance of real-time operating systems (RTOS), while concurrently, energy conservation methods strive to minimize power consumption. Employing a NoC-based multi-core system facilitates task parallelization, enabling the partitioning of applications into tasks that can be executed simultaneously across computational units. To oversee energy conservation methods, a specialized management unit dynamically monitors power usage, allocates tasks to computational units, and ensures adherence to real-time and power constraints. Table 1 categorizes representative works from the literature based on varied objectives.

Some studies have exclusively focused on offloading RTOS task scheduling onto adaptable computing platforms, neglecting the integration of energy conservation methods. Conversely, others have incorporated energy conservation methods within operating systems, with only one study [9] integrating them alongside an open-source RTOS. Notably, none of these works have explored clock gating for power optimization under the considered constraints, as delineated in Table 1.

Table 1
Comparison of task deployment

Reference	Task allocation	Multicore utilization	Optimize solution
[1]	√	√	√
[2]	✖	✖	√
[9]	√	√	√
[10]	√	√	√
[11]	√	✖	√
Proposed	√	√	√

Research indicates that offloading task scheduling from running processors can yield faster scheduling and improved predictability, reducing runtime overhead and event response time. Task scheduling can be offloaded to co-processors or hardware, with some work even replicating the entire RTOS in hardware. When dedicated kernel services such as scheduling and inter-task communication were offloaded to hardware, computing performance improved significantly. Hardware-based task scheduling markedly diminishes context switch overhead, evidenced by latency reductions.

Despite extensive research on power-saving techniques, few studies have explored their integration with RTOSs. For instance, a low-power task scheduling approach under e EDF was introduced in [12], resulting in a 27% energy conservation [13]. Similarly, a hypervisor architecture for low-power applications, dynamically adjusting operating frequency and voltage to achieve a 33% power reduction was proposed in [14]. However, these works primarily targeted Linux OS thus leaving a gap in research focusing on RTOSs and power-saving techniques.

The growing data capacity in contemporary Internet of Things (IoT) and CPS applications raises serious concerns about real-time embedded systems' memory power consumption [15]. Because real-time systems require huge memory capacities and DRAM refresh operations, memory can account for as much as 20-50% of CPU power, compared to around 10% in general-purpose systems. A novel swap mechanism is suggested to lower memory power usage to address this problem. This plan optimizes power reductions in both CPU and memory while making use of fast NVM storage. In contrast to conventional real-time task models, the task model is expanded to take storage and memory routes into consideration. This allows for the evaluation of the worst-case execution time by taking into account the overlap of CPU and memory latency.

Field programmable gate arrays (FPGAs) can be used more easily thanks to operating systems for RCOS, which abstract hardware specifics, use virtualization, and maintain shared resources [16]. They improved performance and cut down on energy usage by enabling the simultaneous execution of hardware functions on the same FPGA. Applications were able to take advantage of FPGA benefits thanks to RCOS, which also addressed issues with restricted areas and configuration port accessibility. Key ideas were explained, cutting-edge RCOS were emphasized, and future trends including real-time processing specialization, low energy consumption, dependability, safety, and security were noted.

Controlling peak power consumption is essential in contemporary multicore mixed-criticality (MC) systems to avoid thermal problems that can compromise system timeliness and dependability [17]. To minimize peak power usage during runtime, we provide an online peak power and thermal management heuristic for multicore MC systems that make use of dynamic slack and per-cluster dynamic voltage and frequency scaling (DVFS). Our method optimizes the system's peak power and temperature by choosing the best task for slack assignment. While satisfying deadline limitations in various criticality modes, experimental validation on the ODROID-XU3 platform with embedded real-

time benchmarks demonstrates up to a 5.25% decrease in system peak power and a 20.33% reduction in maximum temperature compared to existing approaches.

Multiprocessor systems-on-chips (MPSoCs) based on the NoC technology are becoming more and more common in modern embedded systems, especially for multimedia streaming applications [17]. DVFS in conjunction with task-level retiming has shown to be a successful method for dramatically lowering energy consumption in these systems. An energy-aware scheduler for real-time streaming applications in dissimilar NoC-MPSoCs is recommended in this research. R-CTG is a method that pursues to reduce retiming latency while preserving energy economy. The scheduler, ALI-EBAD, performs better in terms of energy efficiency than current job schedulers, according to experimental results.

To minimize uncertainty propagation in cloud service environments, a work proposes a unique scheduling architecture in response to these problems [19]. In this way, the architecture successfully mitigates the effects of unpredictable task execution and data transfer durations by controlling the count of workflow tasks that are directly waiting on each service instance. To dynamically modify plans in response to uncertainties, the suggested uncertainty-aware online scheduling algorithm (ROSA) combines proactive and reactive tactics. Employing simulation trials, ROSA outperforms five common algorithms, obtaining notable gains in costs (up to 56%), variance (up to 70%), resource utilization (up to 37%), and fairness (up to 37%).

A task scheduling policy that handled the coexistence of classic hard real-time jobs and sporadically arriving interactive tasks in the context of smart industrial systems integrating IoT and CPS technologies were presented in [18]. The policy is a two-phase process that used offline scheduling based on evolutionary algorithms to meet strict real-time deadlines and "virtual real-time tasks" for interactive work. Interactive tasks were scheduled online, and workloads were updated regularly to accommodate changes. According to experimental data, energy consumption might be reduced by 66.8% without compromising deadlines, allowing interactive tasks to have waiting periods of less than three seconds.

Moreover, energy-efficient task scheduling is essential for MPSoC designs in contemporary embedded systems [19]. We study this problem in Heterogeneous MPSoCs (HMPSoCs) based on NoC equipped with DVFS. Our energy-efficient task scheduling heuristic (ETSH) algorithm improves state-of-the-art methods by converting intra-data dependencies into inter-data dependencies represented by directed acyclic graphs (DAGs). Findings from synthetic and real-world task graphs (TGs) show that average energy efficiency can be as high as 38% and up to 20%, respectively, with and without coarse-grained software pipelining.

Smart industrial systems increasingly offer flexible processes incorporating human interactions with hard real-time operations, leading to varied task characteristics [18]. This is due to recent breakthroughs in IoT and cyber-physical systems. This paper suggests a novel task scheduling approach that uses two-phase scheduling and "virtual real-time tasks" to address this. Genetic algorithms are used in offline scheduling to identify processor voltage levels, memory locations, and to reserve virtual real-time jobs for interactive tasks. This helps to guarantee that hard real-time task deadlines are fulfilled. Interactive jobs are managed by online scheduling during the virtual real-time task periods. Offline scheduling should be updated periodically to account for changing interactive workloads. The experimental findings show an average energy consumption decrease of 66.8% without missing any deadlines and guaranteeing waiting times of less than three seconds for interactive jobs.

In order to satisfy the performance requirements of multimedia streaming applications, the use of MPSoCs based on NoC technology is expanding in contemporary embedded systems [16]. DVFS in conjunction with task-level coarse-grained software pipelining effectively reduces MPSoC energy

usage at the cost of extra delay. In this study, we present an energy-aware scheduler on heterogeneous NoC-MPSoCs based on voltage frequency island (VFI) for real-time streaming applications. We present R-CTG, a technique that outperforms R-DAG by integrating re-timing with DVFS to minimize latency without sacrificing energy efficiency. The experimental results show that our scheduler, ALI-EBAD, has a higher energy efficiency than competing task schedulers.

Furthermore, task mapping is essential for load balancing and communication optimization in multiprocessor sys-tem-on-a-chip software design. Compared to earlier approaches, the research [20] suggests an ILP-based strategy that takes into account both characteristics while using fewer variables. Additionally, it presents an enhanced ε-constraint method for flexibility and a task-processor-cluster strategy to boost scalability. The efficacy of the suggested strategy is authorized by experimental findings on a range of CPU systems.

Handling power consumption in real-time systems grants challenges, particularly concerning processing time for retrieving and monitoring voltage regulators' standards. While power gating has been discovered for power reduction, its effect on real-time capabilities remains unaddressed. Incorporating power-saving policies into RTOSs necessitates thorough attention to DVFS as well as clock gating, confirming an equilibrium between power optimization and real-time operational proficiency.

3. Problem Statement and its Proposed Solution

3.1 Problem Description

In multi-core systems with NoC designs, task deployment gifts significant complications to real-time performance, energy optimization, and system steadfastness. Multiple PEs operating in various power states are repeatedly used in these systems. Fault tolerance methods such as clock gating, task duplication, and DVFS are also mutual. The supervision of energy efficiency is made more challenging by the substantial energy expenditure associated with data processing and inter-PE communication. Vigorous power management familiarizes variability that compromises real-time performance and can affect erratic scheduling and neglected deadlines. Moreover, to avoid data loss and guarantee continuous operation, conserving system reliability obliges the use of effective fault management procedures, such as task redundancy and duplication.

3.2 Solution Framework

A comprehensive solution framework that syndicates cutting-edge task management approaches with energy-saving measures has been put forth to discourse these issues. In order to minimize energy consumption while meeting real-time restrictions, the framework practices a mixed-integer non-linear programming model for job allocation and scheduling. It takes into account computational requirements, task dependencies, and communication configurations. Utilizing DVFS, voltage, and frequency stages are vigorously adjusted to poise energy savings with performance essentials. Power usage is diminished during phases of inactivity with clock gating. Task duplication through multiple PEs advances reliability and guarantees that operations endure even in the event of hardware disasters. Besides, task scheduling effectiveness is increased by offloading it to a devoted hardware unit, and multipath data routing lessens NoC communication overhead. The framework's capacity to increase energy efficiency, real-time sensitivity, and system reliability is authenticated by experimental results.

4. Methodologies and Techniques

A multi-core system design is generated using the proposed methodology to look into ways to cut power usage without bargaining real-time performance. The system involves NoC with four compute units organized in a 2D-mesh topology and set in a master-slave formation, as shown in Figure 2. Expending an offloading strategy for every PE advances context switching and restraints real-time processes. While the principal PE is in control of voltage scaling, power supervision, and mapping techniques, each auxiliary PE is in custody of clock gating and frequency scaling. The hardware module that can be reconfigured is specified by the dotted outline.

One auxiliary way to increase system performance is to dispense task scheduling responsibilities to a co-processor. Exhausting a NoC topology, this scheme connects three slave PEs and one master PE to formulate a multi-core design. A voltage regulator is constructed into every slave PE to vigorously adjust voltage ranks to boost power output. Each slave PE also has a router attached to RTOS and a network interface processor (NIP) for task accomplishment.

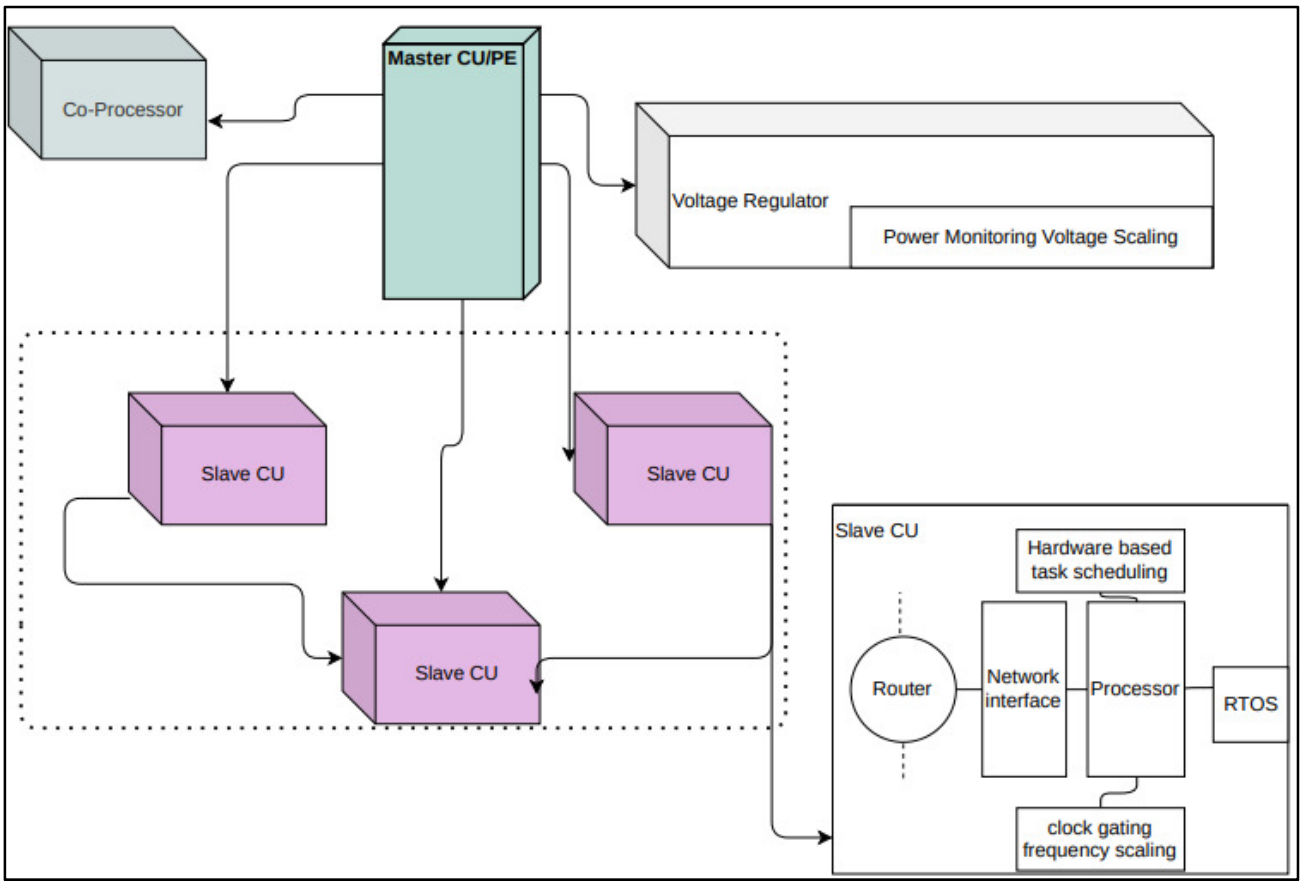


Fig. 2. Hardware structure comprising of a main processing element associated with NOC

The system strategy, power consumption checking, and task entrustment to the slave PEs are under the purview of the master PE. It is supplementary fixed to a co-processor, which knobs task scheduling, to expand system efficiency. The master PE is offloaded and latency is lessened by the co-processor, which is in control of scheduling mutually independent and dependent responsibilities.

In count, a job scheduling method is submitted for the master PE to enhance task distribution. With the assistance of this algorithm, which allocates jobs intelligently based on dependencies, communication configurations, and power measurements, the multi-core design's computing resources are utilized efficiently.

4.1 Algorithms

By allocating task scheduling obligations to a specialized co-processor, the co-processor task scheduling procedure benefits multi-core schemes activate more energy-efficiently. It also minimizes the interrelated power consumption and declines the processing liability on the primary PEs.

Algorithm 1. Co-processor task scheduling

```
Output: A task list L allocated to every slave PE
Input: Task list T with priorities for every task tj in T for each task tj in T do
if parent(tj)=heta, then
Add tj to the allocated task list L (master) for each slave PE do
Send tj to the coprocessor
End
return L
```

Every task in the prioritized task list is examined by the algorithm as it operates. The coprocessor is tasked with scheduling dependent tasks, whilst jobs without dependencies are assigned directly to the master PE. By shifting the scheduling responsibilities to the co-processor, the computational burden on the primary PEs is decreased and tasks can be assigned more effectively. Through task scheduling burden distribution to the co-processor, the primary PEs can reduce power consumption or even go into sleep mode.

In RTOS, Algorithm 2 optimizes task assignment in a multi-core system by considering both computation and communication energy. Primarily, it prepares the output parameters and nature tasks based on their enslavements and priority inside the RTOS. Then, it treasures PE with the lowest energy requirement, computes task assignment, implementation time, and communication parameters consequently. This algorithm guarantees efficient resource utilization and diminishes energy consumption in multi-core methods running RTOS.

Algorithm 2. Task scheduling algorithm

```
Input: yi and hi for all i in M0
Output: xik, uij, and tsi for all i in M0
initialize O[i]=−1 for all i in M0, sort tasks in M0 according to their dependencies and priority
for each task i in M0, where hi=1 do
Set MinEng to infinity for each PE k in N do
Calculate ecompk and ecommk using CalculateCompEnergy and CalculateCommEnergy Set
Eng to ecompk + ecommk if Eng < MinEng then
Set MinEng to Eng Set O[i] to k
end
Calculate xik according to O[i]=k, calculate uij and tsi according to xik, tcommi, and tcompi
using CalculateTaskCompletionTime and CalculateTaskStartTime
end
```

5. Conclusions

The NOC-based multicore design has publicized noteworthy achievements in system performance, latency fall, and energy efficiency with the accumulation of a co-processor for job scheduling. From side to side, the transfer of task scheduling obligations to the co-processor, the system proficiently condenses the overhead correlated to context switching and inter-processor communication. This causes execution configurations that are more expectable and established,

exclusively for real-time applications. This work contributed by designing a diverse multi-core architecture specifically tailored for real-time systems. It incorporated power-saving techniques and offloading methodologies, including a co-processor to optimize scheduling and reduce latency. Moreover, a task mapping strategy was introduced to efficiently allocate tasks to secondary computational units, taking into account task dependencies and power consumption metrics while ensuring real-time performance.

References

[1] Akgün, G., Kolarov, B., Kalberlah, H., Wulf, C., Willig, M., et al. (2024). Exploration of Power-Savings on Multi-Core Architectures with Offloaded Real-Time Operating System. *IEEE Access, 12*, 11294-11315. https://doi.org/10.1109/ACCESS.2024.3354178.

[2] Mo, L., Zhou, Q., Kritikakou, A., & Liu, J. (2022). Energy efficient, real-time and reliable task deployment on noc-based multicores with DVFS. In *2022 Design, Automation & Test in Europe Conference & Exhibition* (DATE) (pp. 1347-1352). IEEE. https://doi.org/10.23919/DATE54114.2022.9774667.

[3] Gomatheeshwari, B., Gopi, K., & Mathias, A. (2023). Low-complex resource mapping heuristics for mobile and iot workloads on NoC-HMPSoC architecture. *Microprocessors and Microsystems, 98*, 104802. https://doi.org/10.1016/j.micpro.2023.104802.

[4] Tariq, U.U., Wu, H., & Abd Ishak, S. (2020). Energy and memory-aware software pipelining streaming applications on NoC-based MPSoCs. *Future Generation Computer Systems, 111*, 1-16. https://doi.org/10.1016/j.future.2020.04.028.

[5] Amin, W., Hussain, F., Anjum, S., Khan, S., Baloch, N. K., Nain, Z., & Kim, S. W. (2020). Performance evaluation of application mapping approaches for network-on-chip designs. *IEEE Access, 8*, 63607-63631. https://doi.org/10.1109/ACCESS.2020.2982675.

[6] Anuradha, P., Majumder, P., Sivaraman, K., Vignesh, N. A., Jayakar, A., et al. (2024). Enhancing high-speed data communications: Optimization of route controlling network on chip implementation. *IEEE Access, 12*, 123514-123528. https://doi.org/10.1109/ACCESS.2024.3427808.

[7] Ismael, G.A., Salih, A.A., AL-Zebari, A., Omar, N., Merceedi, K.J., et al. (2021). Scheduling Algorithms Implementation for Real Time Operating Systems: A Review. *Asian Journal of Research in Computer Science, 11*(4), 35-51.

[8] Haur, I., Béchennec, J.L., & Roux, O. H. (2021). Formal schedulability analysis based on multi-core RTOS model. In *Proceedings of the 29th International Conference on Real-Time Networks and Systems* (pp. 216-225). https://doi.org/10.1145/3453417.3453437.

[9] Han, J.J., Lin, M., Zhu, D., & Yang, L.T. (2014). Contention-aware energy management scheme for NoC-based multicore real-time systems. *IEEE Transactions on Parallel and Distributed Systems, 26*(3), 691-701. https://doi.org/10.1109/TPDS.2014.2307866.

[10] Li, D., & Wu, J. (2014). Minimizing energy consumption for frame-based tasks on heterogeneous multiprocessor platforms. *IEEE Transactions on Parallel and Distributed Systems, 26*(3), 810-823. https://doi.org/10.1109/TPDS.2014.2313338.

[11] Xie, G., Chen, Y., Xiao, X., Xu, C., Li, R., & Li, K. (2017). Energy-efficient fault-tolerant scheduling of reliable parallel applications on heterogeneous distributed embedded systems. *IEEE Transactions on Sustainable Computing, 3*(3), 167-181. https://doi.org/10.1109/TSUSC.2017.2711362.

[12] Mo, L., Kritikakou, A., & Sentieys, O. (2018). Controllable QoS for imprecise computation tasks on DVFS multicores with time and energy constraints. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 8*(4), 708-721. https://doi.org/10.1109/JETCAS.2018.2852005.

[13] Yoon, S., Park, H., Cho, K., & Bahn, H. (2022). Supporting swap in real-time task scheduling for unified power-saving in CPU and memory. *IEEE Access, 10*, 3559-3570. https://doi.org/10.1109/ACCESS.2021.3140166.

[14] Wulf, C., Willig, M., Akgün, G., Göhringer, D. (2021). Operating Systems for Reconfigurable Computing: Concepts and Survey. In: Jahre, M., Göhringer, D., Millet, P. (eds) *Towards Ubiquitous Low-power Image Processing Platforms*. Springer, Cham. https://doi.org/10.1007/978-3-030-53532-2_4.

[15] Ranjbar, B., Nguyen, T.D., Ejlali, A., & Kumar, A. (2020). Power-aware runtime scheduler for mixed-criticality systems on multicore platform. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40*(10), 2009-2023. https://doi.org/10.1109/TCAD.2020.3033374.

[16] Tariq, U.U., Ali, H., Liu, L., Hardy, J., Kazim, M., & Ahmed, W. (2021). Energy-aware scheduling of streaming applications on edge-devices in IoT-based healthcare. *IEEE Transactions on Green Communications and Networking, 5*(2), 803-815. https://doi.org/10.1109/TGCN.2021.3056479.

[17] Chen, H., Zhu, X., Liu, G., & Pedrycz, W. (2018). Uncertainty-aware online scheduling for real-time workflows in cloud service environment. *IEEE Transactions on Services Computing, 14*(4), 1167-1178. https://doi.org/10.1109/TSC.2018.2866421.

[18] Yoo, S., Jo, Y., & Bahn, H. (2021). Integrated scheduling of real-time and interactive tasks for configurable industrial systems. *IEEE Transactions on Industrial Informatics, 18*(1), 631-641. https://doi.org/10.1109/TII.2021.3067714.

[19] Ali, H., Tariq, U.U., Liu, L., Panneerselvam, J., & Zhai, X. (2019). Energy optimization of streaming applications in IoT on NoC based heterogeneous MPSoCs using re-timing and DVFS. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation* (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI) (pp. 1297-1304). IEEE. https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00240.

[20] Huang, K., Zhang, X., Zheng, D., Yu, M., Jiang, X., et al. (2018). A scalable and adaptable ILP-based approach for task mapping on MPSoC considering load balance and communication optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 38*(9), 1744-1757. https://doi.org/10.1109/TCAD.2018.2859400.