Computers, Materials & Continua
DOI: 10.32604/cmc.2023.031223
Article

Tech Science Press

Energy-Efficient Scheduling Based on Task Migration Policy Using DPM for Homogeneous MPSoCs

Hamayun Khan[1,*], Irfan Ud din[2], Arshad Ali[3] and Sami Alshmrany[3]

[1]Department of Electrical Engineering, Faculty of Engineering & Technology Superior University Lahore, 54000, Pakistan
[2]Department of Computer Science, Faculty of Computer Science & IT Superior University Lahore, 54000, Pakistan
[3]Faculty of Computer and Information Systems, Islamic University of Madinah,
Al Madinah Al Munawarah, 42351, Saudi Arabia
*Corresponding Author: Hamayun Khan. Email: hamayun.khan@superior.edu.pk

Received: 12 April 2022; Accepted: 22 June 2022

Abstract: Increasing the life span and efficiency of Multiprocessor System on Chip (MPSoC) by reducing power and energy utilization has become a critical chip design challenge for multiprocessor systems. With the advancement of technology, the performance management of central processing unit (CPU) is changing. Power densities and thermal effects are quickly increasing in multi-core embedded technologies due to shrinking of chip size. When energy consumption reaches a threshold that creates a delay in complementary metal oxide semiconductor (CMOS) circuits and reduces the speed by 10%–15% because excessive on-chip temperature shortens the chip's life cycle. In this paper, we address the scheduling & energy utilization problem by introducing and evaluating an optimal energy-aware earliest deadline first scheduling (EA-EDF) based technique for multiprocessor environments with task migration that enhances the performance and efficiency in multiprocessor system-on-chip while lowering energy and power consumption. The selection of core and migration of tasks prevents the system from reaching its maximum energy utilization while effectively using the dynamic power management (DPM) policy. Increase in the execution of tasks the temperature and utilization factor (u_i) on-chip increases that dissipate more power. The proposed approach migrates such tasks to the core that produces less heat and consumes less power by distributing the load on other cores to lower the temperature and optimizes the duration of idle and sleep times across multiple CPUs. The performance of the EA-EDF algorithm was evaluated by an extensive set of experiments, where excellent results were reported when compared to other current techniques, the efficacy of the proposed methodology reduces the power and energy consumption by 4.3%–4.7% on a utilization of 6%, 36% & 46% at 520 & 624 MHz operating frequency when particularly in comparison to other energy-aware methods for MPSoCs. Tasks are running and accurately scheduled to make an energy-efficient processor by controlling and managing the thermal effects on-chip and optimizing the energy consumption of MPSoCs.

Keywords: Dynamic power management; dynamic voltage & frequency scaling; dynamic thermal management; multiprocessor system on chip; complementary metal oxide semiconductor reliability

1 Introduction

The dissipation of energy becomes the captious design constraint of system-on-chip (SoCs) over the past decade as it limits the performance, reliability and battery life. Due to advancements in computational embedded devices and increasing multi-task execution than ever before [1].

MPSoC is widely deployed in high-performance computing and application-specific embedded systems such as gaming and aerospace-based systems for real-time response. It contains high-performance ARM Cortex-A7, ARM Cortex-A15 and an energy-efficient INTEL PXA-270 processor [2]. Introduced a technique that reduces resistance and energy because lack of concentration can affect the reliability and life span of the chip as well as overall systems performance [3]. Due to the high processing of tasks temperature on the chip increases. Various mechanisms are used to reduce the thermal effects due to high heat and decrease the performance of the system because high heat causes the chip to be damaged. A central processing unit is working as the main processing unit for performing the instructions read and write operation. CPU unit is placed in all the modern embedded systems [4]. The processing unit needs to be updated with time if the processing unit has a higher processing speed it can execute and manage intense tasks efficiently at short intervals of time. Advancement in the processing unit makes our system run heavy tasks but it can have some issues e.g., Dimension, cost, energy, power utilization, performance, reliability and processing speed. Switching of the task is the major issue with the evolvement of the processor [5]. DPM mainly deals with the development of policies that analyze the run time behavior of the system to reduce the power consumption of the MPSoC system [6].

Introduced simulated annealing based ($\mathrm{LPPWU_{Sa}}$) optimization strategies for reducing the energy and solving the system-level low power design problems [7]. Proposed a dynamic thermal management (DTM) based energy optimization technique for the MPSoC platform with dynamic voltage and frequency scaling (DVFS) enabled homogeneous processors. These are one of the most reliable techniques to reduce and stabilize the temperature of the multi-core system. In a multi-core system, an exponential decrease in the temperature also reduces the power utilization [8]. Introduced homogeneous multiprocessing often known as symmetric multiprocessing (SMP) is a common type of multiprocessing in computer systems, in which two or more identical processors share a single main memory for process execution [9]. Introduced dynamic thermal management (DTM) based technique that efficiently manages the thermal responses of a processing system. Many techniques are combined to manage temperature and thermal responses including DVFS, DPM and Dynamic voltage scaling (DVS). These techniques are very useful but they cause some reliability and performance issues. They are mostly used to resolve on-chip power dissipation problems [10]. Modern MPSoCs are based on CMOS chips. The switching frequency regulates how often the switches occur. Double-edge-triggered flip-flops can be utilized to minimize dynamic power for MPSOC technology. An embedded device's power consumption is categorized as $P_{dynamic}$ and P_{static} [11].

The accumulative power of a CPU is the $\sum$ of the $P_{dynamic}$ because of switching and the P_{static} occurs due to leakage of power. The primary factor in CMOS circuits is dynamic power dissipation that occurs because of transistor switching. $P_{dynamic}$ can be calculated using Eq. (1):

$$P_{dynamic} = C_{load} \cdot N_{cs} \cdot V_{dd} \cdot f \tag{1}$$

$$f = R \cdot (V_{dd} - V_t) / \cdot V_{dd.}{}^2 f \tag{2}$$

Load capacitance is denoted as C_{load}, N_{cs} is the number of circuit transitions while supplied voltage is denoted as V which is equivalent to switching voltage. f is the clock frequency while V_t represents the threshold voltage. R is a constant and reduction in supply voltage is represented as V_{dd} as shown in Eq. (2). L_a is the total number of logic gates in the CMOS circuit. The total power of the CMOS circuit is represented in Eq. (3):

$$P_{total} = (C_{eff}\ V_{dd.}{}^2 f + L_a\,(V_{dd.} R_{3.}\ \ ) \tag{3}$$

Most of the embedded computing circuits aim to give maximum performance while using minimum power. In static power, the dissipation of power occurs when the circuit is not changing states due to leakage current. The short circuit power is utilized when both positive channel metal-oxide-semiconductor (PMOS) and negative channel metal-oxide-semiconductor (NMOS) have switched ON for a short period unless the path between supply voltage is directed with the ground [12].

2 Literature Review

A decrease in the chip size of a multi-core processor certainly increases the number of transistors on a chip more rapidly than before. The chip can require more energy due to which a gradual increase in power density is observed that affects the reliability of a multi-core processor [13]. Scheduling and task allocation techniques are facing issues during the migration of tasks to balance and manage power on multiprocessor systems. Tab. 1 represents the comparison of various online and offline DPM-based scheduling techniques [14].

Table 1: Summary of offline & online DPM approaches

Algorithm	Speed set	Periodicity	Scheduler	Dynamic slack
HSCTB1	Continuous	A-periodic	Offline EDF	No
HSCTB2	Discrete	A-periodic	Offline EDF/FP	Implicit
LC-EDF	Discrete	Periodic	Online EDF	Implicit
Energy Saving-RHS	Continuous	Periodic	Offline Rate Monotonic	Implicit
ERTH	Discrete	Sporadic	Online EDF	Explicit

Fig. 1 shows that higher power consumption can increases higher temperature and higher resistance eventually causing lower possible speed and hotspots that cause the permanent failure of the device [15].

Introduced a technique that increases the life span of the chip by reducing the consumption of power in multiprocessors-based systems. DPM-based technique affects the performance of the processor by suddenly switching the system from idle to running state. An efficient dynamic power management policy keeps both the reliability and performance while considering power degradation within allowed limits. Designs of these policies are considered to be an active and burning research topic in the field of MPSoC while designing embedded systems few challenges occur such as size, cost, power consumption and reliability [16].

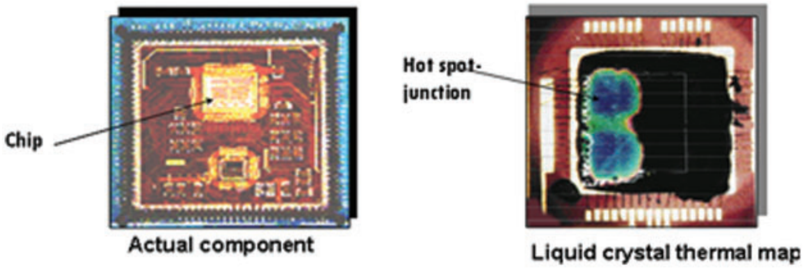

Figure 1: High utilization effects on system on chip (SOC)

2.1 Dynamic Power Management

DPM technique selectively turns off all those components that are not in use. DPM is used in several portable systems but its applications are not yet explored because of the complexity of interfacing heterogeneous components. The fundamental problem in the implementation of DPM techniques is the non-uniform workload during the execution of the task. To solve this problem DPM uses a predictive algorithm that predicts the future workload by using different predictive models. It covers several system-level DPM approaches to save energy. DTM technique is used to find an optimal solution to avoid peak temperature which causes hotspots on chips [17]. Energy is not directly affected by temperature but when temperature increases from the threshold value some cooling mechanism is required to reduce the temperature of the system. Cooling mechanisms consume energy to reduce the temperature [18]. In a single-core processor if we increase the frequency by 50% that roughly increases the power consumption two times while in dual-core systems power consumption increases by 30% if we increase the supply voltage by keeping the frequency constant. Due to this power increases more rapidly because power is directly proportional to the square of supply voltage [19]. DPM allows MPSoCs to minimize power and energy consumption by optimizing the dynamic power. We can reduce the frequency which saves a considerable amount of power but causes performance degradation in the multiprocessor. In the same way, as the supply voltage decreases the dynamic power is reduced almost four times but it has an overhead by reducing the supply voltage. An increase in circuit delay occurs so the circuit cannot operate at the same frequency. If we decrease the supply voltage and frequency the dynamic power decreases cubically but an increase in the execution period occurs linearly that degrading the performance of the chip [20].

DPM is a design technology that reconfigures the whole computing system dynamically in such a way that requested services can be provided with the minimum number of active CPUs with suitable performance levels [21]. Proposed a dynamic voltage and frequency scaling technique that is used to improve the performance and consider the load balancing issue in multi-cores of a processor. DVFS technique dynamically set the workload on the cores for this an irregular parallel divide and conquer algorithm is used to equally share the workload that reducing 31% of energy consumption at 400 MHz [22].

When task execution is interrupted in modern CPUs during the transition to a low-power state each low-power state σx is defined by its power consumption $P_{\sigma x}$ while the time and resource requirements of entering or leaving that state are represented as $\delta_x \rightarrow x,\ \delta_x \rightarrow s,\ E_s \rightarrow x,\ and\ E_s \rightarrow s$ taking the sum of both the transition overheads of initial and final is necessary to get the total energy overhead that is linked with low-power state σx namely δ_x and E_x for comparison. Power usage is lower but the time and energy overhead are higher in the transition state to move the MPSoC into the low-power state as shown in Fig. 2. Where P is the processor's power usage in its normal condition when no jobs are running while P_u is a specific idle state with a low transition overhead occurs. If the CPU is kept active throughout the idle interval the power consumption will be considered at the lowest speed. Various parameters define different low-power states. The two alternative state transitions represent a low-power state $\sigma1$ with moderate power usage and a rapid breakeven time while on other hand the low-power state $\delta2$ reduces the power usage. The transition of state in low power mode occurs for short period due to the involvement of time and energy overheads [23]. Processors can achieve an active state rapidly when the transition's energy overhead is minimal in such conditions the power state is known as the power-saving state.

$$TB_e = \max\left(\delta x \frac{E_x - \delta x_s * P_{\sigma x}}{P_u - P_{\delta x}}\right) \quad (4)$$

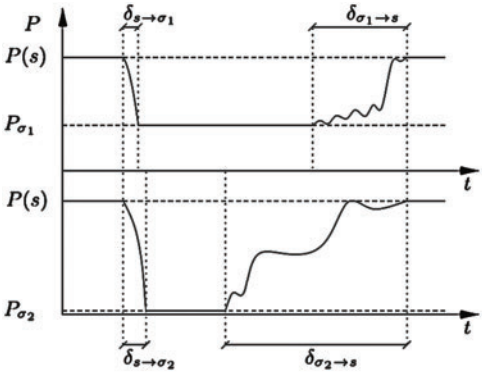

Figure 2: Two low-power states of an MPSoC

Eq. (4) represents the parameter B_e known as the break-even time. B_e refers to the minimum length of idle interval that must be provided to schedule and utilizes the sleep state δx efficiently. B_e is the sum of the total time required to complete the state transition and the duration of idle time that is required to find a way to reduce the shifting energy [24]. The below Eq. (5) determines the BET_{sleep} break-even time.

$$BET_{sleep} = \max\left(t_{pa} \frac{E_{Pa} - P_{sleep} * t_{pa}}{P_{idle} - P_{sleep}}\right) \quad (5)$$

The short circuit power is utilized when both PMOS and NMOS are on for a short period. Eq. (6) represents the transition of the state E_0 and its power dissipation in running and the idle state as P_w and P_s [25].

$$P_w = T_{be} = E_0 + P_s * (T_{be} - T_0) \quad (6)$$

In Eq. (7) (T_{be}) breakeven time measures the length of the idle state of the CPU to optimize power.

$$T_{be} = \left(\frac{E_0 - P_s * T_0}{P_a - P_s}\right) \quad (7)$$

DPM analyzes the run time behavior of the system to reduce the power consumption of the MPSoC. During the running state of an application a selective shut-down of the system components occur that are in the idle state increases the performance. When the CPU starts to transition the energy required for the state transition from sleep to idle and from idle to running is represented as E_i and its power dissipation at state 1 is denoted as P_a and at state 2 its P_s. For highperformance delay due to the state transition must be less than T_{be} as shown in Eq. (8):

$$T_{be} = max\left[\left(\frac{E_i - P_s * T_0}{P_a - P_s}\right) * T_0\right] \tag{8}$$

Fig. 3 illustrates the behavior of MPSoC. On the left side, the MPSoC is running while on the other hand the device is in an idle state. The energy consumption on both ends is equal because of the break-even time in DPM-based techniques [26].

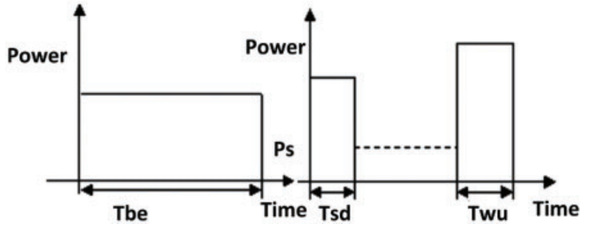

Figure 3: Task mapping in the system on chip working/idle state

Proposed an accuracy measurement model based on scheduling for medical imaging using a high-quality multiprocessor CPU and general processing unit (GPU) based computing environment to speed up the simulation rate and enhance the real-time performance [27]. Fig. 4 illustrates the optimization of power using the DPM technique that shows if the workload is not uniform on a system therefore the idle component of the system is considered [28].

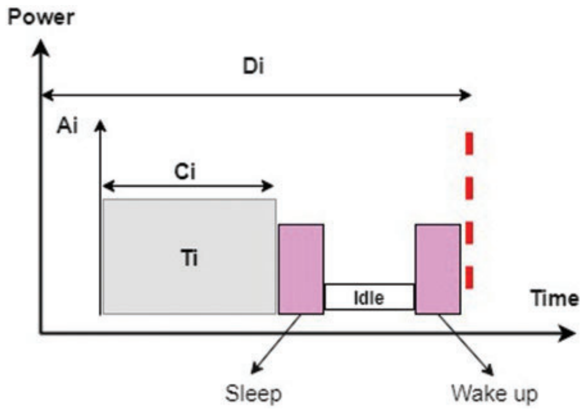

Figure 4: Optimization of power with DPM

2.2 Predictive Techniques

A technique that measures the length of the next idle period of the processor using a simulation tool for real-time multiprocessor (STORM). The decision can be quickly made when the processor is in sleep mode and this policy doesn't predict the length greater or less than the break-even period (T_{be}). Introduced an exponential average scheme that predicts the next idle period length by considering an exponential average of the actual lengths of the previous idle period and the predicted one as shown in Eq. (9):

$$ln + 1 = b \cdot ln + (1 - b) \cdot ln \tag{9}$$

The new predicted values are represented as ln +1 and the last predicted value is denoted as ln. Where ln is the latest idle period while *b*' is considered as an attenuation constant ranging between 0 to 1 [29]. The hardware architecture of Marvel X-Scale Intel PXA-270 MPSoC is widely used for predictive offline DPM techniques for both homogeneous and heterogeneous processors using STORM [30].

3 Problem Definition

The most critical concerns in multi-core embedded systems are the performance and life span of the chip. The task scheduling and switching of jobs from one core to another are one of the major issues in today's MPSoC. Increasing power dissipation and energy utilization increases on chip temperature and resistance which reduces the life span of the chip. It also affects the reliability (hotspots, thermal cycles) as well as lowers the maximum speed for all battery powered devices, particularly in embedded systems that cause multiple performance and reliability issues. The key design difficulty in a task migrationbased system is an accurate forecast of energy, power, coolest core, utilization factor and the workload that needs to be relocated on an individual CPU. In multiprocessor systems task switching across various cores is a prevalent problem. Because the destination core may be in sleep mode there may be a delay while moving tasks from one core to another. Task migration is a technique for reducing power and energy while improving performance.

Power and energy optimization on multi-core systems are developed to address MPSoC dissipation concerns. The influence of tasks may be recognized when looking for the best solution task factors can be examined as tasks have an impact on each other. An energy-efficient task migration policy based on an EDF algorithm that optimizes energy while considering different configurations for migration of load is proposed in this research work. Normally, task schedules on each core are independent.

Problem 1: *Energy-Efficient Multiprocessor Scheduling Technique (EEMS).*

Consider a system having a periodic task set $\tau \ = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5 \ldots, \ \tau_n)$ over $m_{identical}$ processors where many tasks in τ have a common deadline *D* and are ready at time 0. Each task τi ∈ τ is linked with CPU requirement that is approximately equal to power consumption function *Pi(s)* and CPU-cycles *ci* at the given CPU speed. The objective is to reduce the total energy consumption, of the CPU by scheduling for τi ∈ τ over $m_{identical}$ to complete before the deadline by allowing the suitable task migration among processors and core switching technique.

3.1 Proposed Energy-Efficient Multiprocessor Scheduling Algorithm with Task Migration

In this section, we have proposed EA-EDF an optimal algorithm for the *EEMS* problem. Since the power consumption *Pi(s)/s* is increasing at each cycle for every task τi ∈ τ in a given periodic task set, the CPU executes each task τi ∈ task set τ at some speed *s.* For this we have proposed EA-EDF an energy-efficient earliest deadline first scheduling algorithm based on the DPM technique for task migration policy and a guarantee on the energy consumption for MPSoC by monitoring the utilization factor. An energy-efficient policy for various configurations of cores to predict the power and energy profiles using both hardware & software architecture of the Intel PXA-270 MPSoC. Scheduling technique based on task migration reduces energy and produces energy-efficient results according to the utilization factor (u_i). The proposed model uses an EA-EDF scheduler for energy and power optimization and is independently demonstrated to be the best-performing technique at run time.

The suggested policy operates with a set of parameters and generates effective outcomes for the given workload while also optimizing the chip's lifespan and improving QoS by lowering energy consumption. The task set generator provides randomized task sets under a ready task queue that are being used to generate workloads under various constraints as well as the number of CPU cores specified by the user in extensible markup language (XML) based on the application given to STORM as an input. EA-EDF the scheduler can also have a set of prepared tasks that are scheduled according to the scheduling policy by considering suitable migration of task policy that produces energy and power profiles on the individual cores according to the given set of parameters in XML. A periodic task set $\tau\ = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5 \ldots.,\ \tau_n)$ and a utilization factor that is non-increasing $\left(u_{i=}u_{i+1} \text{ for all } i = 1, \leq i \leq n,\ \text{where } u_{i=}\frac{\alpha_i}{p_i}\right)$, consider $u_i\ = \sum_{j=0}^{i}\ = 1$ for $\forall \mathrm{i} = 1,\ \leq \mathrm{i} \leq \mathrm{n}$ Let ϕ' is a set of CPU with $\tau\ \&\ m\ \leq\ n\ \&\ c_{a1}, c_{b1}, c_{c1}, \ldots c_m$, capacities $c_i\ \geq\ c_{i+1}$ for $\forall \mathrm{i}\ =\ 1,\ \leq\ i\ \leq\ m,\ \ \tau$ can be scheduled on the uniform MPSoC that meets all the deadlines $u_n\ \leq\ c_m$; where $u_a \leq c_a$ for all $k\ = 1, 2, .\mathrm{m}$; Overall utilization factor: $u_{i=}\frac{\alpha_i}{p_i}$ for $\forall\ \ c\ =\ \tau_i\left(c_i,\ p_{i,}, d_i\right)$ and n periodic task $\tau\ = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5 \ldots.,\ \tau_n)$ scheduled on a CPU the total u_i is defined as $\mathrm{u}_\tau\ = \sum_{k=1}^{n}\frac{c_a}{p_a} \leq 1$; for $\forall load$ a set of $\tau = \tau_i\left(c_a,\ p_{a,}, d_a\right)$; for $\forall\ n$ periodic task $\{\tau_1, \tau_2, \tau_3 \ldots.\tau_n\}$ with random deadlines scheduled on a CPU, $Load\ (\tau\,) = max\ \{\mathrm{U}_\tau\ \}$, Eq. (10) is used to measure the densities of MPSoCs as:

$$\partial_a = \frac{c_a}{min\ \{d_a,\ T_{a,}\}} \tag{10}$$

For the elaboration of the power and energy model Eq. (11) elaborates the sub-threshold leakage (I_{st}) that contains reverse bias junction current (I_{rbj}).

$$P_{leakage} = v_{dd}I_{st} + |V_{bs}|I_{rbj} \tag{11}$$

Dynamic energy per cycle is represented as E_{dyn} as shown below in Eq. (12):

$$E_{dyn} = C_{eff} \cdot V^2 \cdot f \tag{12}$$

The total consumption of power is due to static and dynamic power that is used to represent the energy consumption of MPSoC cores. Static power is consumed while the CPU is on, whereas dynamic power is spent during calculation periods as shown in Eq. (13).

$$\int_0^{tmax} p\,(t)\ .dt,\ \ P_t = P_{st} + P_{dy} \tag{13}$$

$$E_{Per-Cycle} = f^{-1} * L_g * v_{dd}I_{subn} + |V_{bs}|I_j \tag{14}$$

The leakage energy per cycle is elaborated in Eq. (14). The delay per cycle is denoted as f^{-1} and the required energy for running state per cycle is $E_{running} = f^{-1} *\ \ P_{on}$ enhances when the frequency is low. Eq. (14) represents the total energy consumption per cycle that is represented as:

$$E_{total\ Per-Cycle} = E_{dynamic} + E_{Per-Cycle} + E_{on} \tag{15}$$

The proposed scheduling algorithm works for all real-time jobs in a running state using the system model represented in Fig. 5 allowing all new tasks in the ready state to go to the scheduling phase and execute on time. When compared to the prior strategies utilized in the earliest deadline first algorithm the scheduling methodology improves the chip's overall working performance. When the tasks are ready to run the proposed scheduler verifies that the tasks are compatible with the scheduler and map them to cores based on the utilization factor (u_i). In the proposed system model dynamic fixed priorities are assigned to the entire n periodic task $\tau = \{\tau_{a1}, \tau_{b1}, \tau_{c1} \ldots\ldots \pi_n\}$ with deadline ($d_a = \ p_a$).

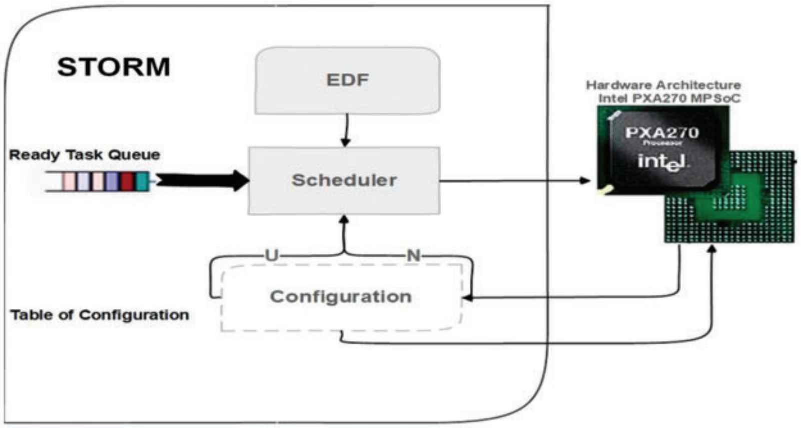

Figure 5: System model

3.2 Core Configurations for EEMST

Scheduling and switching of τ jobs on various cores configurations are based on the utilization factor (u_i). The amount of tasks in the running state requires a suitable task migration policy that defines the criteria for scheduling of *ready* τ on the core. When tasks are running the u_i remains constant but as the number of tasks increases due to the concurrent processing of tasks on the multi-core processor' u_i rapidly increases. For reducing the energy and power consumption optimal energy-efficient core configurations are defined for proper core switching and load balancing using task migration based on this policy. Various core configurations according to the u_i utilization factors are mentioned below:

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{c_a}{p_a} \geq (1\% \rightarrow 9\%), u_i \geq (9.1\% \rightarrow 18\%)$,

$u_i = 7\%, P = 1$ Core configuration {*(1,), (2,),(3), (4),(5),(6),(7),(8)*}

$u_i \geq \ (18.1\% \rightarrow 27\%), \ u_i = 14\%, P = 2$ Core configuration {*(1, 3),(4, 7),(2, 5), (6, 8)*}

$u_i \geq (27.1 \rightarrow 36\%), \ u_i = 24\%, P = 3$ Core configuration {*(1, 3, 5), (4, 7, 8), (1, 2, 6)*}

$u_i \geq (36.1\% \rightarrow 45\%), \ u_i = 31\%, P = 4$ Core configuration {*(1, 3, 57),(2, 4, 6, 8)*}

$u_i \geq (45.1 \rightarrow 54\%), \ u_i = 40\%, P = 5$ Core configuration {(*1, 2, 3, 7, 8),(1, 2, 4, 5, 6)*}

$u_i \geq \ (54.1 \rightarrow 62.5\%), \ u_i = 47\%, P = 6$ Core configuration {*(1, 2, 3, 7, 8), (1, 2, 4, 5, 6)*}

$u_i \geq \ (54.1 \rightarrow 62.5\%), \ u_i = 67\%, P = 7$ Core configuration {*(1, 2, 3, 5, 7, 8), (1, 2, 4, 5, 6, 8)*}

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{c_a}{p_a} > (62.6\%), P \ = 8$ Core configuration {*(1, 2, 3, 4, 5, 6, 7 and 8)*}

Algorithm 1: Proposed Energy Aware-EDF Scheduling with Task Migration

Input: Pall, Psleep, τall , τsize ,Task mappings, Energy and power profiles, migration of task,
τsize —Size of the task set, Pall— Set of total no of CPU, Psleep—CPU to be in sleep mode

(Continued)

Algorithm 1: Continued

```
$\partial$— Configurations of Core selection,
L1—Threshold max_ allowed load of processor, L2—Max load on running core, Lcurrent— Measure the current load of the processor according to the Utilization factor
$\tau$ready—Ready task Set
Ci —Counter 1 = Count time for core selection,
$u_i$—Utilization factor
Configk —Avalible_core configurations in Dual Core System
Config RC $\partial$—Core configurations in running state
Config $\partial$—Selection of next core processor
For $\forall\tau$ ready $\in$ Pall $-$ Psleep
if
    (Selected Configuration with Max_ $u_i <$ threshold $\mu$i )
Lcurrent $<$ L2; then
No change in configuration_Continue the normal execution
else if
     Lcurrent = L2; (Max_ $\mu$i configuration = = Max_temp_allowed)
     No change in P selection_Continue the normal execution
For $\forall\tau \in \tau$ all do
$\tau$= get utilization factor $u_i$ for all $\tau$ ready
 if     Config $\partial$ (selection of new_Core config $\leq$ Number_core config_select) then
        Change the state from sleep mode to idle
else if
  Lcurrent (Maximum current _load of_Configuration = = L1 Max threshold _load on_running core)
Switching of tasks to _ Config $\partial$;
ConfigRC $\partial$ (number of_core config_running = Config $\partial$ (Next_core processor_ config)
end if
until $\tau$ is not $\acute{\varnothing}$
return CPU selected, $\tau$migration
```

4 Simulation and Results

A simulation tool for real-time multiprocessor (STORM) is used to perform the experimental valuation on a homogeneous multiprocessor system and illustrate the importance due to advancements in semiconductor technology that increases the power density of a multiprocessor with the increase in demand of real-time applications based on a complex circuitry. STORM receives ready tasks from the XML input file as shown in the system model and schedules it on the hardware architecture of Marvel X-Scale Intel PXA-270 MPSoC by considering the suitable core configuration with DPM capabilities. Core configurations are chosen based on the utilization factor whereas the power model for all cores is the same.

Intel PXA-270 processor's power model is used for each core in the test scenarios because the Intel-PXA270 offers seven power states. We tested the behavior of our proposed method by running all tasks τ till the arrival of the worst-case execution time (WCET). For this we created a task set and stored the data in an XML input file. The task set's size ranges from 5 to14 tasks with each task's period falling within [0.15 ms, 14 ms] as indicated. We executed the task set mentioned in Tab. 2 on the Intel-PXA270 MPSoC processing platforms. The XML input contains, starts time, WCET, period, priority, deadline, hardware architecture, number of cores and proposed EAEDF scheduling mechanism. The proposed scheduling algorithm using task migration ability is implemented on homogeneous multi-core architecture that has more than one core and shares the same architecture and microarchitecture integrated on a single chip.

Table 2: Task set n = 14

Tasks →	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9	τ_{10}	τ_{11}	τ_{12}	τ_{13}	τ_{14}
Period	0.15	9	6	10	9	6	8	10	7	10	12	13	14	14
Deadline	10	9	6	8	9	8	6	7	6	7	10	5	11	11
WCET	5	3	2	6	5	4	5	4	5	4	5	4	4	4
Priority	1	2	3	4	5	4	4	3	7	3	10	11	5	5
Activation	0	0	0	2	3	4	2	3	0	3	0	5	6	0

Tab. 3 represents various power consumption states of Intel PXA-270 over multiple operating frequencies respectively by considering various states of running, idle and sleep. It also represents the current of the Intel PXA-270 CPU on various frequencies. All hardware characteristics from the Intel-PXA270 processor are employed in our tests to achieve better performance and minimize delays in the execution process. The proposed EA-EDF migrates tasks to a core that is physically away from the core with the highest power consumption and temperature. This section presents the evaluation of our improved energy-efficient scheduling algorithm for the optimization of energy and power. The proposed energy aware-EDF-based scheduler gives improved results and more energy optimization as compared to previous techniques. The Intel PXA-270 MPSoC is used to measure CPU energy usage. According to the proposed algorithm the job with the earliest scheduling deadline is prioritized when the periodic task set is examined the strategy works well that's why the context switch is valuable.

Table 3: Power consumption specification of Intel PXA-270 MPSoC at various frequencies

Frequency	Typical active power	System bus	Idle power	Current
624 MHz	925 mW	208 MHz	260 mW	770 mA
520 MHz	747 mW	208 MHz	222 mW	630 mA
416 MHz	570 mW	208 MHz	186 mW	500 mA
312 MHz	390 mW	208 MHz	154 mW	380 mA
312 MHz	375 mW	104 MHz	109 mW	260 Ma
208 MHz	279 mW	208 MHz	129 mW	150 mA

Considering the same experimental characteristics such as deadlines and consumption. The simulation indicates that the STORM simulator performance criteria are equivalent to the real platform values. Tab. 4 represents the comparison of experimental results of energy consumption on 520 MHz at various $u_i = 6\%, 10\%, 20\%, 36\%, 55\%, 60\%, 62.5\%$ and $u_i \geq 63\%$ when all the cores are running.

Table 4: Energy consumption at different u_i on Intel PXA-270 MPSoC at 520 MHz

Utilization/Active CPU	No of task	Proposed EA-EDF	Active power	Idle power	System bus frequency	Current
$u_i = 6\%$, 1P	1	0.47 J	747 mW	222 mW	208 MHz	630 mA
$u_i = 10\%$, 2P	3	0.85 J	747 mW	222 mW	208 MHz	630 mA
$u_i = 20\%$, 3P	14	1.02 J	747 mW	222 mW	208 MHz	630 mA
$u_i = 36\%$, 4P	16	2.31 J	747 mW	222 mW	208 MHz	630 mA
$u_i = 55\%$, 5P	19	3.05 J	747 mW	222 mW	208 MHz	630 mA
$u_i = 60\%$, 6P	24	3.89 J	747 mW	222 mW	208 MHz	630 mA
$u_i = 62.5$, 7P	26	4.20 J	747 mW	222 mW	208 MHz	630 mA
$u_i \geq 63\%$, All P	27	4.10 J	747 mW	222 mW	208 MHz	630 mA

Tabs. 5 and 6 represents the comparison of proposed EA-EDF at various workloads u_i as compared to other currently used techniques on 520 and 624 MHz. The simulation results show that the proposed EA-EDF is more energy-efficient as compared to other techniques at lower utilization and gives 4.7% & 4.3% more energy efficient results on 10%, 27% & 36% u_i as compared to GEDF, RT-DPM, TBP, PDTM on 520 & 624 MHz in terms of power & energy. The Intel PXA-270 MPSoC is used to measure CPU energy usage. The proposed EA-EDF algorithm prioritized the job with the earliest scheduling deadline.

Table 5: Energy consumption comparison of different schedulers at 520 MHz

Utilization	No of task	Proposed EA-EDF	Energy GEDF	Energy PDTM	Energy Uniform RT-DPM	Energy TBP	Active power mW	Idle power mW	System bus MHz
$u_i = 6\%$	1	0.57 J	2.41 J	0.66 J	2.51 J	2.88 J	747	222	208
$u_i = 10\%$	3	0.75 J	3.81 J	3.97 J	4.87 J	3.02 J	747	222	208
$u_i = 20\%$	14	1.02 J	5.73 J	5.41 J	5.21 J	4.77 J	747	222	208
$u_i = 36\%$	16	2.31 J	6.12 J	6.89 J	6.79 J	6.22 J	747	222	208
$u_i = 55\%$	19	3.05 J	6.72 J	5.2 J	7.88 J	7.13 J	747	222	208
$u_i = 60\%$	24	3.89 J	7.23 J	7.44 J	9.21 J	7.72 J	747	222	208
$u_i = 62.5$	26	4.20 J	8.11 J	8.76 J	10.96 J	8.38 J	747	222	208
$u_i \geq 63\%$	27	8.10 J	8.34 J	9.38 J	11.12 J	9.43 J	747	222	208

Table 6: Energy consumption comparison of different schedulers at 624 MHz

Utilization factor	No of task	Proposed EA-EDF	Energy GEDF	Energy PDTM	Energy uniform RT-DPM	Energy TBP	Active power mW	Idle power mW	System bus MHz
$u_i = 6\%$	1	0.57 J	1.41 J	0.88 J	0.66 J	0.51 J	925	260	208
$u_i = 10\%$	3	0.95 J	1.81 J	1.02 J	0.97 J	0.87 J	925	260	208
$u_i = 20\%$	14	1.12 J	2.73 J	1.77 J	1.41 J	3.21 J	925	260	208
$u_i = 36\%$	16	1.51 J	3.12 J	3.22 J	1.89 J	3.79 J	925	260	208
$u_i = 55\%$	19	3.13 J	6.12 J	4.13 J	3.12 J	3.88 J	925	260	208
$u_i = 60\%$	24	4.09 J	7.23 J	5.32 J	4.44 J	4.21 J	925	260	208
$u_i = 62.5$	26	4.50 J	8.11 J	6.38 J	5.76 J	4.96 J	925	260	208
$u_i \geq 63\%$	27	4.60 J	8.34 J	7.43 J	7.38 J	5.12 J	925	260	208

The comparison of energy consumption at various utilization factors u_i on 520 MHz as shown in Fig. 6 illustrates the results of the proposed energy-aware EA-EDF scheduler that behaves similar to GEDF and PDTM *for* $\forall\tau \rightarrow u_\tau => 62.5\%$, but optimizes 4.7% of the overall energy at *for* $\forall\tau \rightarrow u_\tau < 62.5\%$.

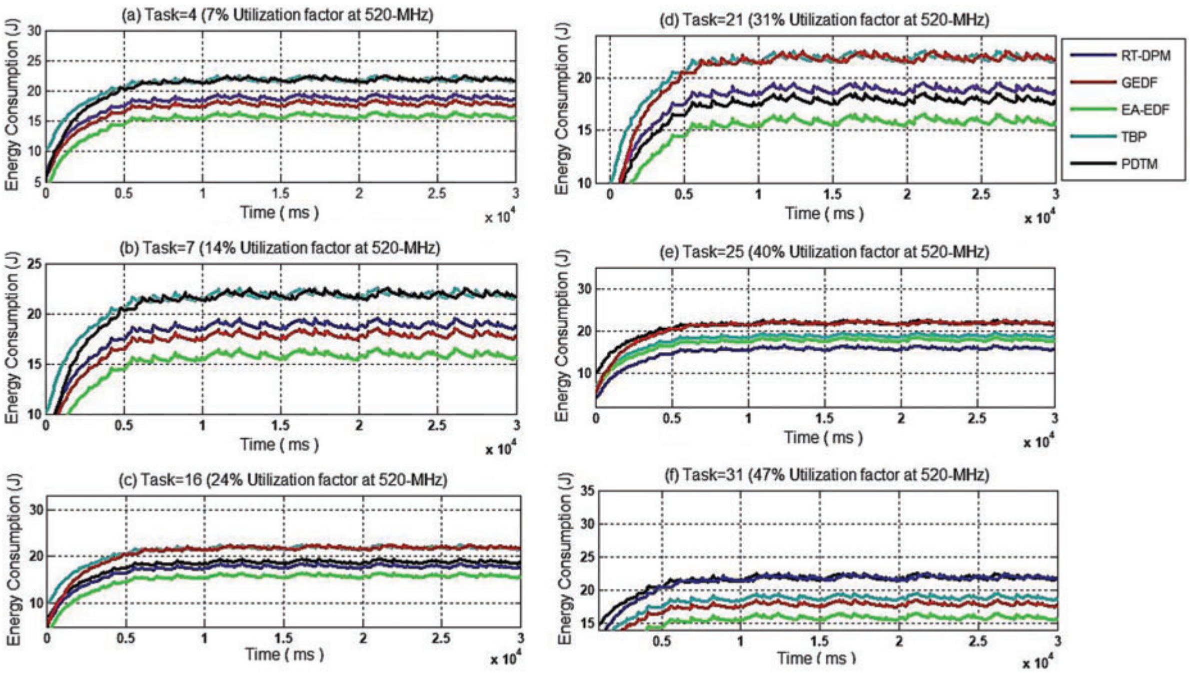

Figure 6: Comparison of energy consumption u_i at 520 MHz

Fig. 7 represents the power consumption on various cores of MPSoCs with the increasing number of tasks using EA-EDF at $u_i = 36$. Only 4 cores $Core_{1,}$ $Core_{2,}$ $Core_3$ & $Core_{4,}$ are running by intelligently migrating and switching the load towards the low energy cores before u_i meeting the threshold. The features of the task set include the software architecture, start time and worst-case execution time, period, priority and deadline. While hardware architecture contains the no of cores and the proposed scheduling algorithm that is defined in the XML file for achieving the experimental results in Fig. 8.

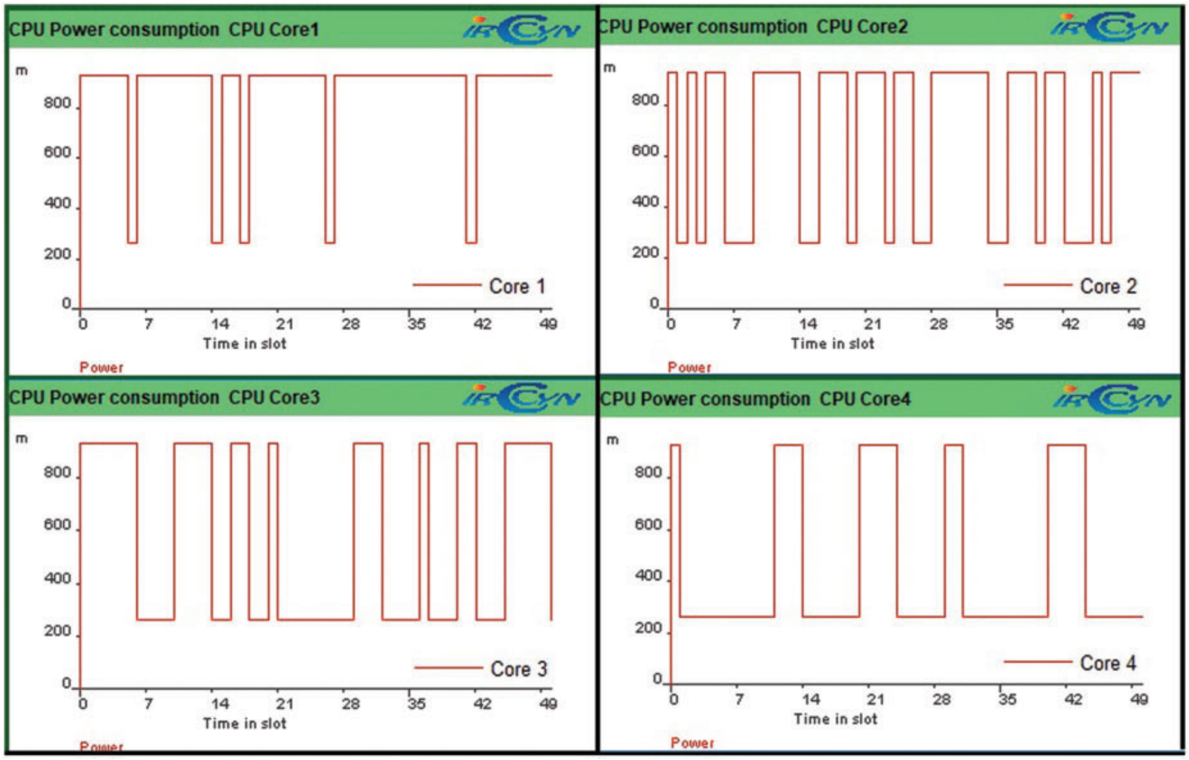

Figure 7: CPU power consumption at various CPU cores under $u_i = 36\%$ utilization at 520 MHz

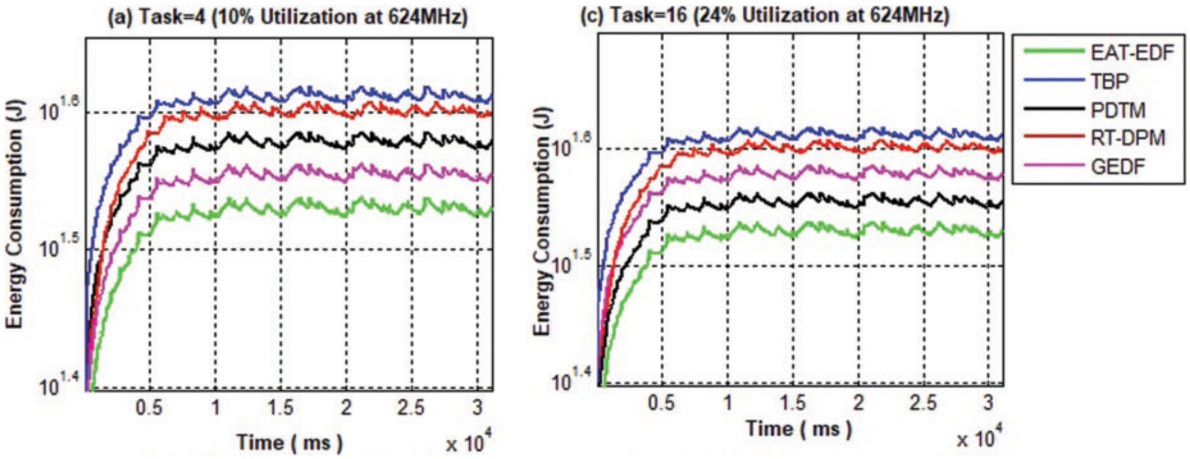

Figure 8: (Continued)

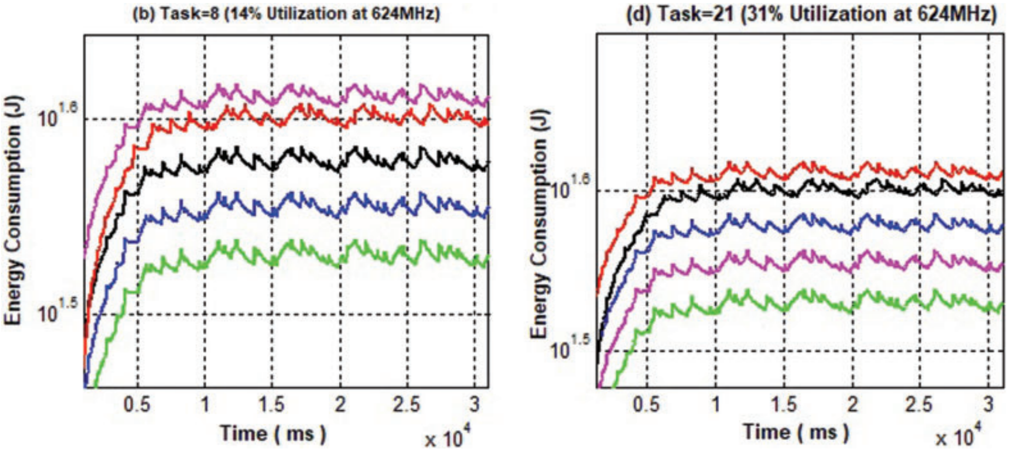

Figure 8: Comparison of energy consumption u_i at 624 MHz

The comparison of energy utilization of EA-EDF at $u_i = 6\%, 10\%, 20\%, 36\%, 55\%, 60\%, 62.5\%$ and $u_i \geq 63\%$ when all the cores are in running state on 624 MHz as shown in Fig. 8. The proposed energy-aware EA-EDF scheduler had an impact on the system performance and enhances the life span of the chip by optimizing the average energy of $4.7\%\, for\ \ \forall\tau \rightarrow \mathrm{u}_\tau < 36.5\%$ by giving smoother energy utilization patterns between individual cores for a set of periodic R-T tasks with a common deadline.

5 Conclusions

This research paper targets energy-efficient multiprocessor scheduling problems over $\tau i \in \tau$ over $m_{identical}$ homogeneous processors for a set of periodic *R-T* tasks with a common deadline for energy optimization for this problem we have developed an accurate energy optimization-based task migration policy that will be used to calculate the load of the destination core. For the selection of cores, various configurations have been proposed in terms of energy efficiency and utilization factor that enables a processor to reduce power and energy consumption by adopting a suitable task migration policy. An increase in power utilization reduces the life span of the chip and has become an integral chip design issue for the battery-operated multiprocessor system. The proposed EA-EDF model is based on an optimal scheduling policy for accurate migration of tasks without missing their deadlines by considering the hardware architecture of Intel PXA-270 MPSoC in the STORM simulator. The objective is to increase the performance of the homogenous multi-core systems by gradually decreasing the thermal cycles, power and energy consumption. whereas Less number of running cores are combined in configuration. Proposed EA-EDF algorithm enables load balancing and allows processes to run on various cores at various times as decided by the scheduler using periodic task sets with implicit deadlines on homogeneous multiprocessor platforms. In future work, our algorithm can be used for heterogeneous multiprocessor platforms apart from this the proposed technique gives us more efficient results by optimizing 4.3%–4.7% energy on a utilization of 6%, 36% & 46% at 520 & 624 MHz operating frequency when compared to previously deployed GEDF, TLP, PDTM, RT-DPM TBP energy-based optimization techniques.

Acknowledgement: The researchers wish to extend their sincere gratitude to the Deanship of Scientific Research at the Islamic University of Madinah for the support provided to the Post-Publishing Program1.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Somdip, S. Isuwa, S. Saha, A. K. Singh and K. M. Maier, “CPU-GPU-memory DVFS for power-efficient MPSoC in mobile cyber physical systems,” *Future Internet*, vol. 14, no. 3, pp. 91–103, 2022.

[2] R. A. prasath, J. Alzubi and M. Ramachandran, “A high performance scalable fuzzy based modified asymmetric heterogene multiprocessor system on vhip (AHt-MPSOC) reconfigurable architecture,” *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 2, pp. 647–658, 2022.

[3] T. Yu, R. Zhong, V. Janjic, P. Petoumenos and J. Zhai, “Collaborative heterogeneity-aware os scheduler for asymmetric multicore processors,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1224–1237, 2020.

[4] H. Wang, X. Guo, S. X. D. Tan, C. Zhang and H. Tang, “Leakage-aware predictive thermal management for multicore systems using echo state network,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 7, pp. 1400–1413, 2019.

[5] S. Z. Sheikh and M. A. Pasha, “A dynamic cache-partition schedulability analysis for partitioned scheduling on multicore real-time systems,” *IEEE Letters of the Computer Society*, vol. 3, no. 2, pp. 46–49, 2020.

[6] M. Shan and O. Khan, “Accelerating concurrent priority scheduling using adaptive in-hardware task distribution in multicores,” *IEEE Computer Architecture Letters*, vol. 20, no. 1, pp. 17–21, 2020.

[7] J. Feliu, J. Sahuquillo and S. Petit, “Thread isolation to improve symbiotic scheduling on smt multicore processors,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 2, pp. 359–373, 2019.

[8] P. Fruehauf, A. Munding, K. Pressel, P. Schwarz and M. Vogt, “Chip-package-board reliability of system-in-package using laminate chip embedding technology based on cu leadframe,” *IEEE Transactions on Components Packaging and Manufacturing Technology*, vol. 10, no. 1, pp. 44–56, 2019.

[9] W. Alberto, E. Kim, R. Renato and R. Andre, “Power consumption analysis in static cmosgates,” in *26th Symp. on Integrated Circuits and Systems Design (SBCCI)*, Curitiba, Brazil, pp. 1–6, 2013.

[10] G. Khadidja, B. M. Kamel, B. Aboue and A. K. Singh, “A new efficient multi-task applications mapping for three-dimensional network-on-chip based MPSoC,” *Concurrency and Computation Practice and Experience*, vol. 1, no. 10, pp. 1–20, 2021.

[11] R. A. Prasath, J. A. Alzubi and M. Ramachandran, “A high performance scalable fuzzy based modified asymmetric heterogene multiprocessor system on chip (AHt-MPSOC) reconfigurable architecture,” *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 2, pp. 647–658, 2022.

[12] L. Chenyun, X. Li, C. Tang and Q. Wu, “A power management circuit compatible with Ti smart reflex and xilinx ultrascale MPSoC dual voltage mechanism,” in *10th Int. Conf. on Computing and Pattern Recognition*, Shanghai, China, pp. 357–362, 2021.

[13] E. Jiang, L. Wang and J. Wang, “Decomposition-based multi-objective optimization for energy-aware distributed hybrid flow shop scheduling with multiprocessor tasks,” *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 646–663, 2021.

[14] C. Y. Liang, S. Liu, E. Y. Chung and J. L. Gaudiot, “An energy and performance efficient dvfs scheme for irregular parallel divide-and-conquer algorithms on the intel scc,” *IEEE Computer Architecture Letters*, vol. 13, no. 1, pp. 13–16, 2013.

[15] A. Italo, O. Antonio, T. Barros, I. M. Sardina, C. P. Bianchini *et al.,* “Distributed-memory load balancing with cyclic token-based work-stealing applied to reverse time migration,” *IEEE Access*, vol. 7, no. 1, pp. 128419–128430, 2019.

[16] K. Baital and A. Chakrabarti, “Dynamic scheduling of real-time tasks in heterogeneous multicore systems,” *IEEE Embedded Systems Letters*, vol. 11, no. 1, pp. 29–32, 2018.

[17] K. Huang, K. Wang, D. Zheng, X. Jiang, X. Zhang *et al.,* “Expected energy optimization for real-time multiprocessor socs running periodic tasks with uncertain execution time,” *IEEE Transactions on Sustainable Computing*, vol. 6, no. 3, pp. 398–411, 2018.

[18] B. James, V. Tenentes, B. M. AlHashimi and G. V. Merrett, “Online tuning of dynamic power management for efficient execution of interactive workloads,” in *IEEE/ACM Int. Symp. on Low Power Electronics and Design (ISLPED)*, Boston, United States, pp. 1–6, 2017.

[19] B. G. Maryam, “MPSoC based dynamic power management in wireless sensor networks,” in *Int. Conf. on Information Communication and Embedded Systems*, Chennai, India, pp. 1–6, 2014.

[20] P. Bogdan, R. Marculescu and S. Jain, “Dynamic power management for multidomain system-on-chip platforms: An optimal control approach,” *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 18, pp. 1–20, 2013.

[21] W. Yankai, S. Wang, B. Yang, L. Zhu and F. Liu, “Big data driven hierarchical digital twin predictive remanufacturing paradigm: Architecture, control mechanism, application scenario and benefits,” *Journal of Cleaner Production*, vol. 248, no. 1, pp. 119–299, 2020.

[22] J. Haris, M. Shafique, J. Henkel and S. Parameswaran, “System-level application-aware dynamic power management in adaptive pipelined MPSoCs for multimedia,” in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, Washington, D.C, pp. 616–623, 2011.

[23] K. George and D. Pnevmatikatos, “Dynamic power and thermal management of noc-based heterogeneous mpsocs,” *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 7, no. 1, pp. 1–26, 2014.

[24] S. E. Abdellaoui and Y. Fakhri, “Power management strategies in energy-harvesting wireless sensor networks,” *International Journal of Communication Networks and Information Security*, vol. 13, no. 1, pp. 136–142, 2021.

[25] Y. Yongkun, Y. Zhao and Y. Liu, “Recent progress in organic field-effect transistor-based integrated circuits,” *Journal of Polymer Science*, vol. 60, no. 3, pp. 311–327, 2022.

[26] X. Zhang, X. Chen, W. Sun and X. Z. He, “Vehicle re-odentification model based on optimized densenet121 with joint loss,” *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3933–3948, 2021.

[27] Z. Xiaorui, W. Zhang, W. Sun, H. Wu, A. Song *et al.,* “A Real-time cutting model based on finite element and order reduction,” *Computer Systems Science and Engineering*, vol. 43, no. 1, pp. 1–15, 2022.

[28] P. M. Kumar and R. Nagendra, “Simulation of real time multiprocessor static scheduling algorithms,” *I-Manager’s Journal on Embedded Systems*, vol. 7, no. 1, pp. 30–38, 2018.

[29] U. Richard, A. M. Deplanche and Y. Trinquet, “Storm a simulation tool for real-time multiprocessor scheduling evaluation,” in *IEEE 15th Conf. on Emerging Technologies & Factory Automation (ETFA)*, Bilbao, Spain, pp. 1–8, 2010.

[30] C. A. Kivilcim, T. S. Rosing, K. Mihic and Y. Leblebici, “Analysis and optimization of mpsoc reliability,” *Journal of Low Power Electronics*, vol. 2, no. 1, pp. 56–69, 2006.