Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032999
Article

Tech Science Press

An Optimal DPM Based Energy-Aware Task Scheduling for Performance Enhancement in Embedded MPSoC

Hamayun Khan[1,*], Irfan Ud Din[2], Arshad Ali[3] and Mohammad Husain[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: 03 June 2022; Accepted: 07 July 2022

Abstract: Minimizing the energy consumption to increase the life span and performance of multiprocessor system on chip (MPSoC) has become an integral chip design issue for multiprocessor systems. The performance measurement of computational systems is changing with the advancement in technology. Due to shrinking and smaller chip size power densities on-chip are increasing rapidly that increasing chip temperature in multi-core embedded technologies. The operating speed of the device decreases when power consumption reaches a threshold that causes a delay in complementary metal oxide semiconductor (CMOS) circuits because high on-chip temperature adversely affects the life span of the chip. In this paper an energy-aware dynamic power management technique based on energy aware earliest deadline first (EA-EDF) scheduling is proposed for improving the performance and reliability by reducing energy and power consumption in the system on chip (SOC). Dynamic power management (DPM) enables MPSOC to reduce power and energy consumption by adopting a suitable core configuration for task migration. Task migration avoids peak temperature values in the multi-core system. High utilization factor (u_i) on central processing unit (CPU) core consumes more energy and increases the temperature on-chip. Our technique switches the core by migrating such task to a core that has less temperature and is in a low power state. The proposed EA-EDF scheduling technique migrates load on different cores to attain stability in temperature among multiple cores of the CPU and optimized the duration of the idle and sleep periods to enable the low-temperature core. The effectiveness of the EA-EDF approach reduces the utilization and energy consumption compared to other existing methods and works. The simulation results show the improvement in performance by optimizing 4.8% on u_i 9%, 16%, 23% and 25% at 520 MHz operating frequency as compared to other energy-aware techniques for MPSoCs when the least number of tasks is in running state and can schedule more tasks to make an energy-efficient processor by controlling and managing the energy consumption of MPSoC.

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

1 Introduction

An Embedded system is defined as a system with dedicated functionality with software utilities embedded into the hardware to perform the dedicated task. Moreover a typical embedded system can be viewed as a base hardware platform that executes applications, software and any peripheral devices connected to different ports [1]. The hardware platform consists of the processing units like CPU, graphics processing unit (GPU) communication channels and memory elements. In an operating system application-level code belongs to software implementation while power efficiency can be achieved by controlling power consumption in all the parts of the overall system [2].

In recent years embedded systems are growing phenomenally in terms of new features and applications. Mobile embedded devices now offer the integration of multimedia applications, wireless data features, internet browsing and phone. In addition, these new trends also urge managing power consumption. Further, it's known that the use of various electronic gadgets based on MPSoCs like mobile devices has now reached approximately equal to the world's population [3].

Designing an embedded system that gives high efficiency while dissipating low power is a major challenge for embedded system designers [4]. Power management in embedded devices is intended to improve effciency and performance in terms of energy in embedded systems. Due to advancements in computational embedded devices the execution of the task is eventually easier than ever before using multiprocessors [5,6].

An energy-aware online scheduling-based technique is used for high-performance homogeneous systems that consider both spatial and temporal gradients and their correlation with temperature evolution that enhances the system's performance while dynamic voltage and frequency scaling (DVFS) is used for the smooth execution of tasks [7]. Scheduling is based best speed fit earliest deadline first (EDF) technique that chooses the suitable processor when the task is allocated for execution and the proposed technique doesn't consider the fastest core. The performance and reliability of the chip are improved than that of the global earliest deadline first algorithm that considers the priority of the task while executing [8]. Migration of tasks can occur at any time with different speed processors and tasks were assigned based on deadlines those tasks that can have an early deadline can have higher priority [9].

Chip size is decreasing and the number of transistors on the chip is gradually increasing and the dimensions of the chip are getting smaller than before. The increased temperature on the chip affects the reliability and performance of the Circuit that is used in the embedded device [10,11]. Introduced a technique in which power can be reduced without changing the supply voltage by using the frequency scaling technique while the amount of overall power is the same because the energy can be decreased by Voltage scaling [12].

Most of the embedded computing circuits aim to give a maximum performance while using minimum power. Electronic circuits are viewed as a combination of different CMOS devices. CMOS devices are the basic building block of all computing systems. CMOS semiconductors uses positive channel metal-oxide-semiconductor (PMOS) or negative channel metal-oxide-semiconductor (NMOS) semiconductor PMOS. CMOS circuits consume less power as compared to other devices which are using just one type of transistor. This property makes them unique for embedded systems which are mostly used in battery-operated [13].

CMOS chips are the foundation of today's MPSoC. A device power consumption is divided into two categories dynamic and static. Dynamic is also known as switching power and static is also known as leakage power. Leakage power became the primary power consumer in designs smaller than 90nm whereas switching is the main contribution in larger designs greater than 90nm. Both forms of power can be reduced using power reduction [14,15]. Energy-efficient scheduling techniques for multi-processor are introduced with an EDF scheduling algorithm that extends the battery life. EDF can be extensively used for both offline and online scheduling. Offline scheduling is inflexible and has deterministic timing behavior for a deterministic system. In offline scheduling complexity is not important and have low run-time overhead. In online scheduling the parameters of each task are known when the job is released and is widely used for unpredictable workloads [16].

For many years the primary factor in CMOS circuits has been dynamic power dissipation which happens as a result of transistor switching $P_{dynamic}$ is also known as switching power and it can be calculated using Eq. (1):

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

C_a is the load capacitance, N_{cs} is the average number of circuit transitions per clock cycle while V is the voltage supplied which is equal to the switching voltage in most cases and f is the clock frequency. Lowering the supply voltage increases circuit latency that causes circuits to fail especially in heterogeneous systems multiple types of cores support each other [17]. Eq. (2) shows that the frequency of the clock is almost linearly proportional to the source voltage.

$$f = K.\frac{V_{dd} - V_{ths}}{V_{dd^2}} \tag{2}$$

where V_{ths} is the threshold voltage and k is a constant. The decrease in chip size in recent technological generations allowed for a significant reduction in supply voltage V_{dd} resulting in a quadratic reduction in power consumption [18]. The energy consumption (E_{cc}) of executing a task is defined in Eq. (3):

$$E_{cc} = P_{switching}.t = C_{eff}.V^2.f\frac{C_b}{f} = C_{eff}.V^2.C_b\frac{V_{dd} - V_{ths}}{.V_{dd^2}} \tag{3}$$

where t is the actual task executing time and C_b is the actual number of execution cycles for a task among static dynamic and short circuit power dynamic power is the most dominant component of all in MPSoCs [19].

The main contributions of this paper elaborates and mathematical prove the concept of energy efficiency using scheduling. A peak-Energy and power-aware energy management task scheduling scheme that is conducted at offline phase using our proposed EA-EDF scheduling algorithm for optimizing the energy.

In this paper mainly proposing task switching techcnique using EDF scheduling mechanism to manage the energy and peak power consumption and achieve further reduction in energy/peak in the MPSoC.

2 Literature Review

CPUs with more transistors are now available due to rapid development in semiconductor technology [20]. High-end CPUs are widely used in real-time embedded systems however total chip power consumption in CMOS is considerably increasing with rapid growth in technology can be calculated using Eq. (4) mentioned below.

$$P_{total} = P_{static} + P_{dynamic} \tag{4}$$

The static power consumption P_{static} is computed by adding the leakage and short current power while the chip is in working condition the $P_{dynamic}$ the parameter represents the charging and discharging of the output capacitance [21]. It's difficult to reduce static power usage which is influenced by a variety of factors in the semiconductor manufacturing process. Eq. (5) represents the reduction in dynamic power consumption shown below.

$$P_{dynamic} = C.K.V_{dd}.f \tag{5}$$

where f represents frequency V represents supply voltage and C denotes capacitive load and K is the proportionality constant. The DVFS approach uses the slack time that arises when scheduling tasks to adjust the voltage and frequency of the CMOS chip [22]. DPM technique is reducing energy losses when the slack time arrives reducing energy usage by transitioning to a low power state. Energy overhead is imposed due to break-even time increasing gradually. There will be a loss as a result of the state transfer [23,24]. The below Eq. (6) determines the break-even time (BET) sleep.

$$BET_{sleep} = \max\left(t_{pa}\frac{E_{Pa-}P_{sleep}.t_{pa}}{P_{idle} - P_{sleep}}\right) \tag{6}$$

Recognizing the significance of low power and energy consumption and thermal concerns in multiprocessor systems. Several strategies for managing real-time scheduling while addressing thermal limits were previously presented to regulate peak temperature, energy and performance. The majority of them use DPM and DVFS [25]. EDF scheduling algorithm can have dynamic priorities. The scheduler can select those tasks that are in the ready queue whose deadlines are coming next and are very useful for the uni-processor. The utilization factor of the process is less than is approximately equal to 1. In hard real-time systems, these jobs are schedulable for a preemptable task and can attain 100% utilization for a central processing unit. Using the EDF scheduling technique the utilization of the CPU improves because of the implication of dynamic priority. The priority of selecting each task depends on the fixed deadlines [26,27].

2.1 Energy Monitoring Phase in MPSOCs

In this section the mechanism for energy monitoring is discussed using Intel PXA270 MPSoC. Intel PXA270 processor has a variety of operating modes. For optimization of energy, there are three modes active, idle and sleep. Power modes are drastically different in terms of their capabilities including the transition time according to the energy required to perform. The power consumption for various states of Intel PXA270 is Mined from the datasheet of the Intel® PXA270 Processor Electrical, Mechanical and Thermal Specification used for enabling the low-power mode [28]. Several energy-saving features are built into the CPU. The four main energy-saving modes are the ones that stand out the most. Tab. 2 demonstrates the consumption of the Intel PXA270 MPSoC on various power modes as well as the nominal current consumption. In the STORM Simulator's XML specification input a tag named CPU is used to represent this type of processor in simulation [29].

2.2 Higher Power Consumption Causes Thermal Problems

This section elaborates the issues causes due to high task utilization that mainly leads to higher possible power consumption. Higher power consumption on-chip causes higher temperature and higher resistance that lower the possible speed of MPSoC. Lowering the temperature on the chip reduces the resistance lowers the power consumption and causes a higher possible speed. Fig. 1 illustrates the effects of high-power dissipation that produces heat because of (energy consumption/loss or waste) as an undesirable derivative of its primary action illustrates that chip power dissipation is increasing as the processing capabilities of MPSoCs grow resulting in a large increase in chip temperature. As the chip size is shrinking and the numbers of transistors are increasing per unit area on the chip this leads to more power consumption, complexity and increases in the temperature of the chip. The designers of MPSoCs face significant difficulty in achieving the maximum performance system with high utilization [30].

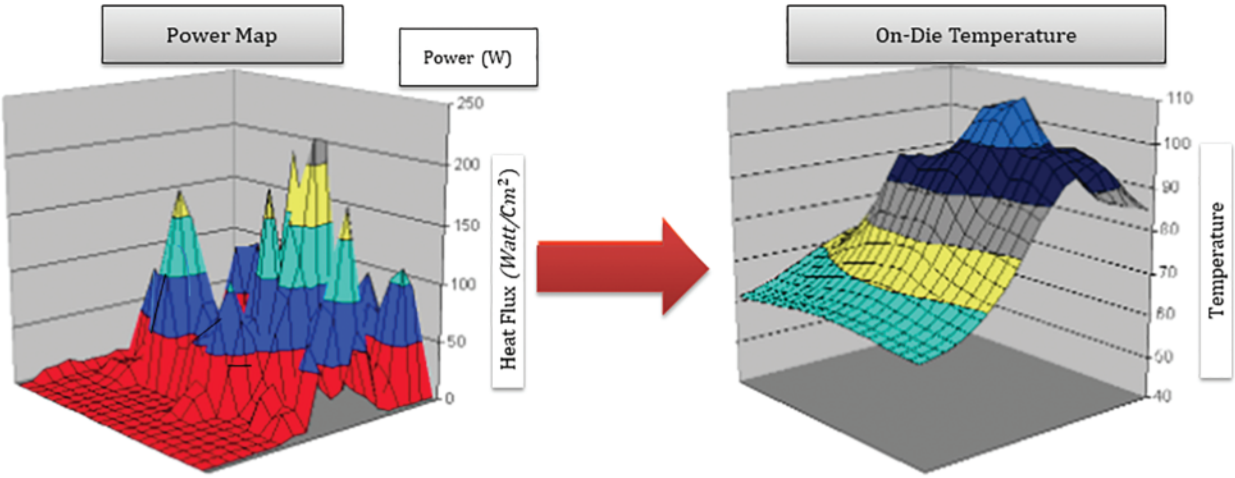

Figure 1: Effects of high power dissipation produce thermal upsurge

3 Problem Statement

Power consumption, scheduling and switching of tasks from one core to another core is one of the major issues in today's MPSoC. Given a system model having various CPUs are represented by *m* and a periodic task contains the power model of MPSoC and a task model that normally schedules the workload on each core individually. The optimal solution includes a task migration and core switching technique to reduce the energy consumption of the system.

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 $\tau i \in \tau$ 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 $\tau i \in \tau$ over $\mathrm{m}_{\mathrm{identical}}$ to complete before the deadline by allowing the suitable task migration among processors and core switching technique.

3.1 Proposed EA-EDF & Task Model

This section elaborates the methodology that is opted for the proposed scheduling technique and task model. The proposed system model gives a comprehensive description of the elements that are involved in conducting experiments. The major components include a task generator, simulator, core configurations and scheduling algorithm where users can create random task sets in xml.txt file that contains different parameters. The parameters in the extensible markup language (XML) file are used as an input for the simulation tool including period, activation time, deadline, worst-case execution time (WCET), priority and best-case execution time (BCET). Task set generator contains random task sets that are used to create the workloads under different constraints as well as the no of CPU cores set by the user in XML according to the application provided to STORM as an input. The scheduler can have a set of ready tasks and simulate the input file according to the scheduling policy by considering task migration and task balancing according to the configuration policy for cores that are developed in the proposed algorithm producing the power profiles of the individual core according to the given set of parameters in XML.

Fig. 2 illustrates the proposed EA-EDF scheduler that organizes resources to meet the deadlines and forecast the utilization factor. The power module is integrated as part of the scheduling policy. The STORM Simulator considers each processor to be an execution unit that runs the task. As a result the scheduling method determines the frequency and power states (idle, waiting). To schedule the n tasks, the EA-EDF scheduling approach employs the task scheduling policy and allocates a single processor for each subset after the scheduler assigns work. Task migration allows the scheduler to place some high-power running cores of processors in idle mode and enable other processors that were previously idle. If a job with a higher priority is waiting in the ready task queue the present process is preempted and placed in idle mode. The work in the ready task queue with the shortest deadline receives the highest priority if two jobs have the same priority at time t. The Input XML file can have all the information about the complete simulation duration for the execution of experiments, hardware resources that are required according to the load, architecture for task set and scheduling algorithm. Simulations will be considered using the Intel PXA270 multiprocessor which is preferable because the processor has an intelligent power saving mechanism.

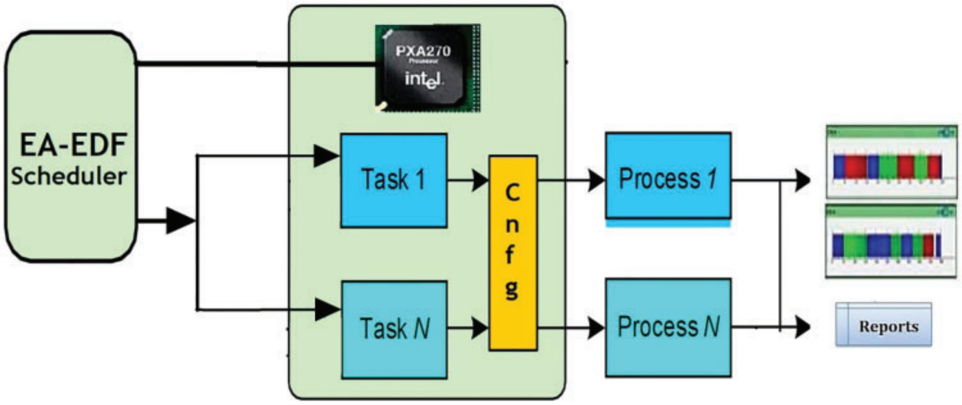

Figure 2: Proposed EA-EDF scheduling model

We adopt a classic real-time task model for this by considering an Intel PXA270 MPSoC platform denoted by π and m-identical processor platforms having various processors denoted, m, m is the number of processors used and is given as a parameter. $\pi = \{\pi_a, \ldots\ldots.\pi_m\}$ a periodic task model denoted byτ, that contains a ready task τ_{ready} defined by a_{0x} c_a, $p_{a,}$, d_a where a_{0x} is the activation time, C_a represents WCET, P_a is the period, d_a the relative deadline, P_{0x} is the priority of the task.

As our proposed scheduler is dynamic so priority will remain fixed for all tasks and the task can be instantiated an infinite number of times. The task set contains n periodic task $\tau = \{\tau_a, \tau_b, \tau_c \ldots\ldots.\pi_n\}$ with a constrained deadline $(d_a = p_a)$. In this research, we assume that the task deadlines are equal to the period (d_a, p_a) and the tasks are scheduled by the proposed EDF scheduling policy. Moreover tasks are assumed to be independent and preemptable as we are using an implicit task model that considers preemption at any interval during task scheduling.

3.2 CPU Power/Energy Model

This section elaborates the power model that calculates the static and dynamic consumption of power in CMOS circuits. Eq. (7) illustrates the dynamic power consumption $P_{dynamic}$ of CMOS circuits.

$$P_{dynamic} = C_{eff}.V^2.f \tag{7}$$

where V is the required supply voltage, while the frequency component is represented as f while C_{eff} shows the effective switching capacitance. DVS reduces the consumption of dynamic power elements due to quadratic dependence on voltage. Subthreshold leakage is the major contributor to leakage that can enhance significantly with adaptive body biasing. The threshold voltage($V_{th)}$, sub-threshold current (I_{sub-th}) and cycle time t_{cycle} are considered a function of the V_{dd} and V_{bb} as the bias voltage is represented as in Eq. (8).

$$V_{th} = V_{th1} - K_1, V_1 - K_2 \tag{8}$$

whereK_1, $V_1 - K_2$ and V_{th1} are technology constants. The processor power model follows a suitable core configuration. Specifically, the C-states contain one active and rest as low power state (sleep) and deep sleep states. The P-states contain active states and mainly differ in power/energy consumption and operating frequency. Subthreshold leakage (I_{subn}) contains (I_j) as reverse bias junction current is expressed below in Eq. (9).

$$P_{leakage} = v_{dd}I_{subn} + |V_{bs}|I_j \tag{9}$$

Eq. (10) represents the consumption of energy over a period [$\mathrm{t_a}$, $\mathrm{t_b}$] can be calculated using:

$$E = \int_{\mathrm{ta}}^{\mathrm{t_b}} p\,(t)\,.dt, \tag{10}$$

The consumption of energy over workload w_i that requires to be scheduled on a processor with a speed s_a, is integral of power over the length as shown in Eq. (11).

$$E\,(W, S) = \left(V.I_{le} + C_a.V^2.f\right).(w_i/s_a) \tag{11}$$

Energy consumption of cores of the MPSoC is represented as static power and dynamic power. As long as the CPU is in on state static power is consumed while dynamic power is consumed during computation times as shown in Eq. (12).

$$\int_0^{tmax} p\,(t)\,.dt, P_{total} = P_{static} + P_{dynamic} \tag{12}$$

$P_{dynamic}$ depicts the charging and discharging of the output capacitance while the chip is working and is related to the switching activity α and c_a load capacitance V represents the supply voltage and f defines operational frequency as mentioned below in Eq. (13).

$$P_{dynamic} = C_a.V^2.f \tag{13}$$

Static power was approximately negligible in earlier processors: For n no of transistors the parameter that is considered as design-dependent is represented as K_{design}, while $\mathrm{I_{leak}}$ is defined as technology-dependent $P_{dynamic}$ parameter. The dynamic energy per cycle E_d and the leakage energy per cycle is defined in Eqs. (14) and (15) respectively.

$$E_{dynamic}= C_{eff}.V^2.f \tag{14}$$

$$E_{Per-Cycle} = f^{-1} * L_g * v_{dd} I_{subn} + |V_{bs}| I_j \tag{15}$$

where f^{-1} represents the delay/cycle. The energy required to keep the system in an ON state per cycle is represented as $E_{on} = f^{-1} * P_{on}$ increases with an increase in lower frequencies are equal to the total energy consumption per cycle as defined in Eq. (16).

$$E_{total\ Per-Cycle} = E_{dynamic} + E_{Per-Cycle} + E_{on} \tag{16}$$

The original goal of EA-EDF and DPM is to reduce the energy consumption by switching the processor to low power state. Generally, in MPSoC consumption of energy gives a higher hierarchical view of energy saving so it is obvious to measure the energy. However, the impact of a EA-EDF scheduling algorithm and the workload is investigated on different components of the hardware mainly CPU because its a major energy-consuming component. The energy consumption of the CPU decreases with an increase in sampling periods of a system with tasks $\tau = \{\tau_a, \tau_b, \tau_c \ldots\ldots \pi_n\}$ by reducing the workload that results in energy optimization when energy-aware EDF is employed on the energy consumption function of CPU h_1 and h_2 as shown below in Fig. 3.

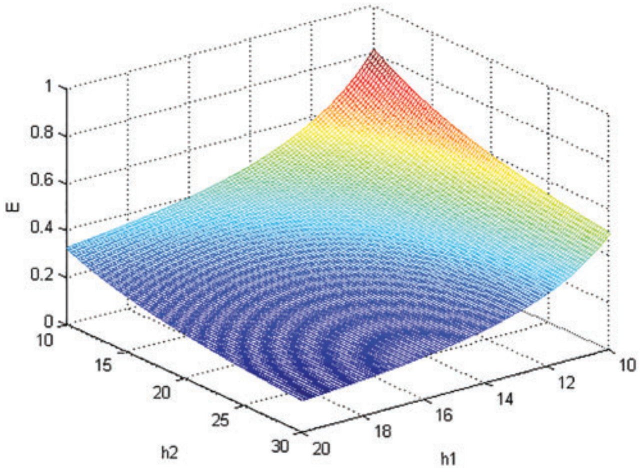

Figure 3: Normalized CPU energy consumption using proposed EDF scheduling model

3.3 *Core Selection and Task Scheduling Classification*

In this section the selection of core is elaborated as the operational frequency and supply voltage of a CMOS device determine its power consumption as shown in Eq. (17) is representing the association between CPU power consumption and supply voltage:

$$P_{dynamic} \propto V^2 \tag{17}$$

CPU requires higher supply voltage while working at a higher frequency as compared to processor operating at a lower frequency stated in Eq. (18). As a result CPUs have the higher capacity by supplying the higher voltage.

$$f \propto \left(\frac{V - V_{threhold}}{V}\right)^{\alpha} \tag{18}$$

In CMOS transistor f is the frequency where $V_{threhold}$ is the threshold voltage of transistors and α is a measure of the velocity saturation. All the hardware and software parameters are used from Marvell's X-Scale technology based on the Intel PXA270 processor.

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 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. In this research, the researcher is working on an EDF scheduling algorithm that is responsible to schedule tasks on various configurations depending on the utilization factor of the Cores. Selection of core depends on the number of tasks that are in running mode when tasks are in running state the utilization factor remains constant but due to an increase in the number of tasks the utilization factor of the multi-core processor increases as discussed in the system and task model.

We are considering configurations for octa-core multiprocessor in which the migration of task depend on the utilization factor. There are a total of eight possible configurations in a processor to run different tasks and the coolest core configuration can have the highest priority to be selected first. Tasks that are in a ready state are placed in the ready task queue. The core configurations are defined against the range of u_i for energy-aware EDF scheduling algorithm is mentioned below:

- **Configuration 1**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} \geq (1 \rightarrow 9\%)$ One core is in running state and the rest of the seven cores are in sleep mode, the tasks can schedule either one core configuration {(1), (2), (3),(4),(5),(6),(7),(8)}

- **Configuration 2**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} \geq (9 \rightarrow 18\%)$ Therefore two cores are in running state and the rest of the six cores are remain in sleep mode, tasks can schedule on any of the two core configurations {(1,2),(3,4),(5,6),(7,8)}

- **Configuration 3**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} \geq (18.1 \rightarrow 27\%)$ Therefore three cores are in running state and the rest of the five cores are remain in sleep mode, tasks can schedule on any of the core configurations {(1,2,3,),(4,5,6,),(1,7,8)}

- **Configuration 4**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} \geq (27.1 \rightarrow 36\%)$ Therefore four cores are in running state and the rest of the four cores are remain in sleep mode, tasks can schedule on any of the configurations of core set {(1, 2, 3, 4), (5,6,7,8)}

- **Configuration 5**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} \geq (36.1\% \rightarrow 45)$ Therefore five cores are in running state and the rest of the three cores are remain in sleep mode, tasks can schedule on any of the core configurations {(1, 2, 3,4,5), (4,5,6,7,8) }

- **Configuration 6**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} \geq (45.1 \rightarrow 54\%)$ Therefore six cores are in running state and the rest of the two cores are remain in sleep mode, tasks can schedule on the configuration of core set {(1, 2, 3,4,5, 6) }

- **Configuration 7**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} \geq (54.1 \rightarrow 62.5\%)$ Therefore seven cores are running and the rest of the two cores have remained in sleep mode, tasks can schedule on the core configuration {(1, 2, 3,4, 5, 6,7)}

- **Configuration 8**

For $\forall\tau \rightarrow u_\tau = \sum_{k=1}^{n} \frac{ca}{pa} > 62.5\%$ Therefore all the eight cores are in running state, tasks can schedule on the core configuration {(1, 2, 3,4, 5, 6,7,8) }

Demand bound (d_{bi}) is applied with response time to ensure that cores are enough $\forall\tau$ to meet their deadlines (d_a). The function of (db) gives computation time for $\forall\tau$ to complete the operation within time interval $[t_a, t_b]$ as shown in Eq. (19).

$$d_{bi}\left[t_a, t_b\right] = c_i;\ \forall\left[r_i \geq t_a, d_a \leq t_b\right] \tag{19}$$

Eq. (19) represents the demand bound for n periodic task set $\tau = \{\tau_a, \tau_b, \tau_c \ldots\ldots \pi_n\}$ with a constrained deadline $(\mathrm{d_a} = \mathrm{p_a})$.

$$d_{bi}\left[t_a, t_b\right] = \sum\nolimits_{i=1}^{n} db_i\left[t_a, t_b\right]; \forall\left[r_i \geq t_a, d_a \leq t_b\right] \tag{20}$$

We are using EA-EDF to map the tasks on different cores while the configuration is defined according to the range of utilization factors. The counter used in the algorithm calculates how many times a configuration is selected due to which it is easy to avoid the repetitions of the same configuration again and again. Spatial gradients are also avoided due to the counter because when a core has not been selected the value of the Counter is zero and once a Configuration is selected its counter for the running duration is increases unless it reaches the threshold utilization and tasks start migrated to other configuration.

4 Experimental Technique

Section 4 elaborates the experimental technique using the STORM Simulator. Additional support is provided by considering the hardware architecture of Intel PXA-270 MPSoC with DPM capabilities. Core configurations are selected according to the task load and utilization factor e.g., we assume that the power model is the same for all the cores. In the test scenarios we have adopted the power model of the Intel PXA270 processor for each core because Intel PXA270 supports 7 Power-states (from 13 to 624 MHz) and 3 major C-states including active, idle and sleep. We have performed various experimental tests. In the test scenario, we applied energy-efficient EDF as a real-time task scheduler on each configuration of the core by considering the synthetic task set. We have evaluated the behavior of our proposed algorithm in which all the tasks run until the C_a WCET. For this, we generated a task set and put the values in an XML input file and the size of the task set is between 5–27 tasks by considering the period of each task is between [0.15, 30 ms] as mentioned. We executed the task set mentioned in Tab. 1 on the Intel PXA270 MPSoC processing platforms with 4 cores and 8 cores, respectively by considering the various states of running, idle and sleep as mentioned in Tab. 2. During the tests it is observed that the deep idle and sleep modes consume the energy with a difference. To verify the scalability and performance of the proposed algorithm, we have to change the number of available CPU cores within the range 4–8. Fig. 4 illustrates the complete experimental setup of the research design that is based on a comprehensive task migration-based EA-EDF algorithm that split an application into various no of tasks according to the task set parameters.

Table 1: 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

Table 2: 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

Table 3: Energy consumption comparison of different schedulers at 520 MHz

Utilization	No of task	Proposed	Energy	Energy	Energy	Energy	Active	Idle	System
		EA-EDF	GEDF	PDTM	Uniform	TBP	Power	Power	Bus
					RT-DPM		mW	mW	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

The features of the task set include the software architecture, start time and worst-case execution time, period. Priority, deadline, hardware architecture no of cores, proposed scheduling algorithm that is added into XML file. This XML File is then given to the simulator STORM where the scheduling policy is already implemented after the execution of the new scheduling-based task migration policy on Eclipse JAVA Neon 64bit using the storm jar file the simulator can produce various Power profiles, Energy-efficient results according to the load utilization factor and CPU load curves, CPU memory consumption, CPU energy consumption for Individual cores using both hardware & software architecture of the processor model for all type of hardware configurations already mentioned in an XML file as input to the simulator Scheduling technique for Multicore Intel PXA-270 MPSoC. 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, 14 ms] as indicated. We executed the task set mentioned in Tab. 1 on the Intel-PXA270 MPSoC processing platforms.

Fig. 4. Illustrates the proposed experimental model that assess the reliability using the mentioned experimental setup. The proposed scheduling technique calculates the energy and power and switches the tasks on the core based on their utilization factor. In our experiments all hardware parameters are used from the intel PXA270 processor various configurations are used that are depending on the utilization factor of the core.

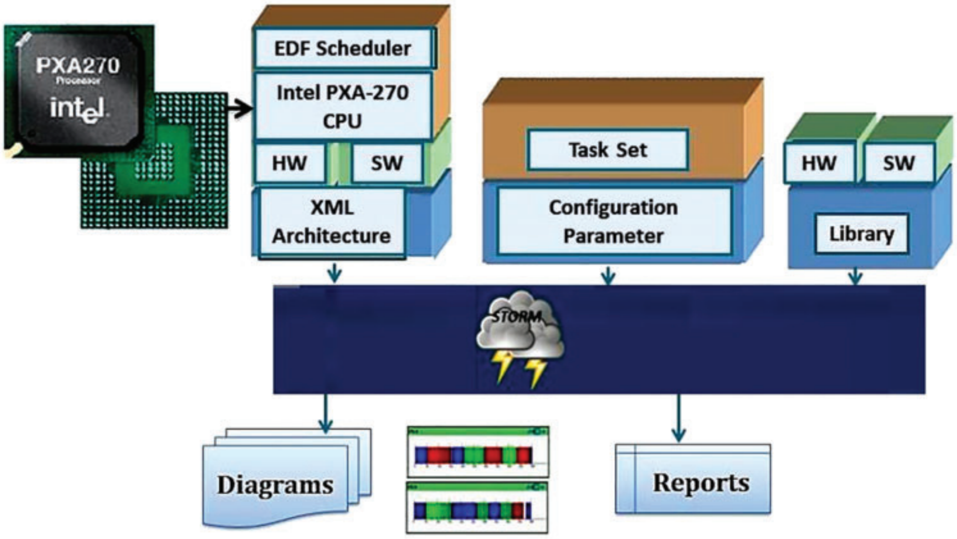

Figure 4: Proposed experimental model for energy-efficient scheduling

The utilization of the processor reaches near to threshold. The tasks start migrating to the core that is physically distant from the core with high power consumption due to high utilization of task and temperature for attaining better performance and avoiding delay in the execution process. Selection of core configuration is based on the stae of the core. Due to high load on CPU tasks are scheduled and migrated to the coolest core that is in sleep mode and is prior shifted to idle mode before the migration of the task.

Tab. 2 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.

5 Simulation Results & Analysis

The CPU energy consumption is measured concerning Intel PXA-270 MPSoC. The proposed Algorithm selects the task having the earliest scheduling deadline selected on priority. The technique works efficiently when a periodic task set is considered that's why the value of the context switches. Simulation shows that STORM Simulator provides performance criteria that are similar to the real platform values by considering the same experimental parameters such as deadlines and energy consumption. The comparison of energy consumption at various utilization factors u_i on 520MHz as shown in Fig. 5 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 4.8% of the overall energy at for $\forall\tau \rightarrow u_\tau < 62.5\%$. Tab. 3 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.

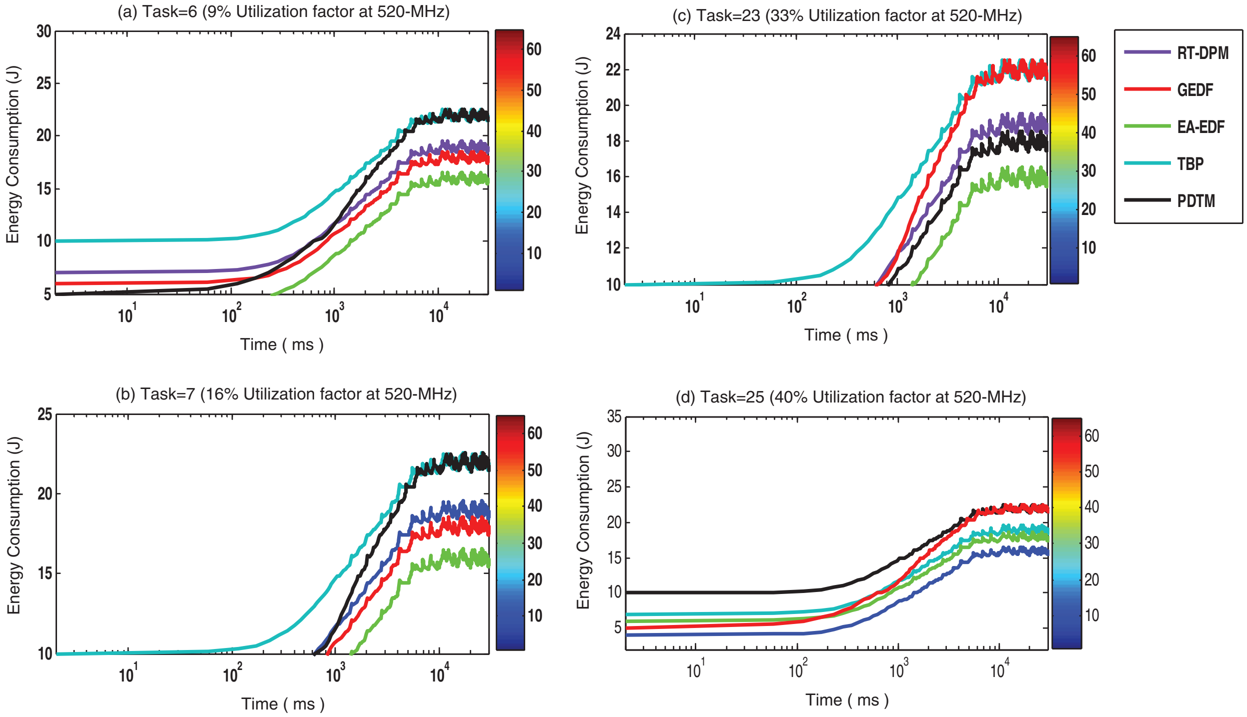

Figure 5: Comparison of energy consumption u_i at 520 MHz

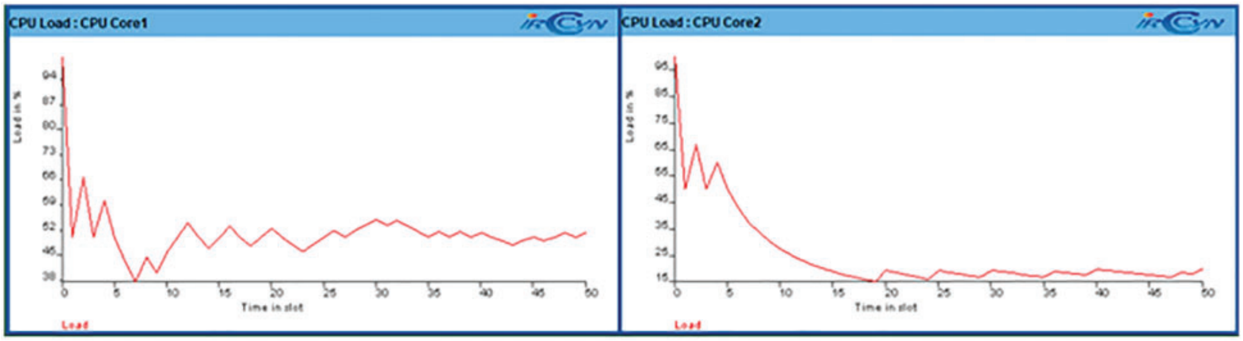

Figure 6: CPU load curve at CPU cores1 & cores2 under $u_i = 36\%$ utilization at 520 MHz

Fig. 6 represents the impact of CPU load on individual core over time due to increasing the number of periodic tasks with implicit deadlines are reducing the energy and power consumption using the proposed approach. It also illustrates that by increasing the task load on the CPU the power denisties are increasing rapidly but due to EA-EDF scheduler the tasks on core 1 switches to core 2 and dissipate less energy as compare to core1.

Tab. 4 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 4.8% & 4.3% more energy efficient results on 9%, 16%, 33% & 40% u_i as compared to GEDF, RT-DPM, TBP, PDTM on 520 MHz & 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 4: Energy consumption comparison of different schedulers at 624 MHz

Utilization factor	No task	of 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

6 Conclusions

This research paper targets energy-efficient multiprocessor scheduling problems over $\tau i \in \tau$ over 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 offline 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. The 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 both homogeneous and heterogeneous multiprocessor platforms apart from this the proposed technique gives us more efficient results by optimizing 4.3%–4.8% energy on utilization of 9%, 16%, 33% and 40% at 520 MHz & 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] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[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 33*, vol. 33, 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] 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, Country, pp. 616–623, 2011.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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, no. 4, pp. 1–20, 2013.

[26] 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.

[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] 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.

[29] 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.

[30] 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, 2018.