MDPI

Article

Optimization of energy consumption in battery-operated real-time systems

Abstract: This research proposes a model for work scheduling that minimizes energy usage by implementing a suitable scheduling algorithm for dynamic power management (DPM) and dynamic voltage scaling (DVS), and evaluated the effectiveness of power management approaches. The task model used is an implementation of the Least Slack Time (LST) algorithm with a preassigned priority as the tiebreaker. Our findings demonstrate that the use of DPM and DVS can significantly reduce the energy consumption of computing devices. Applying DVS leads to an 84.03% reduction in energy consumption, while using DPM reduces the total energy consumed by devices by 66.76%. Combining DPM and DVS yields similar energy savings to that of DVS alone (83.22%) but with the potential for greater energy savings due to external device involvement. While this combination can lead to longer execution times, the significant reduction in energy consumption makes it a worthwhile trade-off for energy-efficient computing. Our objective strategy is easily adaptable to different applications.

Keywords: Least Slack Time; Dynamic Voltage Scaling; Dynamic Power Management; Dynamic power; Static power; Device power

1. Introduction

When it comes to embedded systems, it is crucial to consider energy consumption, as energy-aware real-time systems are more reliable. Real-time energy systems are increasingly consuming electricity and optimising their consumption will result in significant savings and environmentally friendly benefits, as evidenced by recent statistics. The demand for data centre services has grown steadily over the past years, with a projected rate of growth of 4 % to 5 % annually, based on an International Energy Agency IEA report. A corresponding increase in energy consumption is expected as a result of this growth, with data centres projected to represent 3.2% of the world's electricity consumption by 2025 [1]. The optimised energy consumption is essential for the efficient and reliable functioning of battery-operated real-time systems, in order to ensure that they operate as long as possible. It is especially relevant for applications where the system is situated on a distant or hard to reach location and, in some cases, it may not be feasible or costly to replace batteries regularly. According to a study by the Department of Energy, wireless sensor networks are consuming as much as 40% energy for data transmission. The energy consumption of wireless remote network equipment and the extension of battery life may be substantially reduced through optimisation of protocols and techniques, as well as by enhancing performance in these devices [2]. By reducing the need to replace batteries and the energy costs associated with the charging of batteries, energy consumption can also lead to cost savings. Furthermore, energy saving can contribute to mitigating concerns regarding the environment associated with storage of batteries.

Some of the techniques that can be used for energy optimization include Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS). DPM is used to dynamically adjust the power consumption of a system based on its workload and is typically used in battery-operated real time systems to extend battery capacity. The idea behind DPM is to adjust the processer's voltage and frequency to match the workload as the processor's power consumption is directly proportional to its voltage and frequency. This is achieved by increasing the voltage and frequency of the processor when the workload is

Citation: . *Journal Not Specified* , , .

Received:
Accepted:
Published:

high and reducing them when the workload is low [3]. DVFS is to adjust the voltage and frequency of the processor dynamically based on the current workload. When the workload is high, the voltage and frequency are increased to provide the necessary processing power, and when the workload is low, the voltage and frequency are decreased to save energy. This approach enables the processor to operate at the minimum possible voltage and frequency while meeting performance requirements [4]. The combination of Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS) techniques is often used to optimize energy consumption in real-time systems [5]. This combination is capable of delivering a substantial reduction in energy consumption but maintaining the required level of performance. The DPM will adjust the system's power consumption on the basis of workload, and the DVFS is adapting processor voltage and frequency so as to match this workload. The system can achieve even more significant energy savings by using this combination. For example, DPM can be used to put the system into a low power state during periods of low workload, and DVFS can be used to reduce the voltage and frequency of the processor to further reduce the power consumption. In a period of high workload, DPM can bring the system from low power state and DVFS increases processor voltage and frequency in order to provide sufficient processing power. The combination of DPM and DVFS has been widely adopted in a wide range of devices, including cell phones, computers as well as servers. It's shown to deliver substantial power savings while maintaining the required level of performance [5].
This work focuses on three case studies that uses DPM, DVS and the integration of DVS and DPM along with Least Slack Time (LST) scheduling algorithm to optimize the energy consumption in real time embedded systems that are battery operated.

2. Literature Review

Some of the existing works in optimizing energy consumption are reviewed in this section.
In [6], Zhou Tang et al. proposed an energy-efficient DVFS-enabled workflow task scheduling algorithm (DEWTS) that is based on Dynamic Voltage and Frequency Scaling (DVFS) to reduce the total power consumption. DEWTS is used in processors that are DVFS enabled and applies to most data centres' scheduling systems. The proposed approach is shown to have reduced power consumption by up to 46.5% compared to other state-of-the-art models.
In [7], Tajana Simunic et al. proposed a system combining DVS and DPM techniques, thus enabling more power savings. DVS algorithm was proposed based on change point detection that is used to detect the change in arrival or decoding rates and the frequency setting policy, which uses the change point detection to set the frequency and voltage of the processor. This approach was tested on MPEG video and MP3 audio algorithms on a portable " SmartBadge " device. Then, this approach is combined with the DPM algorithm and shows a factor of three savings in energy.
In [8], Ijaz Ali et al. proposed a dynamic power aware scheduling scheme for mixed criticality tasks on unicore processors. To reduce the overall energy, energy level in high-criticality mode was reduced first to tackle the difficulty in minimizing power trade-off in Hi-criticality mode. When there is an idle period between high critical job execution, the processor will be switched to low critical mode. In terms of energy reduction in mixed criticality, the proposed scheme is shown to be effective and superior to static methods.
In [9], Ridha Mehalaine et al. proposed a flexible scheduling scheme to minimize energy consumption using Dynamic Voltage Scaling (DVS) technique on a multiprocessor system with uncertain data. The proposed approach also reuses the time savings that show the difference between the worst-case execution time (WCET) and the actual execution time (AET). Simulations have been performed using MATLAB to evaluate the model and confirm its reliability for reducing energy consumption in period independent tasks.
In [10], Sheikh et al. proposed some changes to the existing models by incorporating the change in execution time and energy consumption brought about by the number of cache

partitions assigned to a task for a three-dimensional energy optimization problem that includes minimizing the core-, cache- and system level energy consumption. According to the results, optimising cache energy plays a significant role in reducing overall energy consumption.

Some of the limitations of these existing works are that these approaches are specified to a particular device or processor or tasks which limits their applicability to other scenarios or applications. The complexity and the lack of readability limits the practicality of these approaches. Also, there a lot of approaches and methods involved in energy optimization which makes it hard to choose a suitable approach that suits the application.

The contributions of the proposed work include

1. This study aims to develop a mathematical model to explore the effectiveness of DVS & DPM in optimizing battery usage for more efficient and sustainable battery-powered embedded systems. 2. Defining an optimal model that optimizes energy consumption in scheduling tasks. 3. Defining a Least Slack Time scheduling algorithm that can be implemented with Dynamic Power Management (DPM) and Dynamic Voltage Scaling (DVS) techniques and the integration of DPM and DVS. 4. Evaluate the performance of these techniques through simulations using python. 5. Generalising the model so that it can be modified for any application.

3. Task Definition

The tasks are periodic in nature, in which the deadline line of the task is equal to its period and the first instance of the periodic task is released at time 0. The tasks are defined by task name, deadline, Worst Case Execution Time (WCET), Actual Execution Time (AET) which is less than WCET, priority assigned to it.

3.1. Task scheduling algorithm

A task scheduling algorithm based on the Least Slack Time (LST) policy and task priority as the tiebreaker. It starts by setting the current time [11]. The algorithm updates the slack time for each task, which is defined as the amount of time a task has remaining before its deadline, sorts the tasks based on the slack time and priority, selects the task with the least slack time as the next task to be scheduled, adds the task along with its start and end times to the new list, updates the current time with the end time of the scheduled task. The process continues until all tasks have been scheduled. If there are tasks that must be executed in a particular order, then the priority number assigned to the task will take care of the scheduling. For simplicity purposes this priority case is not considered.

Algorithm 1 Least Slack Time

```
procedure TASK_SCHEDULING(taskslist)
    current time = 0                                   ▷ setting current time
    schedule tasks = [ ]          ▷ initialising an empty list for scheduled tasks
    while taskslist not empty do
        slack time = deadline − (current time + WCET)  ▷ calculating slack time for all the tasks present in the list
        Sort the tasks list in ascending based on slack time
        next task = tasklist[0]▷ assigning the task with least slack time as next task
        schedule tasks.append(next task)
        current time+ = AET[nexttask]          ▷ updating current time with actual execution time
        taskslist.drop(next task)
```

4. Methodology

4.1. Case Study 1: Dynamic Voltage Scaling (DVS) for Energy Optimization

The power of a CMOS circuit can be divided into static power, which is independent of the frequency and dynamic power, which is affected by frequency and voltage change.

4.1.1. DVS energy model

The energy model for Dynamic Volatge Scaling aiming to reduce energy consumption is defined here.

- Dynamic power
  It is calculated by $P_{dyn} = aC_LV^2f$ [11] where a is the switching probability, C_L is the load capacitance, V is the voltage, and f is the frequency. In the processor, when there is a frequency, scaling and C_L are not affected. When the frequency is scaled by a factor of s, as an affect, voltage is scaled by s, the current is scaled by a factor of s^2 and the dynamic power is scaled by s^3 which is given by, $P_{dyn}(s) = (1)^{-3} * P_{dyn}(1)$, where P_{dyn} (1) is the dynamic power when there is no scaling factor applied to the system (s=1).

- Static power
  Static power, P_{static}, is defined as the sum of leakage power and power consumption of components which are not scalable. While leakage power changes linearly with the voltage, power consumed by non-scalable components are not affected by the voltage. In this analysis, static power is considered to be constant, and no external device is considered scalable for simplicity of the model.

- Optimal Scaling Factor
  The amount of energy used by the CPU changes depending on the frequency and voltage. Reducing the voltage doesn't always lead to lower energy consumption because it also increases the time it takes to complete a task. This results in more energy being used by the CPU while it's active, as well as more energy being used when the device is in a standby state. The maximum scaling factor for a DVS system that results in minimum energy consumption without considering any deadlines is determined in the rest of this section. This is referred to as the optimal scaling factor. For a task, consider an arbitrary value for WCET and AET, where AET
  le WCET. when the frequency is scaled by a factor of s, the execution time is also extended by the same factor.
  The static CPU energy consumption is given by

  $$E_{static}(s) = P_{static}(s) * (s * AET) = s * P_{static}(1) * AET = s * E_{static}(1)$$

  The dynamic CPU energy consumption is given by

  $$E_{dyn}(s) = P_{dyn}(s) * (s * AET) = s^{-2} * P_{dyn}(1) * AET = s^{-2} * E_{dyn}(1)$$

  Total energy consumption which is the sum of static and dynamic power is given by

  $$E_{cpu}(s) = E_{dyn}(s) + E_{static}(s) = (s - 2 * P_{dyn}(1) + s * P_{static}) * AET$$

  Consider $Q_{dyn} = s^{-2} * P_{dyn}(1), Q_{static} = s * P_{static}, and\ Q_{cpu} = Q_{dyn} + Q_{static}.$

  Then $E_{cpu} = Q_{cpu} * AET$. The optimal scaling factor is achieved by minimizing energy consumption, which involves minimizing Q_{cpu}. Which results in
  $s_{opt} = (2 * P_{dyn}(1)/P_{static})^{1/3}$ [11].

For the above plot, dynamic power is considered to be 125 watts and static power is considered to be 75 watts.

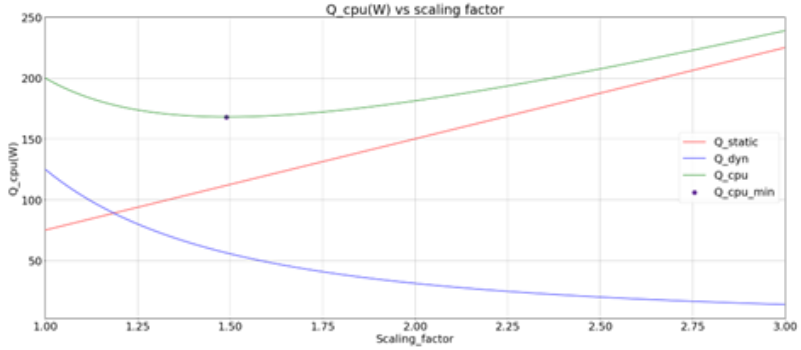

Figure 1. $Q_{CPU}(W)$ vs Scaling factor

The dynamic power consumption, $P_{dyn}(1)$, is related to the amount of switching activity, a, which can vary from task to task. Therefore, the optimal scaling factor is specific to each task. If all the tasks are known, then we must consider static processor utilization. Which refers to the amount of processor time occupied by the task when executed periodically. It is calculated as the ratio of the total execution time of the task over the period of the task [11]. Static processor utilization for n periodic tasks with P as period is given by

$\mu = \Sigma_{k=1}^{n} WCET^k / p^{[k]}$

The actual scaling factor is considered as the minimum of static power utilization and optimal scaling factor.

$s_{(act)} = min(\mu, s_{(opt)})$

Tasks	Deadline	WCET	Priority
Task 1	100	40	1
Task 2	50	45	2
Task 3	100	30	3
Task 4	70	20	4

Table 1. Task specification.

Tasks	Deadline	WCET	Priority
Task 2	50	45	2
Task 4	70	20	4
Task 1	100	40	1
Task 3	110	30	3

Table 2. Task execution order based on Least Slack Time scheduling algorithm

4.1.2. Results

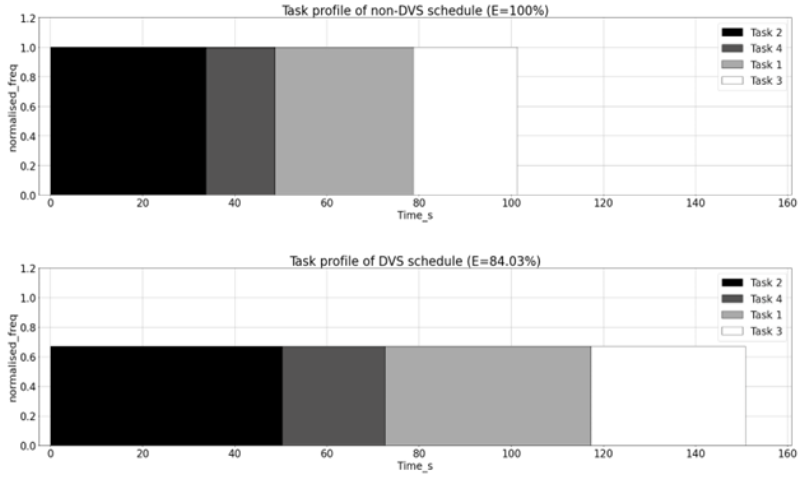

Figure 2. DVS Task profiles for the task set in Table 2

The frequency values are expressed as a proportion of the highest frequency and the energy values are expressed as a proportion of the energy used in the non-DVS scheduling profile. From Figure2, applying DVS increases the time it takes to complete tasks, as the processor runs at a lower frequency. However, this leads to a reduction in overall energy consumption by 84.03%.

4.2. Case Study 2: Dynamic Power Management for energy consumption optimization

If there are a lot of external devices or sensors connected to the system, then DPM is implemented to reduce the energy consumption by switching the states of inactive devices.

4.2.1. Device energy model

It is assumed that the devices have three modes of operation: active, standby (idle), and off. Let D_j be the number of devices required by the task j. The device consumes some power depending on the state of the device; P_j^{act} for active power consumption, P_j^{idle} for standby (idle) power consumption and P_j^{off} =0 for off state. Transition energy consumption is not considered in this model to reduce the complexity and the device is assumed to remain in the same state until the task is executed.

- Active energy
  Assume that the device (D_j is in active state and consumes an active energy E_j^{act}, which is constant but in reality, it would not be constant. Active energy is given by $E_j^{act} = P_j^{act} * AET_j$ [11]

- Idle energy
  Assume that the device D_j is in idle state and consumes an active energy E_j^{idle} which is constant. Idle energy is given by

  $$E_j^{idle} = P_j^{idle} * AET_j \text{ [11]}$$

  If the frequency of the processor is scaled by a factor s, then the execution time is also scaled by the same factor which also increases the device energy consumption.

Tasks	Device 1	Device 2	Device 3
Task 1	Active	Active	Active
Task 2	Idle	Active	Idle
Task 3	Off	Idle	Idle
Task 4	Active	Active	Off

Table 3. Task specification:deadline, WCET and priority can be considered from Table 1

Tasks	Device 1	Device 2	Device 3
Task 2	Idle	Active	Idle
Task 4	Active	Active	Off
Task 1	Active	Active	Active
Task 3	Off	Idle	Idle

Table 4. Task execution order based on Least Slack Time scheduling algorithm

4.2.2. Results

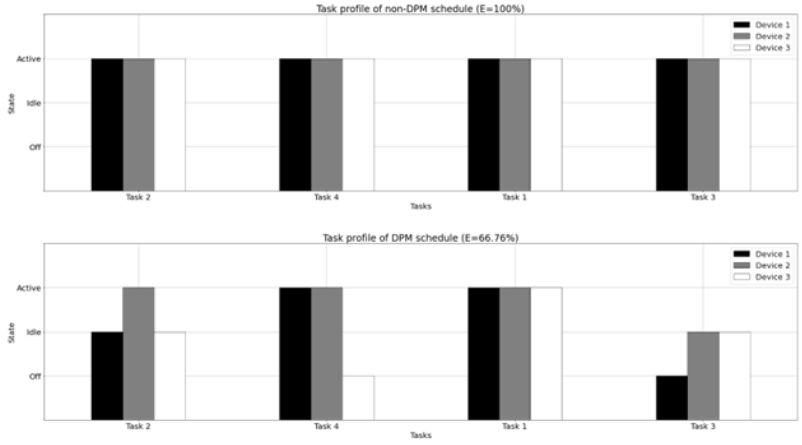

Figure 3. DPM Task profiles for the task set in Table 4

The energy values are expressed as a proportion of the energy used in the non-DPS scheduling profile. The total energy consumed by the devices is reduced to 66.76%. In this example, the processor is considered to be running at max frequency as DVS is not applied, as an effect the execution time is not scaled.

4.3. Case Study 3: Integration of DPM and DVS to optimize energy consumption

For the combination of both DVS and DPM, DPM is implemented first and then DVS is implemented along with the device power. DPM implementation is similar to the case study2.

4.3.1. Modification of DVS model

Consider a system with CPU and a set of devices/sensors. The total energy consumption now includes the energy consumed by the device ($E_{dev}(s)$) along with energy consumed by CPU ($E_{CPU}(s)$. As different tasks trigger different sets of devices, the optimal scaling factor for each task is different. Also, the device is assumed to stay active through out the execution of a task, so it can be added to static power. The total energy of the system for a task k is given by

$$E(s) = (P_{dyn}(s) + P_{static}) * s * AET + P_{dev}^{[k]} * s * AET$$

Tasks	Deadline	WCET	Priority	Device 1	Device 2	Device 3
Task 1	100	40	1	Active	Active	Active
Task 2	50	45	2	Idle	Active	Idle
Task 3	100	30	3	Off	Idle	Idle
Task 4	70	20	4	Active	Active	Off

Table 6. Task execution order based on Least Slack Time scheduling algorithm

$$E(s) = (s^{(-2)} * P_{dyn}(1) + s * (p_{static} + P_{dv}^{[k]})) * AET$$

Similar to DVS in case study1,

The optimal scaling factor is achieved by minimizing energy consumption, which involves minimizing Q_{cpu}. Which results in

$$s_{opt} = [2 * P_{dyn}(1)/(P_{static} + P_{dev}^{[k]})]^{(1/3)}$$

Similar to DVS, static processor utilization, μ is also considered

$$\mu = \Sigma_{(k=1)}^{n}(WCET^{[k]})/p[k]$$

The actual scaling factor is considered as the minimum of static power utilization and optimal scaling factor.

$$s_{(act)} = min(\mu, s_{(opt)})$$

If $P_{static} + P_{dev}^{[k]} 2 * P_{dyn}(1)$, the optimal scaling factor is $S_{opt}1$ which implies that DVS should not be applied.

Tasks	Deadline	WCET	Priority	Device 1	Device 2	Device 3
Task 1	100	40	1	Active	Active	Active
Task 2	50	45	2	Idle	Active	Idle
Task 3	100	30	3	Off	Idle	Idle
Task 4	70	20	4	Active	Active	Off

Table 5. Task specification:deadline, WCET and priority is considered from Table 1 and state of devices is considered from Table 3

4.3.2. Results

Consider the order of tasks from Table 6 and DPM is applied for each task as it is task dependent.

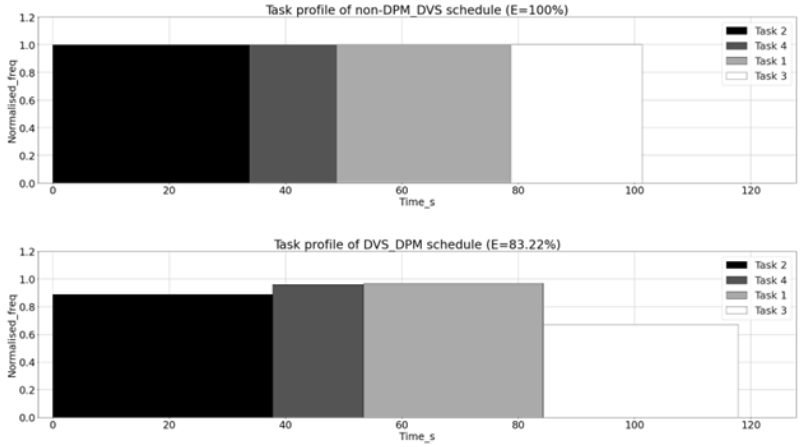

Figure 4. DPM Task profiles for the task set in Table 6

As shown in the above graph the percentage energy saving is similar to that of only DVS, 83.22%. But as this system includes external devices, in absolute numbers this method will have more energy saving than that of DVS(mention section number). The reason for less energy saving is because the external devices are in their respective states throughout the task execution time.

5. Challenges

- As most of the systems are not programmed to operate in full voltage region. Usage of DVS is very limited in the lower regions of the voltage spectrum and there are very few applications which can take advantage of scaling at low voltage levels.
- In digital circuits, Propagation delay is inversely proportional to supply voltage. Most mathematical models do not consider leakage current.
- DVS can cause stability issues such as system crashes and hardware failures if not implemented correctly.
- DVS can affect the temperature of the system, and the thermal management system needs to be designed appropriately to handle the temperature changes.
- Due to the policy limitations in DPM, sometimes the estimation of idling periods might not be accurate which results in under or over estimation of the idle periods.
- A potential loss in energy saving might occur if there is under estimation of idling periods. Similarly, an overestimation can result in both performance delay and energy loss.
- The appropriate combination of DVS and DPM techniques presents various issues because aggressive DVS schemes result in short device idle times, limiting the effectiveness of DPM solutions; similarly, solutions with DPM as the focus may result in excessive CPU power usage. As a result, at the system level, there is a trade-off spectrum between DVS and DPM

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

6. Conclusion

This research aims to propose an appropriate model for work scheduling that minimises energy usage, create a suitable scheduling algorithm for DPM and DVS, and evaluate the effectiveness of power management approaches. This objective strategy is easily adaptable to different needs. The findings of our study demonstrate that the use of dynamic voltage scaling (DVS) and dynamic power management (DPM) can significantly reduce the energy consumption of computing devices. Applying DVS leads to an 84.03% reduction

in overall energy consumption, although it increases the time taken to complete tasks due to the lower frequency of the processor. On the other hand, using DPM reduces the total energy consumed by devices by 66.76%, without impacting the execution time as the processor is considered to be running at its maximum frequency. Furthermore, the combination of DVS and DPM results in similar energy savings to that of DVS alone (83.22%), but with the potential for even greater energy savings due to the involvement of external devices. Although the use of combination of DVS and DPM can lead to longer execution times, the significant reduction in energy consumption makes it a worthwhile trade-off for energy-efficient computing. Future work will involve implementing sophisticated task-splitting scheduling algorithms, taking RAM and thermals into account, and validating them in real-world scenarios. It is possible to investigate enhanced device models that take transition and break-even times into account. It is also possible to implement other scheduling algorithms, such as partial task execution, and investigate various pairings of power management strategies, such as hybrid DPM, fine-grained power management, and application-specific power management. For all next work, real-world validation and correlation must be done.

7. Future Work

- A more advanced scheduling algorithm can be implemented, which can split the task into multiple subtasks which can increase the efficiency.
- A more advanced Device model can be implemented, which takes in the transition times and their respective powers into consideration.
- The device transition can also take break even time as a parameter. Which can help and avoid unnecessary state transitions.
- A different type of scheduling algorithm can be implemented where the devices are active for a fraction of the task execution time rather than staying active throughout the task
- A different type of combination can be implemented for more energy saving. Some of which are Hybrid DPM, Fine-Grained Power Management, Application-Specific Power Management.
- The work shown here is only through a simulation. In the future, real world validation and correlation should be done.

References

1. https://www.iea.org/reports/data-centres-and-data-transmission-networks.
2. https://www.shipleyenergy.com/resources/efficiency/the-importance-of-unplugging-unused-appliances/.
3. Nogues, E.; Pelcat, M.; Menard, D.; Mercat, A. Energy efficient scheduling of real time signal processing applications through combined DVFS and DPM. In Proceedings of the 2016 24th Euromicro international conference on parallel, distributed, and network-based processing (PDP). IEEE, 2016, pp. 622–626.
4. Benini, L.; Bogliolo, A.; De Micheli, G. A survey of design techniques for system-level dynamic power management. *IEEE transactions on very large scale integration (VLSI) systems* **2000**, *8*, 299–316.
5. Chen, G.; Huang, K.; Knoll, A. Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination. *ACM Transactions on Embedded Computing Systems (TECS)* **2014**, *13*, 1–21.
6. Tang, Z.; Qi, L.; Cheng, Z.; Li, K.; Khan, S.U.; Li, K. An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment. *Journal of Grid Computing* **2016**, *14*, 55–74.
7. Simunic, T.; Benini, L.; Acquaviva, A.; Glynn, P.; De Micheli, G. Dynamic voltage scaling and power management for portable systems. In Proceedings of the Proceedings of the 38th annual Design Automation Conference, 2001, pp. 524–529.
8. Ali, I.; Jo, Y.I.; Lee, S.; Lee, W.Y.; Kim, K.H. Reducing dynamic power consumption in mixed-critical real-time systems. *Applied Sciences* **2020**, *10*, 7256.

9. Mehalaine, R.; Boutekkouk, F. Energy consumption reduction in real time multiprocessor embedded systems with uncertain data. In Proceedings of the Artificial Intelligence and Bioinspired Computational Methods: Proceedings of the 9th Computer Science On-line Conference 2020, Vol. 2 9. Springer, 2020, pp. 46–55.
10. Sheikh, S.Z.; Pasha, M.A. An improved model for system-level energy minimization on real-time systems. In Proceedings of the 2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2019, pp. 276–282.
11. Zhuo, J.; Chakrabarti, C. Energy-efficient dynamic task scheduling algorithms for DVS systems. *ACM Transactions on Embedded Computing Systems (TECS)* **2008**, *7*, 1–25.