ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Improved Real-Time Scheduling algorithm for mixed task Set with Constraint of Harvesting Energy

[1] Mona Kumari, [2] Ajitesh Kumar
[1][2] GLA University Mathura

***Abstract* - In mobile system devices, every system must be having own resources for power and cannot depend on the power from the outside, apart from feasibly scheduled the task set, managing the power is the main concern for all type of real-time systems. The consumption of the energy depends upon the power management of the mobile devices. In this paper, we propose an extended and improved real-time scheduling algorithm for a mixed task set with the constraint of harvesting energy which has flexible speed assignment for a set of periodic and aperiodic tasks and ensures the feasibility schedule within their deadlines. This proposed article having some experimental results that show this algorithm is best for performance in terms of available energy after task execution, an average ratio of the task set at lower periodic load and deadline miss ratio.**

Index Terms — RTS, Embedded System, Harvesting energy, DVS, Mix Task Set.

I. INTRODUCTION

In every real time system all task should be completed within their deadlines and produce the correct result for an example anti-locking break system in a vehicle is a best example of hard real time system in this time computing system the real data is not produce expected result with in their deadlines it may cases system fail. Whatever real time system is there may be hard soft or firm is demand to achieve energy neutral operation if its execution requirement can be supported forever despite energy limitations [10].

In today's environment every devices executing on that platform which have their own battery power and needs real data with in a time period and also do not depend upon the power of outside because of the nature of mobility and most of the devices remains beyond the recharge point due to mobility, for example video streaming applications require light weight device that can be movable across the world [11].

So in this type of system those are always in mobility and not to be recharge every time because of unavailability of recharge point it require a strong power management system to enlarge the battery backup time period. However, in some application like wireless sensor node deployed in remote areas for environment surveillance, the replacement and recharging battery is not possible due to remote areas so such type of system also require harvesting energy for recharge their battery power and power management for battery to increase their life time. With the enhancement in the battery power the harvesting feature techniques is incorporated in battery [5]. Harvesting technique is very useful where the battery replacement or recharging is costlier. The harvesting energy is the process of producing electrical energy by using renewable energy resources available in environment. There is a lot of resources available in nature like solar energy, kinetic energy thermal energy etc. Therefore, the energy generated by such resources is sufficient for any battery operated system and there is no need of battery replacement, only requires best power management policies.

Characteristics of Harvesting Real Time Application:

The systems that have the ability to execute the task at lowest standby current to maximize the battery power storage, and in this system it also has the switch on-off ability instantly along with the analog capability for sensor communication and measurement. This system also has to operate on lowest voltage range to maximize the harvest power backup of the battery.

Harvesting Energy real time System Limitations:

1. Renewable energy resource is not available always it depends upon the nature.
2. Energy intensity varying and depends upon the nature of environment like in day time the intensity it too high and in night it may be zero in case of solar energy.
3. Energy storage must be limited and depends upon the battery capacity.

Considering all above constraint, to maximize the possible number of mixed task set to be schedule so that it requires efficient power management.

In this paper we are focusing on scheduling mix periodic task with deadline constraint on single processor with variable speed assignment by using renewable energy resources with limited capacity such as battery or capacitor.

This paper having a proposed approach for those system that are always in mobility and having the problem of energy consumption. By using harvesting constraint this paper successfully implements the algorithm where the task executes with low level energy consumption and recharge their battery whenever the battery goes down form their threshold value.

This research article organized as, in section II we present related work with motivational example in section III we show that some system model and assumption and in section IV as main section of this paper is present the proposed algorithm with an example and in next section V simulation results are discussed and finally in next section we summarize the articles with some references.

II. RELETED WORK AND MOTIVATION

Now days the consumption of energy plays a vital role for mobile devices. There is a lot work done in this area and most of the focus on either minimizing the energy consumption or somehow maximizing the system performance such as the target achieved with the energy constraints. In such cases, the ability of the recharge energy storage unit is always disregarded with time.

Rechargeable system has gained a little attention for scheduling real-time tasks on a single processor. The question is arrived how much amounts of energy to find a schedule which is able to execute all the tasks within their deadline i.e. without running out of energy. Moser et al. [12] focused on scheduling tasks with deadlines for mix task set, that run on a single processor system that is powered by a rechargeable battery. The source energy is assumed to be predictable but time-varying. This paper proposed lazy scheduling algorithm (LSA) and experimental result shows it to be optimal in terms of deadline miss ratio. LSA is a variation of the famous earliest deadline first scheduling algorithm, the system starts executing a task only if the task is ready and has the earliest deadline among all ready tasks and the system is able to keep on running until the deadline of the task with maximum power.

In this article, the consumption energy of the computing system is characterized by some highest value which implies that for periodic task, its total energy consumption is directly connected to its execution time through the constant energy of the processing system.

The main drawback of this work is that the LSA algorithm executes tasks at full power and therefore, future tasks will violate deadlines because of limited energy. Moreover, in practice, the total energy which can be consumed by a task is not necessarily proportional to its execution time.

In article [7] presented a scheduling algorithm, EDF (Earlier Deadline with energy guarantee constraint) in which tasks are schedule according to an on-line algorithm that ignores the future energy generator and the arrival time of tasks. It knows only the energy consumption of tasks that are released on the processor that takes into consideration the limits of both time and energy. EDFeg relies on two basic concepts: slack time and slack energy. The main idea behind EDFeg is to run tasks according to the EDF.

Before schedule a task to be execute, they must ensure that the energy storage is sufficient to execute all future coming tasks. When such type of condition is not verified, the processor has to idle that time so that the battery recharges as much as possible and meet to threshold value. The drawback of this approach is that task are executed with a fixed speed which will some time cause energy overflow of the battery, on the other way some times insufficient energy to be shown by battery to execute a particular task set.

In the paper Allavena et al. [2] describe an off-line scheduling algorithm that uses voltage and frequency selection variables (DVFS) for a frame based real time task. While they allow to reduce power consumption by slowing down task set execution under their deadline constraints, this approach shows on the unrealistic assumption that both the instantaneous consumption power and production power are constant.

In article [10], Liu et al. gives an energy aware dynamic voltage and frequency selection algorithm, called EA-DVFS, for periodic tasks. They show in EA-DVFS is to efficiently use the slack time availability and further

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

reduce the ratio of miss rate of deadline. This algorithm, whether or not the system slows down the task execution for energy saving depends on the available energy. If the system has sufficient power, the task is executed at its full speed of execution, otherwise, this execute at a lower speed level. Unfortunately, this algorithm has limited impact since, in most real time applications, the energy storage has a non-constant recharging rate and every task is characterized by its own power consumption which can vary along time limit.

Motivational Example:
We have three mixed tasks $T_{S1}(0, 3, 12)$, $T_{S2}(1, 3, 10)$, $T_{S3}(0, 2, 8)$ that are required to execute with their deadline constrain. And the system that we consider to schedule this set of mixed tasks is managed by a harvesting energy resources like solar energy. The attribute of tasks T_{S1}, T_{S2} and T_{S3} are having release time r(t), worst case execution time and deadline.

So we are going to be assume that the harvesting energy with a constant rate H(r) during in day and which is approximately equal to 2J/sec, and the processor that we consider for scheduling the set of tasks can allow three different frequency or say that speed levels S_{low}, S_i, S_{max}. At S_{low} task is consumed 1J of energy per second, at S_i task may be consume 3.5J/sec and at S_{max} task consume 7.5J/sec. Suppose that the total capacity of energy storage in battery (BC) = 28J, and energy available in store at a time

At starting time t=0 $E_C(t) = 22J$.

At t = 0, there are only two task T_1 and T_3 is ready to execute and as the deadline of T_1 is earlier than T_3 that are given, so according to scheduling algorithm EDF, priority of T_1 is higher than T_3. So, at t = 0 T_1 start its execution. But before that, we must be check that is there sufficient energy available in battery to execute the task T_1 at its maximum speed until its total execution.

$$E_{T1}(S_{max}) = 7.5*2 = 15J < 22J = E_C(t=0)$$

As the requirement of T_1 to complete its execution at maximum speed is less than the available energy, so that T_1 execute at S_{max} until its completion.
At t=1 another task T_2 is released and as the deadline of T_2 is earlier than T_1, therefore T_2 preempt T_1. Execute T_3,
So that before start of execution firstly check that energy availability.

$$EC(t=1) = 22 - 7.5 + 1 = 15.5J$$

So that the need of energy to execute T2 to complete its execution at maximum speed Smax is 15.5J which is less than the available energy at store, therefore T2 execute at Smax until its completion, and complete its execution at t = 3. Therefore, at t = 3

$$EC(t) = 22 - 16 + 2 = 8J$$

Here, T1 resume its execution, and we find the remaining energy require to complete its execution without failing its deadline at Smax is 8J which is less than the available energy, therefore, T1 execute at Smax until its completion, and completed at t = 4. So that

at t = 4 EC(t) = 9 - 7 + 1 = 3J

now T3 starts its execution, and as the energy requirement of T3 at Smax is 24J, which is more than the available energy. Therefore, T3 execute at Slow. And at Slow, T2 takes 12s to complete its execution, which is more than its deadline time.

III. HARVESTING ENERGY SYSTEM MODEL AND ASSUMPTION

For a real time, uniprocessor system that contains a set of independent pre-emptible mix task set T1, T2, T3 ......Tn. for each task Ti has a different attribute:

Ai = arrival time of task Ti

worst case execution time of task Ti at defined speed Si
Ei at defined speed = Energy required Deadline of mixed task set is given

Here we use the real time dynamic priority scheduling algorithm that is EDF so according to EDF the task is assigned the priority and schedule the set of independent mix task set.

Here We use DVS process that are capable to execute the task set in variable speed levels like Vmin, Vmid and Vmax with the corresponding speed level Smin, Smid and

Smax. Here any task set run at any speed level between Smax and Smin.

Power of Energy Consumption $P = C * S^3$ -------- (1)

The response time of task Ti at speed Si is the sum of its own execution time interval and the execution time of its higher priority tasks which one is preempted.

Source of Harvesting Energy Generation:
The harvesting energy resource is depending upon behavior of environment like solar and wind etc, such type of the system highly varying according to time.

Energy generated by resource = time interval* rate of generation of energy ------(2)

Real time system that are considered the rate of harvesting energy generation. That is approximately equal to 1.5J/sec and may be it should be constant in day time and rate will be zero in night time due to solar system.

Storage of Harvesting Energy:
We assume that we are having maximum storage energy capacity C and if no task is executed it may be reaches to maximum capacity and shows overflow

Consumed energy is always lie in between zero energy and capacity of battery .... (3)
For running the task, battery power at a time and the respective energy Ed in between time interval is consumed from the battery where to be stored. That we have following assumption that energy stored at t time will be vary time to time depends upon the energy consumptions.

Energy Consumption:
Energy consumption means battery power may be consumed and depend upon the processor speed to execute the particular task set. The Energy consumption in time interval is depend upon the power consumption at the same time interval. Energy consumed maximum when process run at maximum level of processor speed and when it works low speed it consume lesser energy.[14]

IV. PROPOSED METHODOLOGY WITH EXAMPLE

In harvested energy system mix task set will be executed at minimum possible level speed that is just enough to meet its deadline and saved energy should be used to next higher level if it requires to any task to meet their deadline or to be avoid to energy overflow due to recharging of the battery.

Every mix task is executed at minimum level of energy even the battery has full power and compute the task at full speed of the processor.Whenever any mix task set should be arrived. There are two different situations with respect of energy.

First Situation: When mix task requirement in maximum speed is lesser than available battery energy C, then we start to calculate slack time period for mix task and if slack is greater than zero, means slack is available then reserve a time slot from next start time of a task to deadline, and energy equals to the energy required by Ti to complete its execution at maximum speed.

Now we calculate the lowest feasible speed for a task set For that the available energy in battery will be added with the harvested energy and reduce that consumption energy which are less than the available energy of the battery.
And the execution must be completed before the deadline of the task set.

After the execute of task at assigned lower speed up to completion of task and slack time whichever is earlier.
So If task execution is not finished before the latest start time of a task, then execute the remaining portion of task at full speed in work on the reserve slot with reserve energy.

Second Situation: If energy requirement of task at maximum speed is greater than total available energy EC(t), then again we will calculate lowest feasible speed for a task execute the task at assigned speed until its completion and arrival of some higher priority task whichever is earlier.

Improved Harvesting real Time Scheduling Algorithm(IHA-RTS) This improved algorithm is helping out the process of battery recharge by using harvesting energy system so that the battery will recharge after its threshold value reaching.

Step 1: Ready Queue and reserve queue for harvesting energy should be initialized by zero
Step 2: Schedule the task on the basis of EDF algorithm
Step 3: If the ready queue is empty then process the task

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Step 4: And if available energy in the battery is greater than zero and the task execution with highest speed is less or equal to the available energy
Step 5: then calculate the slack availability
Step 6: If available slack is zero then this task should be entering in the ready queue and run the task in higher speed of the processor. And if the slack is available
Step 7: Execute the task as per the scheduling algorithm
Step 8: Otherwise, if any other task arrived at a time which are the less than running task
Step 9: and their dead line is also less than the running task
Step 10: Then we update the ready queue and coming task preempt the running task
Step 11: Now we update the energy available with battery by available subtracted by consumed energy
Step 12: If the energy required by the coming task to execute is lesser or equal to available
Step 13: then switch to step 5
Step 14: otherwise step 22
Step 15: Now we continue to execute the task in assigned lower speed
Step 16: And if the task is complete before the deadline then remove the task set from ready queue and reserve queue
Step 17: Now saved energy to be freeze and time slot for next coming task
Step 18: Now the available battery power is added the reserve and available
Step 19: otherwise remaining portion of task set should be run at higher speed level of the processor
Step 20: And now consumed energy should be less from the available energy
Step 21: now available energy in the battery is C
Step 22: Now check the available energy is not zero and energy consumed at higher speed level is greater than available energy then
Step 23: Compute the lower possible speed to execute the task by using AFSA (Algorithm for speed assignment)
Step 24: Now execute the task at assigned possible speed at the end of the task set
Step 25: if available energy is less than or equal to zero
Step 26: Then calculate the common available slack period to execute the task in ready queue when energy will be zero
Step 27: and if the energy is less and slack availability is also less than zero
Step 28: Now recharge the battery and wait for that
Step 29: and if the time is greater than coming task then insert the mix task set in the ready queue
Step 30: Stop all process

Algorithm for speed assignment for mix task set (AFSA)

This algorithm will help the IHA-RTS for assignment of the speed level so that we can saved the maximum power by adjusting the speed level of the processor to consuming the energy of the battery.
This algorithm can also be useful for any real time application for assigning of the speed level to execute the mixed task set to accomplish the deadline parameter of the task.

Step 1: Start by preparing the possible speed level in the process
Step 2: By increasing order of the speed limit it may sort
Step 3: if assigned speed at any level is equal to n level
Step 4: then calculate the energy level of the battery with harvested energy rate is subtracted by consumed energy to execute the task at this level
Step 5: And the energy consumed by this speed level the task should complete the deadline then assign this speed level to the processor
Step 6: Otherwise we increase the one step size on the speed level
Step 7: And Again go to the step 4
Step 8: End

This assignment technique works for both periodic as well as aperiodic task set for ant real time battery application.

V. EVALUATION AND DISCUSSION

This proposed approach is implemented and executed by using RTOSX. For verifying this approach, we are going to see this approach performance by using an example. This example is tested and show the result and compare with the existing approach that shows the improvement over the performance in different parameters.

For performance evaluation we are having the remaining energy available in the battery after successfully completion of the task set. And the second parameter for performance evaluation is acceptance ration of the task set. Here we check the variation in mix task set load in the system to saving the reaming energy of the battery and meet all requirement of the system.

We are having a three mix task set those are preemptive in nature are T1 with arrival time 0, execution time 2 and their deadline is 10, as same Task T2 having value 0, 3 and with 15 deadline and last task set T3 having attributed 1, 2 and 6.

This task is going to be schedule using battery power and that battery rechargeable and managed by harvesting energy resource, solar energy. Now we assumed some parameters like rate of recharge of battery in day time is 1J/Sec and in night time it may be zero due to solar power.
And we also assume that we are having three level of speed that are the requirement of the AFSA. And we also assumed that the highest level of the frequency used energy by a constant rate 1J/sec and at any middle level that are uses the 3J/sec and at higher speed level that is 8J/sec.
So that higher speed level used the highest energy consumption and consume more n more battery power.
So in this case we also assumed the total power availability of the battery is 32J. By using such assumption, we are going to schedule and execute the above mix task set.
As we see in our assumption that we are having the sufficient battery backup to execute the whole task set in full of speed level that is 8J/sec but according to our algorithm. If the slack period is available, we never go with higher speed level of the processor, and reserve the time slot of the task with their assigned speed by AFSA algorithm. In all aspect we schedule this task set as;

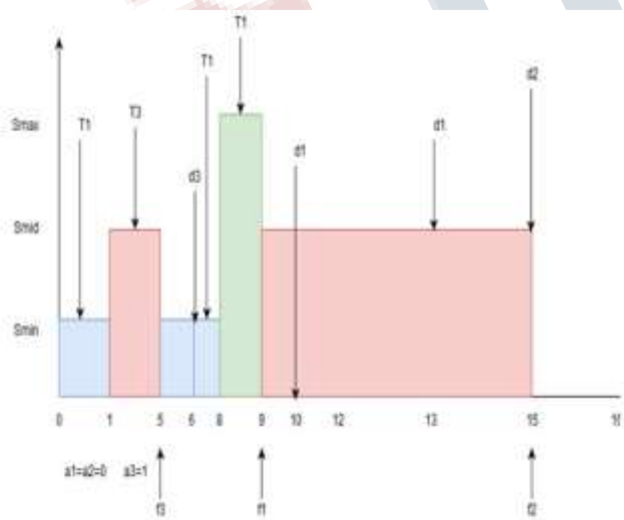


Fig.1 Schedule of the Mixed Task set with energy constraint at different level of speed assignment.

This task set should be schedule without missing of any task deadline and save the battery power up to 3J that can be used in the future coming tasks, that is the motive of this proposed algorithm.

Now we compare the result of our proposed approach IHA-RTS with the Energy aware dynamic voltage and frequency selection (EA-DVFS) [16]. We compare the result in different parameter like acceptance ration of the task and saved remaining energy on the battery.

We see in the figure 3 and 4 how much our proposed algorithm gives better performance in compare to exiting algorithm.

See in figure 3 we have drawn a comparison by using factor mix set periodic load varying from 0.1 to 1.5 and average remaining battery backup after task set execution varying from 0 to 100.

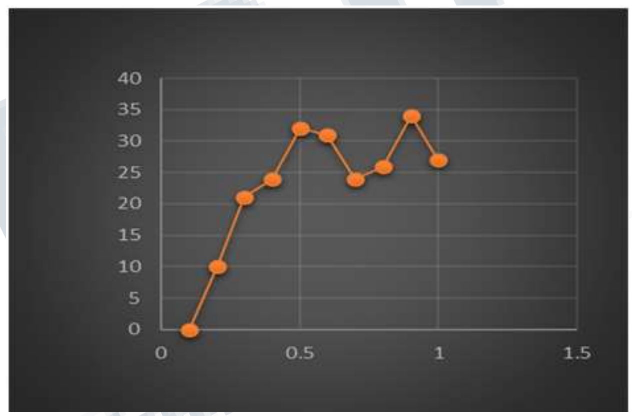
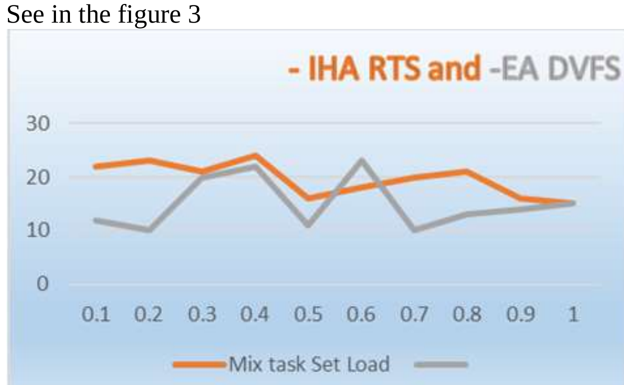


Fig 2: Remaining Battery Energy After Task Execution

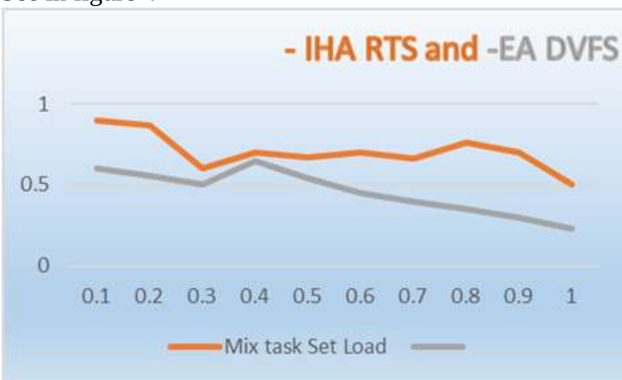
We observe in this figure after a time if we increase the system load then we decrease the saving battery power average.

See in the figure 3



We observe that average saving battery energy better than EA DVFS in less load and same in when we have more task load in the system.

See in figure 4



This experimental result shows that our proposed algorithm is also better in acceptance ration that shows in figure 4 that is varies 0 to 0.5 and 1. Our proposed algorithm gives better performance result that shows in this experiment and the performance decrease in some point when system load increases.

CONCLUSION

In this paper we present an optimal algorithm that works in harvesting energy environment. The experimental result shows that this algorithm allows the task to execute on decided operating speed that will be reduce the overall energy consumption as well as never miss their deadline by using switching speed of the processor and help the recharging of the battery by solar energy system. The example and experimental result shows that this algorithm improves the storage capacity of the battery as well as the acceptance ratio of the task set. This approach performance slowdown when we increase system load. So for future goal is to improve this algorithm so that if we increase the system load then the performance ratio should not be slow down.

REFERENCES

[1] Agrawal, S., Yadav, R. S. and Ranvijay, 2009. A Pre- emption Control Approach for Energy Aware Fault Tolerant Real Time System, International Journal of Recent Trends in Engineering, 381-386.

[2] Allavena, A. and Mosse, D., 2001. Scheduling of frame-based embedded systems with rechargeable batteries, Workshop Power Manage Real-time Embedded System.

[3] Bertogna, M. and Baruah, S., 2010. Limited Preemption EDF Scheduling of Sporadic Task Systems, IEEE Transactions on Industrial Informatics, 579 - 591.

[4] Chetto, M. and Zhang, H., 2010. Performance Evaluation of Real- Time Scheduling Heuristics for Energy Harvesting Systems, EEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom),Green Computing and Communications (GreenCom), 398 - 403.

[5] Dehghan and Maryam, 2010. Adaptive checkpoint placement in energy harvesting real-time systems , 18th Iranian Conference on Electrical Engineering (ICEE), 932 - 937.

[6] Hussein, E. L. G., Chetto, M. and Chehade, R.H., 2011. EH-EDF: An On-line Scheduler for Real-Time Energy Har- vesting Systems, 18th IEEE International Conference on Electronics Circuit and System (ICECS), 776-779.

[7] Hussein, E. L. G., Chetto, M. and Chehade, R.H., 2011. A real-time scheduling framework for embedded systems with environmental energy harvesting, Elsevier on Computers and Electrical Engineering 37.

[8] Liu, S., Lu, J., Wu, Q. and Qiu, Q., 2011. Harvesting-Aware Power Management for Real-Time Systems With Renewable Energy, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1 - 14.

[9] Liu, S., Lu, J., Wu, Q. and Qiu, Q., 2010. Load-matching adaptive task scheduling for energy efficiency in energy har- vesting real-time embedded systems, ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), 325 – 330.

[10] Liu, S., Qiu, Q. and Wu, Q., 2008. Energy Aware Dynamic Voltage and Frequency Selection for Real-Time Systems with Energy Harvesting, conference on Design, Automation and Test in Europe, 236 - 241.

[11] Lu, J. and Qiu, Q. 2011. Scheduling and mapping of periodic tasks on multi-core embedded systems with energy harvesting, Journal, Computers and Electrical Engineering archive, 498 - 510.

[12] Moser, C., Brunelli, D., Thiele, L. and Benini, L., 2007. Real-time scheduling for energy harvesting sensor nodes, Real-Time Syst ; 37(3), 233-260.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering
(IJERCSE)
Vol 5, Issue 2, February 2018

[13] Moser, C., Thiele, L., Brunelli, D. and Benini, L., 2007. Adaptive Power Management in Energy Harvesting Systems ,
Design, Automation and Test in Europe Conference and Exhibition, 1 - 6.

[14] Niu, L. and Quan, G. 2006. System-Wide Dynamic Power Man- agement for Portable Multimedia Devices, Eighth IEEE International Symposium on Multimedia, 97 -104 .

[15] Niu, L. and Quan, G., 2006. Energy minimization for real-time systems with (m,k)-guarantee, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 717-729.

[16] Paul, A. A. and Pillai, A.S. B., 2011. Reducing the Number of Con- text Switches in Real Time Systems, International Conference on Process Automation, Control and Computing (PACC), 1 - 6.

[17] Qadi, A., Goddard, S. and Farritor, S., 2003. A dynamic voltage scaling algorithm for sporadic tasks , 24th IEEE Conference on Real- Time Systems Symposium (RTSS), 52 - 62.

[18] Zhu, L., Tongquan, Wei, T., Yonghe, Guo., Xiaodao, Chen. and Shiyan, Hu., 2010. Energy efficient fault-tolerance task allocation scheme for real-time energy harvesting systems, International Conference on Intelligent Control and Information Processing (ICICIP), 589 - 594.