Real-Time Task Scheduling for Dynamically Variable Voltage Processors*

Yi-Ping You, Chingren Lee, Jenq-Kuen Lee, Wei-Kuan Shih
Department of Computer Science
National Tsing-Hua University
Hsinchu, Taiwan
{ypyou, crlee}@pllab.cs.nthu.edu.tw, {jklee, wshih}@cs.nthu.edu.tw

Abstract

With the demands of power-constrained mobile applications and devices, it becomes a crucial and challenging issue to reduce the power consumptions of embedded systems. In this paper, we focus on the issue of scheduling problems with variable voltage processor core to optimize power consumption in real-time systems. We model the problem with real-time and on-line problems, and our solution is to incorporate the reservation list scheme for variable voltage schedulings. Our decision algorithm consists of a variety of selection criteria including the best effort, average computation time, average power consumption, average energy consumption, pre-defined threshold value, and weighted hybrid schemes for scheduling task. We think our scheme gives a comprehensive study for the problem of scheduling real-time tasks to reduce energy consumptions.

1. Introduction and Related Work

The demands of power-constrained mobile and embedded computing applications increase rapidly. Reducing power consumption hence becomes a crucial challenge for today's software and hardware developers. In CMOS circuits, power is dissipated in a gate when the gate output changes from 0 to 1 or from 1 to 0. Minimization of power dissipation can be considered at algorithmic, architectural, logic and circuit levels [2]. Studies on low power design are abundant in the literature in which various techniques were proposed to synthesize designs with low transitional activities. In this paper, we focus the issue on scheduling problems with variable voltage processor core to optimize power consumption in real-time systems.

Scheduling problems in real-time systems have been largely issued by many researchers [9, 1, 14, 15]. Liu et al. [9] studied the periodic task scheduling and gave an elegant worst case utilization bound for the famous "rate monotonic" scheduling algorithm. Baruah et al. [1] gave the on-line real-time scheduling which has a nice upper bound. Shih et al. [15, 14] gave several on-line algorithms for scheduling real-time tasks in the imprecise computation model. However, power-aware real-time systems are addressed lately. Energy minimization by scaling voltage was first studied in [18]. Hong et al. [5, 6, 4] proposed several heuristics scheduling and synthesis techniques. Ishihara et al. [7] presented a scheduling considering switching activties that affect power consumption a lot. Swaminathan et al. [17] presented a novel low-energy earliest-deadline-first(LEDF) scheduling algorithm and applied it to two real-life task sets. Childers et al. [11] integrated compiler-assisted techniques with power-aware systems and presented scheduling techniques. And other researchers [16, 10, 13] also presented their approach and gave experiments.

2. Preliminaries

In this Section, we describe the task model, power model, and variable voltage model for systems underlaying the assumptions.

2.1. Task Model

Given a set of n periodic tasks $T = \{T_1, T_2, \ldots, T_n\}$. Each task $T_i(c_i, d_i, p_i, \alpha_i) \in T$ is characterized by its computation time c_i at reference voltage, hard

* This work was supported in part by National Science Council under Grant No. NSC-89-2218-E-007-023 and NSC-89-2219-E-007-012 and by Computer & Communications Research Labs, Industrial Technology Research Institute under Grant No. 3XT1200.

deadline d_i, period p_i, and average switching activity α_i. The ready (or arrival) time, a_i, of the task occur periodically with period p_i. Mostly, the deadlines are assumed to be equal to the periods. The switching activity of a task is related to its type and can be obtained by simulation. It acts as an important factor dominating power consumption.

For this paper, tasks are assumed to be non-preemptable so that once a task starts running, no other tasks can be executed until it completes execution.

2.2. Power Model

It is indicated that power dissipation in digital CMOS circuits is composed of three major components: switching power, short-circuit power, and leakage power. The switching power, which leads the total power consumption, is given by

$$P = \alpha C_L V_{dd}^2 f_{clk},$$

where α is the switching activity, C_L the load capacitance, V_{dd}^2 the supply voltage, and f_{clk} the clock frequency [2]. The most effective way to reduce power consumption is to lower the supply voltage, which has a quadratic dependence on power. However, for any task T_i, reducing the supply voltage increases critical path delay τ_c and hence computation time c_i, which is given by $c_i = s_i \tau_c$ where s_i is the number of required control cycles. The critical path delay is strongly depended on the supply voltage by

$$\tau_c(V_{dd}) = k' C_L \tfrac{V_{dd}}{(V_{dd}-V_T)^2},$$

where k' is a constant and V_T is the threshold voltage [2].

For a set of n tasks, we can compute the average power consumption and total energy consumption by $P_{average} = \frac{1}{n}\sum_{j=1}^{n} \alpha_j C_L V_{dd_j}^2 f_{clk_j}$ and $E_{total} = \sum_{j=1}^{n} \alpha_j C_L V_{dd_j}^2 s_j$, respectively.

2.3. Variable Voltage Model

With a DC-DC switching regulator, the voltage can be adjusted as varying as we want. However, using continuously variable voltage is not feasible since it needs large cost to provide any kinds of stable supply voltages. In this paper, we assume that two supply voltages are allowed: high voltage V_H and low voltage V_L. Tasks can be scheduled either at high or low voltage, which is corresponding to different clock frequencies, to reduce the power consumption.

As indicated in [12], the time overhead for a DC-DC switching converter is about 6 micro-second per transition of voltage. Also, the computation can still continue during the period of varying voltages. Hence, we assume the overhead to be ignored in the proposed scheduling.

3. Variable Voltage Scheduling

It has been known that the scheduling problem of a set of non-preemptable independent tasks on a fixed voltage processor is NP-complete [3]. With reduction techniques, we can easily see the same scheduling problem on a variable voltage processor is NP-hard. To optimize the power or energy consumption in real-time systems, we propose an heuristic algorithm scheduling tasks by deadlines with variable voltages.

The proposed scheduler maintains a list, called *reservation list* [15], in which tasks are sorted by deadlines. Due to the characteristics of periodic tasks, tasks arrive periodically with certain periods, we have knowledge of the arrivals and deadlines of tasks in a given interval. In the beginning, all tasks are in the list and sorted by their deadlines. The task with the earliest deadline is then picked to schedule. The scheduler checks if the task is able to execute at a low voltage by deadline without influencing any unscheduled task, which is in the reservation list. It will schedule tasks at low voltage if possible. However, it is usually impossible to schedule all tasks at low voltage by their deadlines since the computation regions extend while lower voltage being supplied. There are perhaps only some tasks that can be scheduled at low voltage. Thus, deciding which tasks to be scheduled at low voltage determines the efficency of processor utilization and the overall power consumption. A good decision derives a good scheduler and minimizes the power or energy consumption. We present several decision algorithms for reducing the power consumptions.

3.1. Scheduling Algorithm

The proposed scheduling algorithm is based on the well-known *EDF (Earliest Deadline First)* algorithm, which as the name implies always execute that task with earliest deadline [9]. Assume there are n periodic tasks to be scheduled. We first sort tasks in ascending order by deadlines, namely T_1, T_2, $\ldots$, T_n, and put them in a list, i.e. the reservation list, which keeps unscheduled tasks. And then, extract tasks from the list one by one to schedule. The detailed description of scheduling algorithm is showed in Figure 1.

The slack time $\delta_i(V)$ means the maximum time interval allowed for current task T_i to execute, while all the remaining tasks in the reservation list are scheduled with supply voltage V. We compare the compu-

Real-Time Scheduling Algorithm with Variable-Voltage Reservation Lists

Input: n unscheduled periodic tasks
Output: Schedule of the n tasks with variable voltages

1. Sort tasks by deadlines in ascending order, i.e. T_1, T_2, ..., T_n.
2. Put them in a list, called *reservation list.*

Repeat 3-6 while the reservation list is not empty.

3. Remove the first task, namely T_i, which has the earliest deadline from the reservation list.
4. Update the slack time of tasks in the current list with both high and low voltage pseudo scheduler, i.e. $\delta_i(V_H)$ and $\delta_i(V_L)$.
5. Compute the computation time of T_i at high voltage and that of low voltage, i.e. $c_i(V_H)$ and $c_i(V_L)$.
6. Schedule T_i with the following manners.
   - If $c_i(V_L) \le \delta_i(V_L)$, schedule T_i at low voltage if possible.
   - If $\delta_i(V_L) < c_i(V_L) \le \delta_i(V_H)$, call *decision algorithm.*
   - If $c_i(V_L) > \delta_i(V_H)$ and
     - if $c_i(V_H) \le \delta_i(V_H)$, schedule T_i at high voltage.
     - if $c_i(V_H) > \delta_i(V_H)$, report possible failures of real-time scheduling.

Figure 1. Reservation list scheduling algorithm for variable voltage problems

tation time of task T_i at both high and low voltages, i.e. $c_i(V_H)$ and $c_i(V_L)$, with $\delta_i(V_L)$ and $\delta_i(V_H)$. There are only three conditions: 1) If $c_i(V_L)$ is smaller or equal than $\delta_i(V_L)$, we can schedule task T_i at low voltage without affecting any task in the future because of no overlaps between task T_i and the unscheduled tasks while those tasks are assumed to be executed at low voltage. 2) $c_i(V_L)$ is greater than $\delta_i(V_L)$ and smaller or equal than $\delta_i(V_H)$. If this happens, we call a decision algorithm to decide if task T_i should be scheduled at low or high voltage. It weighs the alternatives to optimize the overall cost, such as power or energy consumption. 3) If $c_i(V_L)$ is greater than $\delta_i(V_H)$, it means that it is impossible for task T_i to complete jobs at low voltage by its deadline, we can only schedule it at high voltage if the deadline is met. Figure 2 shows the three scenarios.

The computation of slack time of tasks and the decision algorithm used in this algorithm will be described

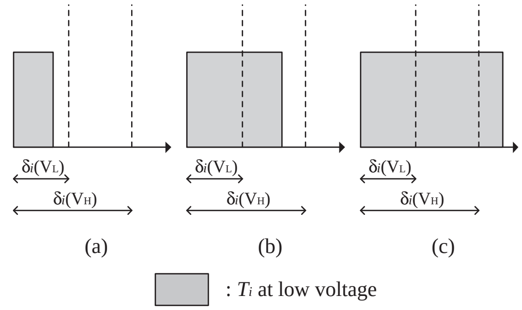

Figure 2. Scenarios of scheduling task T_i.

shortly in Section 3.2 and 3.3.

3.2. Slack Computation

With the assumption in Section 3.1, suppose that we are now going to schedule task T_i, and hence there are still $(n-i)$ unscheduled tasks, i.e., T_{i+1}, T_{i+2}, $\ldots$, T_n, in the reservation list. Recall that the slack time $\delta_i(V)$ is the maximum period allowed for T_i while the remaining $(n-1)$ tasks are scheduled at supply voltage V. To obtain the information for T_i, we first build a pseudo schedule for the $(n-i)$ tasks in the following behaviors. The $(n-i)$ tasks are scheduled in a reversed way, treating the deadlines as arrivals and the arrivals as deadlines and starting from the point of the latest deadline, i.e. d_n the deadline of T_n, by using EDF algorithm [9]. We then record the timing point of the beginning point of the pseudo schedule as $\lambda_i(V)$.

The slack time of the pseudo schedule with supply voltage V can be obtain by the following equation:

$$\delta_i(V) = \lambda_i(V) - Max(a_i, f_{i-1}),$$

where a_i is the arrival time of T_i, f_{i-1} is the finished time of the last task T_{i-1}, and $Max(a, b)$ is a function that returns the maximum value between a and b. Figure 3 gives an example of the slack computation. In Figure 3, there are four tasks in the reservation list. In the reservation list, one is done by a pseudo scheduler to schedule tasks by low voltage, and another is done by high voltage scheduler. The slack time $\delta_i(V_H)$ or $\delta_i(V_L)$ is the time from the finishing time of the last task to the beginning point of the reservation list by each voltage scheduler, respectively.

It should be noted that during the scheduling, we should call exception if any deadline cannot be met when scheduling at high voltage since the forward and backward scheduling are equivalent on the qualification of time-constrained tasks. If there is no backward schedule, there is also no forward schedule. However, when low voltage is supplied, we ignore deadline misses.

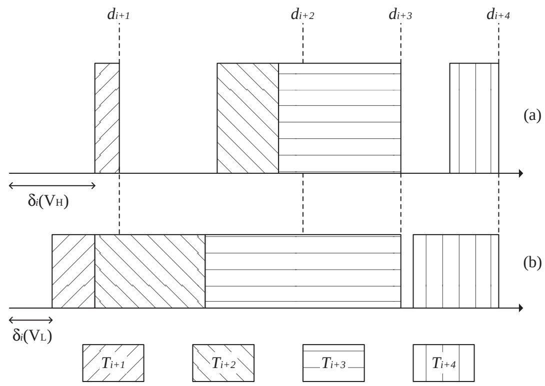

Figure 3. Examples of slack computation while scheduling T_i. (a) tasks at high voltage (b) tasks at low voltage

3.3. Decision Algorithm

Following the notations in the previous sections, assume that we are now scheduling task T_i and the computation time of T_i at low voltage, $c_i(V_L)$, is greater than $\delta_i(V_L)$ and smaller or equal than $\delta_i(V_H)$. It looks as the finishing time of task T_i at low voltage falls into the region bounded by $\lambda_i(V_L)$ and $\lambda_i(V_H)$. To achieve the objective of power or energy optimization, we propose several algorithms to decide if tasks should be scheduled at low or high voltages when weighing trade-offs between tasks. Six approaches are described as follows.

1. *RL-FFS (Reservation List with First-come First-serve Scheduling)*
   As the name implied, this approach provides no cost model for decision. It just schedule the task with the same behavior, scheduling it at low voltage. Here, we just schedule T_i at low voltage if possible and hence someone behind T_i must be scheduled at high voltage. Since no further decision is made, it is efficient to run but may not optimal for power and energy reduction.

2. *RL-ACT (Reservation List with Average Computation Time)*
   When the computation time of T_i at low voltage, $c_i(V_L)$, is in the case $\delta_i(V_L) < c_i(V_L) \le \delta_i(V_H)$, we compare $c_i(V_L)$ with the average computation time of all tasks. Comparing the computation times helps us to figure out if T_i occupies much more time than the others, which means requiring more energy. Recall that $c_i = s_i\tau_c$ in Section 2.2. The computation time of T_i depends on s_i, the number of control cycles required by T_i, while τ_c is constant when the supply voltage is fixed. Therefore, comparing the computation time is just comparing the required control cycles. The average computation time, on the order of average required control cycles, is given by

$$AVG_{CT} \cong AVG_s = \sum_{j=1}^{n} \frac{\epsilon_j \ s_j}{k},$$

   where ϵ_j is 1 if $c_j(V_L) \le p_j$, the period of T_j, and 0 otherwise and k is the number of the tasks whose value of ϵ is 1. If $s_i > AVG_s$, we schedule T_i at low voltage if possible.

3. *RL-APC (Reservation List with Average Power Consumption)*
   As mentioned in Section 2.2, power consumption of a task is computed by $\alpha_i C_L V_{ddi}^2 f_i$, where C_L, V_{ddi}^2, and f_i are constants when comparing cost among tasks with a certain voltage and frequency while the task-dependent parameter, i.e. switching activity α_i, dominates the power consumption. Therefore, we can treat power consumption as switching activity while making comparison among tasks. Similarly to the RL-APC, the average power of the tasks is used to guide the scheduling policy. The average of power consumption, on the order of switching activity, is given by

$$AVG_P \cong AVG_\alpha = \sum_{j=1}^{n} \frac{\epsilon_j \ \alpha_j}{k},$$

   where α_j is the switching activity of task T_j. We schedule T_i at low voltage if possible while $\alpha_i > AVG_\alpha$.

4. *RL-AEC (Reservation List with Average Energy Consumption)*
   It is similar to power consumption while the energy consumption is computed by $\alpha_i C_L V_{dd_i}^2 s_i$. The switching activity α_i and the required control cycles s_i determine the energy consumption. We just compare the product of α_i and s_i among tasks. The average energy consumption is given by

$$AVG_E \cong AVG_{\alpha\star s} = \sum_{j=1}^{n} \frac{\epsilon_j \ \alpha_j \ s_j}{k}.$$

   If $(\alpha_i \star s_i) > AVG_{\alpha\star s}$, we schedule T_i at low voltage if possible.

5. *RL-PTV (Reservation List with Predefined Threshold Value)*
   This approach is based on computation time

of tasks with a predefined critical value which determines the supply voltage of tasks. If c_i does not cross over the threshold, we schedule T_i at low voltage and high voltage otherwise. The critical value should be an arbitrary value between $\delta_i(V_L)$ and $\delta_i(V_H)$. We think that it will perform better results if the value is closer to $\delta_i(V_L)$ since it may impact less tasks in future while high voltage is supplied to T_i.

6. *RL-WHS (Reservation List with Weighted Hybrid Scheme)*
   In this approach, we use a weighted hybrid scheme that chooses parts of the previous five selectors as the decision maker to decide the supply voltage of task T_i. The scheme is done by voting schemes by running through parts or all of the previous schedulers to give the one with most votes for scheduling.

4. Experiments and Discussion

In this Section, we extracted the experimental results from a series of more than one hundred tasks that are ones of a CNC(Computerized Numberical Control) machine controller [8]. Our simulated system is based on a microprocessor which can process with two modes: high supply voltage and low supply voltage. The maximum operating frequency is assumed as 100MHz at 5V supply voltage and the minimum is 50.8MHz at 3V, where the threshold voltage is 0.5V. The characters of the CNC task set is showed in Table 1. We can easily get the computation time of both high and low voltage by the formular described in Section 2.

$Task$	$c_i(5V)$	$c_i(3V)$	d_i	p_i	$\alpha_i(\%)$
T_{smpl}	35μs	68μs	2400μs	2400μs	92
T_{calv}	40μs	78μs	2400μs	2400μs	77
T_{dist}	180μs	350μs	4800μs	4800μs	15
T_{stts}	720μs	1400μs	4800μs	4800μs	7
T_{xref}	165μs	321μs	2400μs	2400μs	69
T_{yref}	165μs	321μs	2400μs	2400μs	18
T_{xctrl}	570μs	1108μs	4000μs	9600μs	67
T_{yctrl}	570μs	1108μs	4000μs	7800μs	34

Table 1. Summary of the CNC task set

To evaluate the proposed RL-scheduling algorithm and the decision methods, tasks are periodically repeated with periods without missing its deadline and the consumed power and energy are measured to evaluate. The total power consumption is illustrated in Figure 4, where the X-axis represents the total number of tasks to be scheduled. It says the variable voltage scheduling is about 62% power reduction against the fixed voltage system. The comparison of results among the six selectors is shown in Figure 5. It seems that the RL-APC algorithms made the better decisions while RL-ACT scheduling is the worst. This is because the RL-APC decides tasks to process at high or low voltage by determining their power consumption while RL-ACT only looks at their computation time. The PTV (Predefined Threshold Value) of RL-PTV is 0.9 (90%), which is considered the modest value according to the results in Figure 6. The value acts a threshold between the ending points of both reservation list to decide what voltages should tasks be scheduled. In this experiment, the RL-FFS, RL-APC, RL-AEC, and RL-PTV all shows good results. However, their scheduling resutls may be performed variantly due to the characters of the task set. The RL-WHS, here, deals with the problem by combining those methods together and weighting their decisions. The RL-WHS takes the most used ones as its decision-maker every time it decides. It almost results good power reductions.

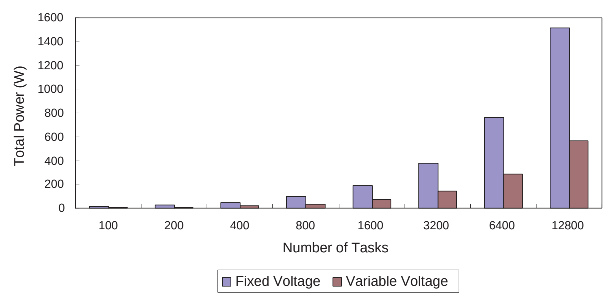

Figure 4. Total power consumption of tasks.

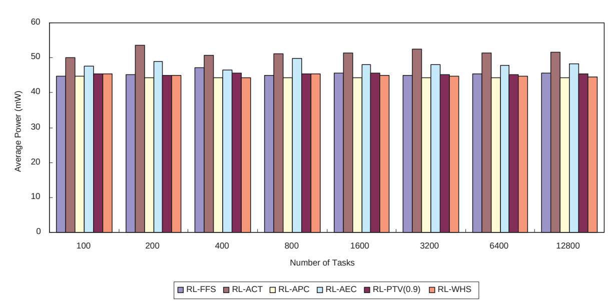

Figure 5. Average power consumption of tasks.

In this paper, we proposed a heuristic algorithm

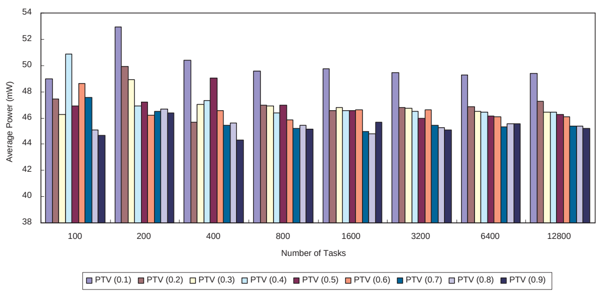

Figure 6. Average power consumption of tasks with different PTVs.

with incorporation of the reservation list scheme for real-time variable voltage scheduling and also a decision algorithm consists of a variety of selection criteria focus on those scheduling problems including the best effort, average computation time, average power consumption, average energy consumption, predefined threshold value, and weighted hybrid schemes for scheduling task. We think our scheme gives a comprehensive study for the problem of scheduling real-time tasks to reduce energy consumptions.

References

[1] S. Baruah, G. Koren, D. Mao, B. Mishra, A. Raghunathan, L. Rosier, D. hasha and F. Wang. "On the competitiveness of on-line real-time task scheduling", *Proc. of the 12th Real-Time Systems Symposium, San Antonio, Texas,* pp. 106-115, December 1991.

[2] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits,* Vol. 27, No.4, pp. 473-484, April 1992.

[3] M. R. Garey and D.S. Johnson, "Computer and Intractability: A Guide to the theory of NP-Completeness," *W. H. Freeman & Co., New York, NY,* 1979.

[4] I. Hong, M. Potkonjak, and M. Srivastava, "On-Line Scheduling of Hard Real-Time Tasks on Variable Voltage Processor," *Proc. of the International Conference on Computer-Aided Design (ICCAD'98),* Nov. 1998.

[5] I. Hong, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power Optimization of Variable Voltage Core-Based Systems," *Proc. of 35th DAC,* pp. 176-181, 1998.

[6] I. Hong, G. Qu, M. Potkonjak, and M. B. Srivastava, "Synthesis techniques for low-power hard real-time systems on variable-voltage processors," *Proc. of Real-Time Systems Symposium,* pp. 178-187, 1998.

[7] T. ISHIHARA, and H. YASUURA, "Voltage Scheduling Problem for Dynamically Variable Voltage Processors," *Proc. of International Symposium on Low Power Electronics and Design (ISLPED'98),* August 1998.

[8] N. Kim, M. Ryu, S. Hong, M. Saksena, C.-H. Choi, and H. Shin, "Visual Assessment of a Real-Time System Design: A Case Study on a CNC Controler," *Real-Time Systems Symposium, 1996., 17th IEEE,* pp.300-310, 1996.

[9] C. L. Liu and J. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Read-Time Environment," *Journal of the ACM,* vol. 20, pp. 46-61, 1973.

[10] A. Manzak and C. Chakrabarti, "Variable Voltage Task Scheduling for Minimizing Energy or Minimizing Power," *Proc. of the International Conference on Acoustics, Speech and Signal Processing,* June 2000.

[11] D. Mosse, H. Aydin, B. Childers, and R. Melhem, "Compiler-Assisted Dynamic Power-Aware Scheduling for Real-Time Applications," *COLP (Workshop on Compiler and OS for Low Power),* Philadelphia, Oct 19, 2000.

[12] Won Namgoong, Mengchen Yu, and Teresa Meng, "A High-Efficiency Variable-Voltage CMOS Dynamic DC-DC Switching Regulator," *IEEE International Solid-State Circuits Conference,* pp. 380-381, 1997.

[13] T. Pering, T. Burd, R. Brodersen, "The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms," *Low Power Electronics and Design, 1998. Proceedings. 1998 International Symposium on,* pp.76-81, May 1998.

[14] Wei-Kuan Shih, Jane W. S. Liu, and Jen-Yao. Chung. *Algorithms for scheduling imprecise computations with timing constraints*, SIAM J. on Computing, Vol. 20, No. 3. 1991.

[15] Wei-Kuan Shih, Jane W. S. Liu, *On-line Algorithm for Scheduling Imprecise Computations*, SIAM J. on Computing, Oct(1996). pp. 1105-1121.

[16] Y. Shin and K. Choi, "Power conscious fixed priority scheduling for hard real-time systems," *Proc. Design Automation Conference (DAC),* pp.134-139, June 1999.

[17] V. Swaminathan and K. Chakrabarty, "Real-time task scheduling for energy-aware embedded systems," $IEEE\ Real\text{-}Time\ Systems\ Symposium\ Work-In-ProgressSessions$, 2000.

[18] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on,* pp.374 -382, June 1995.