

Energy Minimization of Real-time Tasks on Variable Voltage Processors with Transition Energy Overhead

Yumin Zhang

Synopsys Inc.
700 East Middlefield Road
Mountain View, CA 94043, USA
e-mail: yumin@synopsys.com

Xiaobo Sharon Hu Danny Z. Chen

Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
e-mail shu, dchen@cse.nd.edu

Abstract— In this paper, we address the problem of minimizing energy consumption of real-time tasks on variable voltage processors whose transition energy overhead is not negligible. Voltage settings with minimum number of transitions are found first and sequences of lower voltage cycles are evaluated to decide voltage for each cycle of every task. Experimental results demonstrate that our approach can reduce energy consumed by transitions from 41% to 8% and save more energy.

I. INTRODUCTION

Two main system level energy saving techniques are: voltage selection (VS) (also called voltage scheduling [2]) which selects a processor's supply voltage according to tasks requirement, and power management (PM) [1] which shuts down a processor when it is idle. It has been shown that applying VS judiciously can achieve a large amount of energy saving [11].

Even though transitions of voltage values can be done on the fly, incurred energy overhead should be considered [2]. If not handled wisely, energy consumption by transitions can become a dominant part and offset the benefit of having various voltages. Since processors can still execute instructions during transitions [3], and timing overhead is linear, while energy overhead is in quadratic, to voltage difference, we concentrate on energy overhead.

The approaches in [7, 9] consider overhead while running independent tasks on a single variable voltage processor. However, tasks in real-world applications usually have control or data dependencies and many systems have multiple processors. Approaches in [4, 8, 12, 14] tried to minimize energy of dependent tasks on multiple variable voltage processors. But energy overhead is not considered. When energy overhead can be ignored, only the number of cycles at each voltage needs to be determined and that determines the total energy consumption. The sequence of voltage levels does not affect

the energy. Our paper is the first effort that addresses the energy minimization problem with consideration of transition energy overhead on multiprocessor systems.

Our approach, Energy Overhead Consideration (EOC) approach, can work on the voltage selection resulted by approaches in [4, 8, 12, 14], to improve energy saving when overhead cannot be ignored. We first find a voltage setting with the minimum number of transitions within a task (intra-task) and between consecutive tasks (inter-task) while keeping the number of cycles at different voltages determined by [4, 8, 12, 14] unchanged. Then we evaluate each sequence of lower voltage cycles. Only when energy saving of running some consecutive cycles at a lower voltage is greater than energy consumed by transitions involved, will these transitions happen. Our approach can also be used when timing overhead is not negligible. Minimizing the number of transitions is beneficial in minimizing the effect of both timing and energy transition overhead. In the case of non-negligible timing overhead, each sequence of different voltages is checked to see whether timing constraints can still be met when the transition overhead is considered. Lower voltage will be allowed only when timing constraints can still be met. During our study, we found cases that more than two voltages are needed to minimize energy even when overhead can be ignored. This finding alerts us that the claim made in [6] that at most two voltages are necessary to minimize energy has strong conditions and should be used with cautions.

We apply our EOC approach to the VS results by approach in [14] in experiments. The results show that energy overhead has big impact on energy saving. By ordering voltage sequences wisely, we can decrease the number of transitions by 27% comparing with a policy which always starts a task from its highest voltage. Our approach eliminates unbeneficial sequences and reduces energy consumed by transitions from 41% to 8% while saving more energy at the same time. Our approach can save 7.7-25.8% of energy for different overhead.

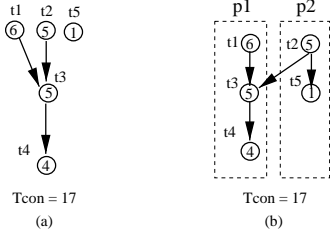


Fig. 1. (a) A 5-task set. (b) Tasks scheduled on P_1 and P_2

In the rest, we describe preliminaries in Section 2. In Section 3, we present our EOC approach that minimizes the number of transitions and improves savings. Experimental results are presented in Section 4 and the paper concludes in Section 5.

II. PRELIMINARIES

The number of cycles that task u takes to finish, N_u , cannot be changed. The VS process changes supply voltage, which in turn changes a processor's cycle time, task u 's delay and the dominant part of the total energy consumption, dynamic energy consumption. For task u , energy saving per V_l cycle when voltage is decrease from V_h to V_l , ΔES_u , is

$$\Delta ES_u = C_u \cdot |V_h^2 - V_l^2| \quad (1)$$

Note that tasks can have different power characteristics, such as effective switching capacitance C_u .

A voltage/frequency converter is needed to supply different voltages. Energy overhead, EO , of a typical voltage/frequency converter when voltage switches between V_h and V_l can be computed as in [2]

$$EO = (1 - \eta) \cdot C_{DD} \cdot |V_h^2 - V_l^2| \quad (2)$$

where η is the efficiency of the DC-DC converter in the voltage/frequency converter and C_{DD} is the capacitor that stores the charge. (1)-(2) tell us that energy can be saved after paying transition overhead if there are enough consecutive cycles running at V_l between transitions.

III. CONSIDERING ENERGY OVERHEAD

In this section, we first show a motivational example in which energy will be consumed unnecessarily if transition overhead is not considered. Then we present our EOC approach that decides a voltage level for each cycle of every task based on the given number of cycles on each voltage in a VS results.

A. A motivational example

The following example motivated us to consider energy overhead while deciding voltage levels for task cycles. Consider a 5-task set and its scheduling on two processors, shown in Figure 1 (a) and (b). In the figure, nodes

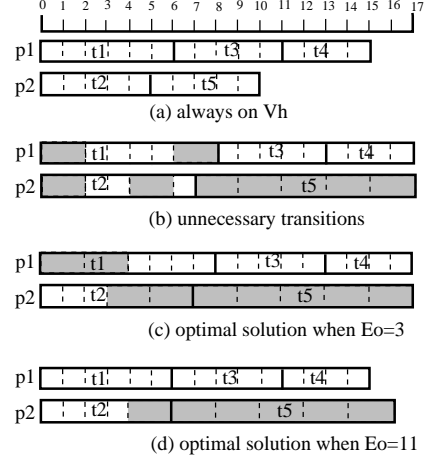


Fig. 2. Different Voltage Settings. Cycles in shade are executed at V_l .

TABLE I
ENERGY OF SETTINGS FOR DIFFERENT OVERHEAD

	N_{V_h}	N_{V_l}	N_{tr}	E_0	E_3	E_{11}
(a)	25	0	0	100	100	100
(b)	16	9	7	73	94	150
(c)	16	9	2	73	79	95
(d)	19	6	1	82	85	93

represent tasks and edges represent dependencies. The number inside each task is the number of cycles needed to finish the task.

Assume P_1 and P_2 can operate on $V_h = 2$ and $V_l = 1$. For simplicity, we estimate $CT_{V_h} = 1$, $CT_{V_l} = 2$, the energy consumption per cycle at V_h to be 4 and at V_l to be 1, and the energy saving per V_l cycle to be 3. Assume that the energy overhead per transition is 0, 3 and 11. Let the timing constraint be 17.

Figure 2 (a)-(d) show four different voltage settings for the scheduling in Figure 1 (b). The number of cycles at V_h and V_l , the number of transitions, and the total energy consumption when energy overhead is 0, 3, and 11 for the four voltage settings are shown in columns N_{V_h} , N_{V_l} , N_{tr} , E_0 , E_3 and E_{11} , respectively, in Table I.

When there is no transition energy overhead, settings in Figure 2 (b) and (c) are optimal solutions. However, when the energy overhead per transition is 3, system implementation in Figure 2 (c) is the optimal solution. When the energy overhead increases to 11, Figure 2 (d) is the optimal solution. This example shows that energy overhead affects overall saving dramatically and must be considered in determining the voltage levels.

B. More than two voltages needed

It is not always true that at most two voltages are needed to minimize energy in the discrete voltage case. The number of voltages needed depends on the voltage

values available. More than two voltages are needed when the combinations of two voltages cannot produce an execution time that is required to minimize the energy consumption. Denote two voltages as V_h and V_l where $V_h > V_l$, and the corresponding cycle time as CT_{V_h} and CT_{V_l} . For a task that takes N cycles to finish, there are in total N different execution times that the combinations of these two voltages can produce. They are

$$N_1 * (CT_{V_l} - CT_{V_h}) + N * CT_{V_h}, \quad N_1 \leq N \quad (3)$$

For any given V_h and V_l , CT_{V_h} and CT_{V_l} are constants. Since N is a constant for a given task, the N different execution times change in the step of $CT_{V_l} - CT_{V_h}$. If $CT_{V_l} - CT_{V_h} > 1$, the N different execution times are not consecutive integers, but rather an array of integers with the same increase step. If the target execution time determined for the best energy is in the middle of two of the N values, a third voltage is needed. Otherwise, the task will have to finish earlier and consume more energy than finishing right on time.

Theorem-1 in [6] states that if a processor can use only a small number of discrete variable voltages, the voltage scheduling with at most two voltages minimizes the energy consumption under any time constraint. There is a strong condition for this theorem to be applicable and that is the execution time by the combinations of the two voltages can meet timing constraint exactly.

C. Determining the voltage setting

When the energy overhead due to voltage transition is not negligible, energy consumed by transitions will offset the saving of running tasks on lower voltage levels. The total energy saving ES is computed as follows.

$$ES = \sum_u \sum_i N_{V_i,u} \cdot \Delta ES_{V_i,u} - NTR_{V_i,V_j} \cdot EO_{V_i,V_j} \quad (4)$$

where $N_{V_i,u}$ is the number of u 's V_i cycles and NTR_{V_i,V_j} is the number of transitions between voltage V_i and V_j . The first term $\sum_u \sum_i N_{V_i,u} \cdot \Delta ES_{V_i,u}$ is known on a given VS result.

To minimize energy consumption, we first find the voltage setting that has the minimum number of transitions, including both intra-task and inter-task transitions. Then each sequence of lower voltage cycles will be examined to decide whether to keep the lower voltage for these cycles.

C.1 Two voltages case

Let's start with the case where only two voltage levels, V_h and V_l , are available. We set the minimum number of intra-task transitions of a task to be 0 if there is no V_l or V_h cycle, or 1 if there is at least one V_l cycle and one V_h cycle for this task.

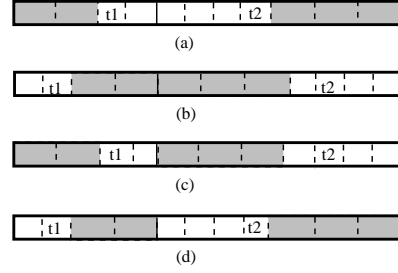


Fig. 3. Different sequences of task cycles

To minimize the number of transitions and save more energy, we also need to minimize the number of inter-task transitions. A minimum inter-task transition setting can be found by a greedy approach that keeps the same voltage across task boundaries whenever possible. For example, in Figure 3, there are two tasks, t_1 and t_2 on the same processor and t_1 is scheduled before t_2 . t_1 has 2 V_l cycles and 2 V_h cycles, while t_2 has 3 V_l cycles and 4 V_h cycles. There are four possible ways of arranging the sequences of these cycles and they are shown in Figure 3 (a) to (d). Intra-task transitions are already minimized in all four settings. Apparently, always having tasks start with V_l or V_h does not minimize the number of inter-task transitions, as shown in Figure 3 (c) and (d). Keeping the same voltage across task boundaries minimizes the number of transitions, as shown in Figure 3 (a) and (b). The greedy approach can be proved to be optimal in finding the minimum number of inter-tasks transitions. Due to space limit, the proof is omitted.

Theorem 1 *Keeping the same voltage across task boundaries whenever possible minimizes the number of inter-task transitions.*

After finding the settings with the minimum number of transitions, we check each sequence of V_l cycles. If the sequence is not long enough to offset the transition overhead involved, these cycles will be changed back to V_h and the number of transitions will decrease by up to 2. Thus the given VS solutions are changed and both the first and second term in (4) are decreased, while the total energy saving ES is increased. In the example in Figure 2, when the transition overhead is 11, the two V_l cycles of t_1 are changed back.

C.2 Multiple voltages case

The multiple voltage case can be handled in the same fashion by first minimizing intra and inter-task transitions and then eliminating unbeneficial lower voltage sequences. We formulate the problem of finding the minimum transition cost as a shortest path problem. We use one set of nodes to represent all possible settings for each task. In these settings, cycles with the same voltage are grouped together. For a task t_i with cycles on

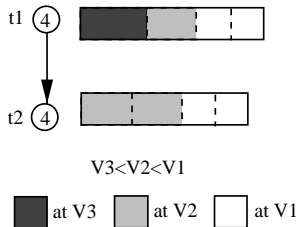


Fig. 4. An example of two tasks, t_1 has cycles at three different voltages, while t_2 has cycles at two different voltages.

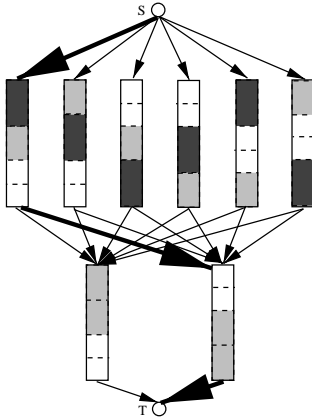


Fig. 5. A complete bipartite graph. The minimum transition overhead setting is linked by wider edges.

m_i different voltages, there are in total $m_i!$ different settings and thus total of $m_i!$ nodes in the set for this task. Even though m_i is not bounded by 2 as stated in [6], it is usually a very small integer. There are edges from each node $n_{i,j}$ in the set for t_i , where $0 < j \leq m_i!$, to every node $n_{i+1,k}$ in the set for t_{i+1} , where $0 < k \leq m_{i+1}!$. A complete bipartite graph between nodes for consecutive tasks t_i and t_{i+1} on a processor is formed in this way. The transition cost on each node is defined as the sum of the overhead of each intra-task transition in the setting. The cost of every edge in the bipartite is defined as the transition overhead between the end voltage of $n_{i,j}$ to the start voltage of node $n_{i+1,k}$. A shortest path of the graph is a setting with the minimum cost of transitions. An example of two tasks t_1 and t_2 scheduled on a processor is shown in Figure 4. t_1 has cycles running at 3 different voltages, while t_2 has cycles on two different voltages. The complete bipartite graph for the example is shown in Figure 5. The shortest path that represents the minimum transition cost is marked with wider edges.

For the continuous voltage case, each task has one voltage for all its cycles and there is no intra-task transition. A transition happens between two consecutive tasks with different voltages on the same processor. Inter-task transition is fixed if we keep the VS solutions. We need to check whether a sequence of cycles at lower voltages pro-

TABLE II
TASK PARAMETERS

set	N_t	N_c (K)	T_{cri} (μs)
s1	9	81	196
s2	50	422	444
s3	101	922	972
s4	151	1501	1848
s5	213	2988	2880
s6	245	2871	2604
s7	305	2643	2284
s8	463	4310	3636
s9	514	4633	3796
ave.	228	2263	2073

vides more saving than the transition overhead. Two tasks t_a at V_a and t_b at V_b can be treated as one sequence if the saving of running t_a at V_a and t_b at V_b is more than the overhead of the transition between V_a and V_b . Only when the saving is more than overhead, will we allow tasks to be executed with lower voltages and transitions between tasks.

One may point out that the increase of voltage will decrease a task's execution time and if the following task cannot start earlier (constrained by other tasks on other processor with a later finish time), there will be idle time on a processor. In this case, we can check the following task's other immediate predecessors to decide the start time for this task.

IV. EXPERIMENTAL RESULTS

We implemented the framework in [14] and used their VS results as a starting point. We conducted experiments on various task sets and systems. Because it is hard to get access to real-world applications, we use 9 task sets randomly generated with a software package, *Task Generation For Free (TGFF)* [13], by D. Rhodes and R. Dick, as did in [8]. The number of tasks in the 9 sets ranges from 10 to 500. Table II shows the number of tasks, number of task cycles, the critical path length of the 9 task sets. Timing constraints are set to be twice of the critical path length.

We tested systems with up to 5 different voltages and the results show that the lower voltage that is closest to V_h is used for most slow cycles. So we concentrate on systems with two voltages. We use the data for the highest and lowest voltages of the StrongARM SA-1100 processor [5] measured in [10]. Timing and energy data at $V_h = 1.65V$ and $V_l = 0.79V$ are summarized in Table III. Our testing system consists of 5 such processors that can operate at V_h and V_l . Assume the saving per V_l cycle is the same for all tasks. The overhead is $1 \mu J$ when the

TABLE III
SA-1100 PROCESSOR DATA

	Vol (V)	Fre (MHz)	CT (ns)	Power (mW)	E/cycle (nJ)
V_h	1.65	251	3.98	696.7	2.78
V_l	0.79	59	16.9	33.1	0.56

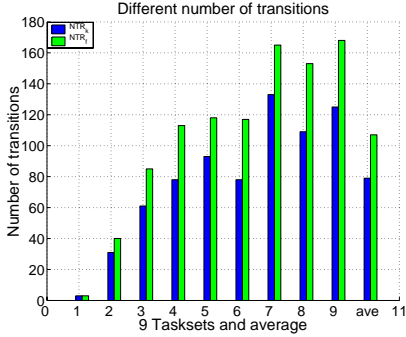


Fig. 6. Different number of transitions by a fixed ordering and our ordering that keeps same voltage across task boundaries whenever possible.

capacitor is optimized to be $5\mu f$ and a typical value of the capacitor can be $100\mu f$ which increases the overhead to $20\mu J$ per transition. Voltage range in [3] is 1.2-3.8V and thus overhead per transition in their system is higher.

Our approach keeps the same voltage across task boundaries and can avoid many transitions comparing with a policy that always lets tasks start from its highest voltage. The number of cycles at V_l is known on a given VS results and it is not changed by arranging the sequences of cycles. With the fixed policy, a task always starts from its V_h cycles if the task has cycles on V_h . The numbers of transitions by our approach and by the fixed policy for the 9 task set are shown in Figure 6. In the figure, the left bar shows the number of transitions by our approach, NTR_k , and the right bar shows the number of transitions by the fixed policy, NTR_f . By keep the same voltage across task boundaries, we can decrease the number of transitions by 27% comparing with the fixed policy. When transition energy overhead is not negligible, the decrease of number of transitions translates directly to the increase of energy saving.

Our approach is not only able to reduce the number of transitions by keeping the same voltages across task boundaries, it can also eliminate non-beneficial V_l sequences to further decrease the number of transitions and the energy consumed by transitions. To measure the effect of energy overhead on the energy saving, we change the overhead per transition to be 0, $1\mu J$ and $20\mu J$. The number of transitions by our approach decreases when the overhead per transition increases, as shown in Figure 7. In the figure, NTR_0 , NTR_1 and NTR_{20} are num-

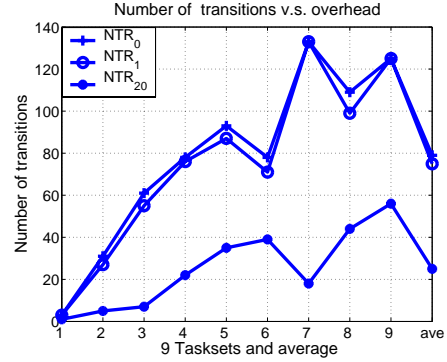


Fig. 7. Number of transitions by our EOC approach decreases when overhead per transition increases.

bers of transitions by our EOC approach when energy overhead per transition is 0, $1\mu J$ and $20\mu J$. However, the decrease of transitions still results in more energy consumption because the overhead per transition increases. The energy saving by our approach for the 9 task sets on systems with different overhead is shown in Figure 8. In the figure, sav_0 , sav_1 and sav_{20} are savings achieved after our EOC approach when energy overhead per transition is 0, $1\mu J$ and $20\mu J$. We can see when there is no overhead, the average energy saving is 25.8%, but that is decreased to 7.7% when the energy overhead per transition is increased to $20\mu J$. This tells us that transition overhead will put a limit on how much energy can be saved through varying supply voltage.

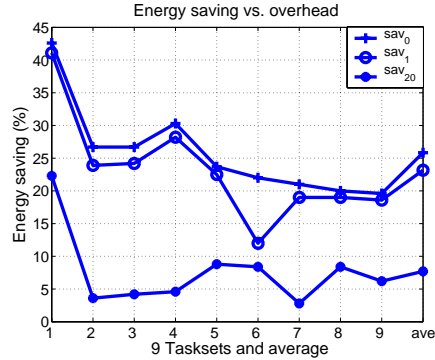


Fig. 8. Energy saving decreases when overhead per transition increases

In the following, we use the data on the system where overhead is $20\mu J$ to show that our EOC approach is very important in reducing energy consumption. In Figure 9, we show the total energy and the energy consumed by transitions for three different cases. In the first case, energy overhead is not considered, tasks always start from their highest voltage cycles and no V_l sequences are eliminated. In the second case, task cycles are orderer to have the same voltage across task boundaries whenever possible. But these is no elimination of V_l cycles. The

third case uses our EOC approach after the number of cycles for V_h and V_l are decided. Energy consumption is represented in percentage of the baseline consumption. The left bar shows the total energy E_{noeoc} and energy by transitions TRE_{noeoc} of the first case where overhead is not considered and tasks are fixed to always start from its highest voltage cycles. The center bar shows the total energy consumption E_k and the energy consumed by transitions TRE_k for the second case where task cycles are ordered to keep the same voltage across task boundaries whenever possible, but no V_l sequences are eliminated. The right bar shows the total energy E_{eoc} and energy consumed by transitions TRE_{eoc} of the third case which uses our EOC approach.

It is clear that when overhead is not considered at all, 8 out of the 9 tasks consume more energy than the baseline. The average energy consumption is 125% and transitions consume 41% of the baseline consumption. When voltage is kept the same across task boundaries whenever possible, the average energy consumption decreases to 105% and energy consumed by transitions decreases to 31% of the baseline. With our EOC approach, the average energy consumption is 92% and the energy consumed by transitions is only 8% to the baseline. The EOC approach is particularly important when overhead per transition is high. If the number of transitions is not decreased wisely, the energy consumption will increase linearly with the increase of overhead per transition and eventually becomes the dominant part and offsets all the benefit of having variable voltages. However, since our approach orders task cycles to minimize the number of transitions and eliminates unbeneficial transitions, we are able to control the energy consumed by transitions to be below 10%. Our EOC approach finishes within seconds for all tasks.

V. CONCLUSION

In this paper, we present an EOC approach which takes into account of energy overhead and improves energy saving. Our EOC approach determines the voltage for each cycle of every task for energy minimization of dependent tasks on variable voltage processors. Experimental results show that our EOC approach can reduce the number of transitions and improve energy saving.

VI. ACKNOWLEDGMENT

This research was supported in part by the National Science Foundation under Grant CCR-9988468, CCR02-08992 and MIP-9701416.

REFERENCES

[1] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on VLSI systems*, June 2000, pp. 299-316.

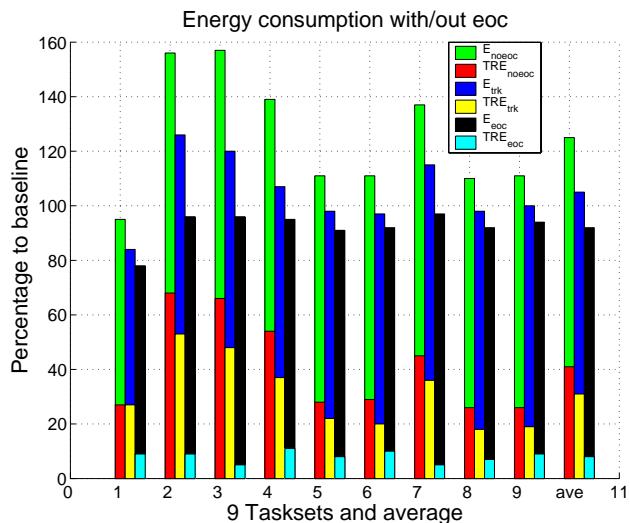


Fig. 9. Energy consumption by tasks and by transitions by our EOC approach are much lower than not considering energy overhead

- [2] T. Burd and R. Brodersen, "Design issues for dynamic voltage scaling," *ISLPED'00*, pp. 1-6.
- [3] T. Burd, "Energy-efficient processor system design," Ph.D. Dissertation, http://bwrc.eecs.berkeley.edu/Publications/2001/Theses/energ_eff_process_sys_des/index.htm.
- [4] F. Gruian, K. Kuchcinski, "LEneS: task scheduling for low-energy systems using variable supply voltage processors," *ASP-DAC'01*, pp. 449-455.
- [5] Intel StrongARM SA-1100 microprocessor developer's manual <http://developer.intel.com/design/strong/manuals/278088.htm>.
- [6] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," *ISLPED'98*, pp. 197-202.
- [7] S. Lee and T. Sakurai, "Run-time power control scheme using software feedback loop for low-power real-time applications," *ASP-DAC'00*, pp. 381-386.
- [8] J. Luo and N. Jha, "Power-conscious joint scheduling of periodic task graphs and a periodic tasks in distributed real-time embedded systems," *ICCAD'00*, pp. 357-364.
- [9] B. Mochocki, G. Quan, and X. Hu, "A realistic variable voltage scheduling model for real-time applications," *to appear in ICCAD'02*.
- [10] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," *MMSA'01*, pp. 251-259.
- [11] G. Quan and X. Hu, "Energy efficient fixed-priority scheduling for real-time systems on voltage variable processors," *DAC'01*, pp. 828-833.
- [12] M. Schmitz, B. Al-Hashimi, and P. Eles, "Energy-efficient mapping and scheduling for DVS enabled distributed embedded systems," *DATE'02*, pp. 514-521.
- [13] <http://helsinki.ee.Princeton.EDU/dickrp/tgff>
- [14] Y. Zhang, X. Hu and D. Chen, "Task scheduling and voltage selection for energy minimization," *DAC'02*, pp. 183-188.