Real-Time Task Scheduling for a Variable Voltage Processor

Takanori Okuma Tohru Ishihara Hiroto Yasuura
Department of Computer Science and Communication Engineering
Graduate School of Information Science and Electrical Engineering
Kyushu University
6–1 Kasuga-koen, Kasuga, 816-8580 Japan

Abstract

This paper presents a real-time task scheduling technique with a variable voltage processor which can vary its supply voltage dynamically. Using such a processor, running tasks with a low supply voltage leads to drastic power reduction. However, reducing the supply voltage may violate real-time constraints. In this paper, we propose a scheduling technique which simultaneously assigns both CPU time and a supply voltage to each task so as to minimize total energy consumption while satisfying all real-time constraints. Experimental results demonstrate effectiveness of the proposed technique.

1. Introduction

Advances of deep sub-micron technologies have been enabling to produce a SOC (Systems-On-a-Chip) that implements whole functionality of a system on a single chip. SOCs are embedded in various electric products such as portable information terminals, digital audio systems, automobiles, and so on. Many of these products are real-time systems which have timing constraints to be satisfied. Another important problem in SOC design is minimization of power consumption. Heat of chips due to high power consumption often prevents realization of high performance SOCs with high transistor density. In portable systems, there is a strong demand for operating the system with a small battery. Thus, design technology for high performance SOCs with low power consumption is an important research issue in the field of real-time system design.

Real-time constraints may be satisfied by employing a high performance processor core. However, the demand for low power consumption is not often satisfied simultaneously. The problem which we should address is how to realize both high-speed computation and low power consumption. In [7], Ishihara et al. have proposed a variable voltage processor which can vary its supply voltage dynamically in order to solve the problem. The variable voltage processor has the following properties.

- The processor has special instructions for controlling power. The supply voltage and the clock frequency can be changed at any time by the instructions in application programs or operating systems.
- The clock frequency is adjusted to the supply voltage, to guarantee the correct operations.

Using the variable voltage processor, tasks with severe real-time constraints can be executed with a high supply voltage (therefore, high execution speed), and tasks with loose time constraints are done with a low supply voltage. Reducing the supply voltage leads to drastic power reduction because power consumption in CMOS circuits typically increases in proportion to the square of its supply voltage.

Figure 1 shows two voltage assignments for a given program whose total execution cycles are 1000×10^6 cycles. Consider that the execution of the program has to be completed within 25 seconds. Also assume that the supply voltage, clock frequency and energy consumption per cycle are related each other as shown in the table in Figure 1. In Figure 1(A) where the processor uses the 5.0V supply voltage, the total energy consumption is 40.0J. On the other hand, in Figure 1(B), the total energy consumption is only 25.0J because the processor uses 4.0V supply voltage. We can observe about 38% energy reduction without violating the real-time constraint.

Using the variable voltage processor, the supply voltage is controlled by application programs or operating systems. Therefore, it is important to establish compiler and operating system techniques which control the supply voltage for energy minimization of real-time systems. If a system includes only a single task, it is easy to find the optimal supply voltage[3]. However, when a system contains multiple tasks, it is not a simple problem to determine an optimal

Supply Voltage	4.0V	5.0V
Clock Frequency	40MHz	50MHz
Energy/Cycle	25nJ	40nJ

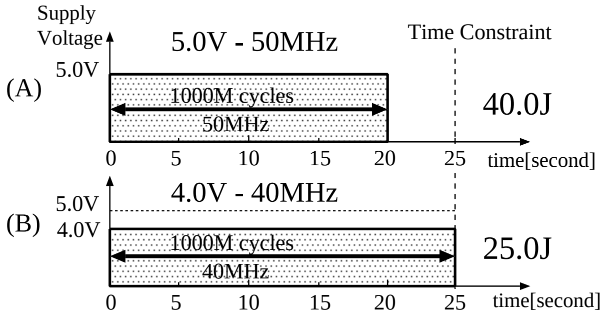

Figure 1. An example of energy reduction

supply voltage assignment which minimizes total energy consumption without violation of real-time constraints. In this paper, we propose a task scheduling technique which simultaneously assigns CPU time and a supply voltage to each task so as to minimize the total energy consumption. Our proposed technique always guarantees real-time constraints of a system if the system meets the real-time constraints under the highest supply voltage. We assume two kinds of systems: one with advance knowledge of task's arrival time, and the other without advance knowledge of it.

In [1], Hong et al. have proposed a design methodology for the low power core-based real-time SOC based on a dynamically variable voltage hardware, and presented an off-line scheduling heuristic algorithm for non-preemptive systems. Also, in [2], they have proposed an on-line scheduling algorithm for the mixed workload of both sporadic and periodic tasks on a variable voltage processor to optimize power consumption. However, the algorithm does not always guarantee the real-time constraints even if the system can meet the real-time constraints under the maximum supply voltage. Therefore, their algorithm cannot be applied to hard real-time systems. The algorithm proposed in this paper always guarantees the real-time constraints.

The rest of the paper is organized in the following way. Section 2 provides a principle of a power delay trade-off. In Section 3, a scheduling technique is proposed which simultaneously assigns CPU time and a supply voltage to each task so as to minimize the total energy consumption. In Section 4, we show experimental results demonstrate effectiveness of the proposed technique. The paper is concluded in Section 5.

2. Power-Delay Trade-off

Power consumption in CMOS circuits typically increases in proportion to the square of its supply voltage as approximated by:

$$P_{dynamic} = \sum_{k=1}^{M} CL_k \cdot Swit_k \cdot V_{DD}^2 \qquad (1)$$

where M is the number of gates in the circuit, CL_k the load capacitance of gate g_k, $Swit_k$ the switching count of g_k per second, and V_{DD} the supply voltage. From the above formula, it can be said that reduction of the supply voltage is the most effective for power reduction. In general, however, reducing the power supply voltage causes increase of circuit delay. The circuit delay can be estimated by

$$\tau \propto \frac{V_{DD}}{(V_G - V_t)^2} \qquad (2)$$

where τ is the propagation delay of CMOS transistor, V_t the threshold voltage, and V_G the voltage of input gate. (1) and (2) demonstrate the power-delay trade-off of CMOS devices. This design trade-off is one of concern for portable system designers.

Therefore, optimizing supply voltage dynamically for each application is a very important and interesting challenge.

3. The Scheduling Technique

3.1. Target Systems

A Real-time system generally consists of both application programs and an operating system which controls execution of the applications. Application programs are divided into several tasks in order to satisfy real-time constraints. A system designer has to estimate the worst case execution time of each divided task. When external events are detected, the operating system decides a schedule of these tasks so as to satisfy the real-time constraints.

In this paper, we assume a single processor system where a variable voltage processor is used as a processor core. The variable voltage processor can change its supply voltage discretely using special instructions for voltage control. This paper assume that the voltage control instructions can be used only by the operating system, and the application programs can not use the instruction. This means that supply voltage can be varied when switching tasks.

The scheduler which is a part of the operating system mainly performs the following jobs.

- The scheduler assigns a CPU time for each task on the assumption that the highest supply voltage is used. This task is called *order scheduling*.

- The scheduler assigns a supply voltage for each task. This task is called *voltage scheduling*.

3.2. Notations

A real-time task J_i is generally characterized by the following parameters.

- a_i : Arrival time.
- O_i : Worst case execution time.
- d_i : Deadline time.
- s_i : Start time of its execution.
- e_i : Completion time of its execution.
- L_i : The remaining time from completion time to deadline.

$$L_i = d_i - e_i$$

In this paper, we additionally define the following parameters to consider energy consumption of the task.

- X_i : Worst case execution cycles.
- F_i : Clock frequency.

  O_i, X_i and F_i have the following relation.

$$O_i = \frac{X_i}{F_i}$$

- V_i : Supply voltage.
- C_i : Average load capacitance.

$$C_i = \sum_{k=1}^{G} CL_k \cdot Swit_k$$

  where G is the number of gates in the processor, CL_k the load capacitance of a gate g_k, $Swit_k$ the average switching count of g_k per cycle.

- E_i : Worst case energy consumption.

  E_i, C_i, X_i, and V_i have the following relation.

$$E_i = C_i \cdot X_i \cdot {V_i}^2$$

Some of the parameters defined above are illustrated in Figure 2.

We assume that F_i and V_i are not changed during execution of J_i. However, when J_i is preempted by another task and is resumed after the preemption, different supply voltage and clock frequency can be used from ones used before the preemption.

The variable voltage processor is characterized by the following parameters.

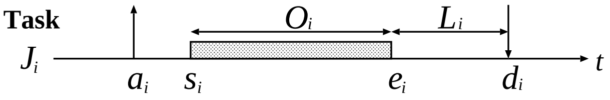

Figure 2. Typical parameters of a real-time task

- (v_j, f_j) : Processor mode. When the supply voltage of the processor is v_j, the clock frequency of the processor is f_j.
- $m = |\{(v_j, f_j)\}|$: The number of processor mode.
- $V_{max} = \max(v_j)$: The largest supply voltage.

For example, when task J_3 is executed with the processor mode 2 ($j = 2$), V_3 and F_3 are defined as follows.

$$V_3 = v_2, \quad F_3 = f_2$$

3.3. Static Voltage Scheduling

The static voltage scheduling assumes that a task set is statically order-scheduled in advance. Several order-scheduling algorithms such as Earliest Deadline First[5, 9], Bratley's algorithm[6], and Latest Deadline First[4] have been proposed in the past. All of these algorithms find a feasible schedule if exists. In the static voltage scheduling technique, the CPU time is assigned to tasks using these algorithms on the assumption that the highest supply voltage is used. Then, the supply voltage is assigned to each task to minimize the total energy consumption. If the task set is order-scheduled successfully, the voltage-scheduler can assign a supply voltage to each task without violation of real-time constraints.

In this section, we formulate a static voltage scheduling problem for the statically order-scheduled task set.

3.3.1 Preliminaries

To simplify the problem, we divide the task set based on the following rules.

Modification rule 1

If there is a time period during which no task is executed (See Figure 3), the period is called an *idle time*. The task set is divided into several subsets by idle times. That is, if there is an idle time between two tasks J_i and J_{i+1}, they are classified into different subsets. Then, J_i is executed at the end of the subset including J_i, and J_{i+1} is executed at the beginning of the subset including J_{i+1}. If there is a task J_k

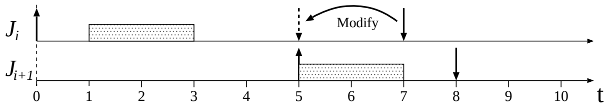

Figure 3. An example of modification rule 1

belonging to the same subset as J_i such that $d_k > s_{i+1}$, the deadline d_k is modified as follows:

$$d_k = s_{i+1} (= a_{i+1})$$

Modification rule 2

A task preempted by another task is divided into several subtasks (See Figure 4). These subtasks are treated as different tasks. If task J_i is divided into n tasks, $J_{i.1}, \ldots, J_{i.n}$, the arrival time and deadline of task $J_{i.j}$ is defined as follows:

$$\begin{aligned} a_{i.1} &= a_i & d_{i.n} &= d_i \\ a_{i.j} &= e_{i.j-1} & d_{i.j} &= s_{i.j+1} \end{aligned}$$

Moreover, the worst case execution cycles $X_{i.j}$ is given by

$$X_{i.j} = \frac{e_{i.j} - s_{i.j}}{\displaystyle\sum_{k=1}^{n} (e_{i.k} - s_{i.k})} \cdot X_i$$

3.3.2 Formulation of Static Voltage Scheduling

A static voltage scheduling problem is formulated as follows:

Given

- the task set $\{J_1, \ldots, J_n\}$ modified by the rules mentioned above,
- X_i, d_i, and C_i for each tasks, and
- processor mode $\{(v_j, f_j) \mid j = 1, \ldots, m\}$,

find the function $\sigma : \mathbf{N} \to \mathbf{M}$ which minimizes

$$E = \sum_{i=1}^{n} C_i \cdot X_i \cdot {V_i}^2 \qquad (3)$$

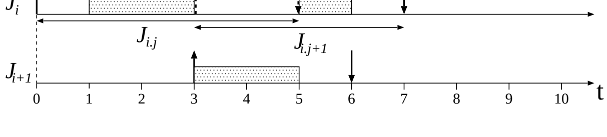

Figure 4. An example of modification rule 2

subject to

$$\forall i = 1, \ldots, n \quad \sum_{k=1}^{i} \frac{X_k}{F_k} \leq d_i - a_1 \qquad (4)$$

where σ, $\mathbf{N}$ and $\mathbf{M}$ are defined as follows:

$$\sigma(i) = j \Rightarrow (V_i, F_i) = (v_j, f_j)$$

$$\mathbf{N} = \{1, \ldots, n\}, \quad \mathbf{M} = \{1, \ldots, m\}$$

By solving the above problem, we can obtain the optimal supply voltage assignment on static voltage scheduling.

3.4. Dynamic Voltage Scheduling

When a scheduler assigns supply voltage dynamically, the scheduling time should be short. In our dynamic voltage scheduling technique, we assume that the scheduler assigns a supply voltage to only one task which is going to run next. Then, the scheduler has to assign supply voltage to it in order for tasks which will have been executed in the future to satisfy real-time constraints.

We define an *occupation period* as the maximum value of period that the next executed task can use without violation of real-time constraints for the future tasks. That is, if the next task is assigned supply voltage so as to finish within the occupation period, the real-time constraints are always guaranteed.

The length of the occupation period is dynamically calculated. The scheduler has to determine the value for the next task. We present two algorithms, called the SD algorithm and the DD algorithm, which determine the length of occupation periods. The SD algorithm assumes that arrival time of all task is known. On the other hand, the DD algorithm assumes that arrival time of all task is unknown. The pseudo codes of the SD algorithm and the DD algorithm are shown in Figure 5 and in Figure 6, respectively.

In the algorithms, the length of the occupation period is calculated by determination of the start point and the end point. In the SD and DD algorithm, the start point is determined as the actual finishing time of the current task or the arrival time of the next task (if it does not yet arrive). In the DD algorithm, the end point is determined as the finishing time of the next task on the assumption that all tasks are assigned the maximum supply voltage and complete at the worst case execution cycle. In the SD algorithm, the end point is determined as the time which is added the minimum remaining time in not yet executed tasks to the end point in the DD algorithm. The SD and DD are optimal algorithms which maximize the length of the occupation period. These proofs have been omitted due to space limitation.

SD Algorithm:

- Input:
  - Task set $\{J_1, \ldots, J_n\}$ which modified by modification rule 2 in Section 3.3.1.
  - Processor mode $\{(v_j, f_j) \mid j = 1, \ldots, m\}$
- Output:
  - The occupation period for J_k
- Algorithm:
  - When J_k starts its execution at time t, the occupation period for J_k is as follows:

$$s_k|_{V_{max}} - t + O_k|_{V_{max}} + \min_{i \geq k}(L_i|_{V_{max}})$$

Figure 5. pseudo code of SD Algorithm

DD Algorithm:

- Initial values:
  - $\mathcal{R} := \emptyset$
  - $t_s := 0$
  - $J_{exe} := J_{idle}$
- Input:
  - Processor mode $\{(v_j, f_j) \mid j = 1, \ldots, m\}$
  - Current execution task J_{exe}
  - Supply voltage V_{exe} which was assigned to J_{exe}
  - Current time t
- Output:
  - J_{next}: Next execution task
  - T_{next}: Occupation period for J_{next}
- Algorithm:

```
if new task arrived then
    $J_{arrive}$ := arrival task
    if Priority($J_{arrive}$) > Priority($J_{exe}$) then
        $\mathcal{R} := \{J_{exe}\} \cup \mathcal{R}$
        if $J_{exe} \neq J_{idle}$ then
            $X_{exe}$ := the rest of $X_{exe}$
        end if
        $t_s := t + O_{arrive}|_{V_{max}}$
        $J_{next} := J_{arrive}$
        $T_{next} := t_s - t$
    else
        $\mathcal{R} := \{J_{arrive}\} \cup \mathcal{R}$
        $J_{next} := J_{exe}$
        Supply voltage is unchnaged
    end if
else if $J_{exe}$ finished then
    $J_{high}$ := the task with maximum priority in $\mathcal{R}$
    $\mathcal{R} := \mathcal{R} - \{J_{high}\}$
    $t_s := t_s + O_{high}|_{V_{max}}$
    $J_{next} := J_{high}$
    $T_{next} := t_s - t$
end if
```

Figure 6. pseudo code of DD Algorithm

4. Experimental Results

We assume a preemptive real-time system where five tasks described in Table 2 are executed on a variable voltage processor. The modes of the processor are shown in Table 1. Two scenarios of task execution are assumed as shown in Table 3 and Table 4. The deadline of each task in the scenario 1 is earlier than that in the scenario 2.

The order and the voltage of tasks are scheduled using the following methods.

Normal: Assign the maximum supply voltage to all tasks.

SS: Statically assign CPU time and supply voltage to each task.

SD: Dynamically assign supply voltage to each task by SD algorithm after statically order scheduling.

DD: Dynamically assign CPU time and supply voltage to each task by DD algorithm.

Note that a_i and d_i are assumed to be known in advance for SS and SD, and unknown for DD. Using any scheduling method, tasks are order-scheduled in the following order:

$$J_1 \rightarrow J_2 \rightarrow J_3 \rightarrow J_4 \rightarrow J_3 \rightarrow J_5$$

where J_3 is preempted by J_4.

Table 5 and Table 6 show the energy estimation results for the scenario 1 and the scenario 2, respectively. In the scenario 1, the energy reduction rate of SS, SD, and DD are 27%, 38%, and 16%, respectively, compared with Normal. In the scenario 2, the energy reduction rate of SS, SD, and DD are 56%, 58%, and 16%, respectively.

From the experimental results, we observe the following facts.

- SS, SD, and DD always give better results than Normal.

Table 1. Processor mode

Supply Voltage	Clock Frequency
5.0V	50MHz
4.0V	40MHz
2.5V	25MHz

Table 2. Task set

Task	X_i[cycle]	C_i[F]
J_1	10000000	10
J_2	8000000	20
J_3	15000000	10
J_4	5000000	10
J_5	4000000	30

Table 3. Scenario 1

Task	a_i[sec]	d_i[sec]	actual execution cycles
J_1	0	0.2	9300000
J_2	0	0.4	7000000
J_3	0	0.8	14000000
J_4	0.4	0.7	3000000
J_5	0.5	0.9	3000000

Table 4. Scenario 2

Task	a_i[sec]	d_i[sec]	actual execution cycles
J_1	0	0.5	9300000
J_2	0	0.7	7000000
J_3	0	1.4	14000000
J_4	0.4	1.0	3000000
J_5	0.5	1.5	3000000

Table 5. Results for scenario 1

Task	Normal	SS	SD	DD
J_1	5.0V	5.0V	5.0V	5.0V
J_2	5.0V	5.0V	4.0V	5.0V
J_3	5.0V	4.0V	2.5V	5.0V
J_4	5.0V	5.0V	5.0V	5.0V
J_3	5.0V	5.0V	5.0V	4.0V
J_5	5.0V	2.5V	2.5V	4.0V
Energy	1615J	1176J	999J	1357J

Table 6. Results for scenario 2

Task	Normal	SS	SD	DD
J_1	5.0V	4.0V	4.0V	5.0V
J_2	5.0V	4.0V	4.0V	5.0V
J_3	5.0V	5.0V	4.0V	5.0V
J_4	5.0V	2.5V	2.5V	5.0V
J_3	5.0V	2.5V	2.5V	4.0V
J_5	5.0V	2.5V	2.5V	4.0V
Energy	1615J	702J	685J	1357J

- The supply voltage assigned by SS is proved to be optimal when execution cycles of all tasks are the worst case ones. However, the voltage is not optimal because the SS scheduler can not utilize the remaining time due to early completion of tasks.

- In SS and SD, looser deadline constraints lead to greater energy reduction rates. This is because looser deadline means increased remaining time which can be utilized by the scheduler for reducing the supply voltage.

- In contrast to SS and SD, power consumption using the DD scheduler is independent of the deadline constraints.

5. Conclusion

We have proposed a scheduling technique which simultaneously assigns CPU time and a supply voltage to each task so as to satisfy its real-time constraints and to minimize total energy consumption. The total energy consumption can be dramatically reduced by dynamic assignment of the supply voltage. Although our proposed algorithm does not always give the optimal supply voltage assignment to minimize the total energy consumption, it always guarantees real-time constraints. Our future work will be devoted to extend the proposed algorithm so as to assign an optimal supply voltage.

Acknowledgment

Semiconductor Technology Academic Research Center (STARC) grant No. PJ-No.987 supported this research. We are grateful for their support. We would like to express our thanks to Dr. Hiroyuki Tomiyama of University of California Irvine for giving valuable comments to this work.

References

[1] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. Srivastava. “Power Optimization of Variable Voltage Core-based Systems”. In *Design Automation Conference*, 1998.

[2] I. Hong, M. Potkonjak, and M. B. Srivastava. “On-Line Scheduling of Hard Real-Time Tasks on Variable Voltage Processor”. In *Proc. of ICCAD-98*, pages 653–656, 1998.

[3] T. Ishihara and H. Yasuura. “Voltage Scheduling Problem for Dynamically Variable Voltage Processors”. In *Proc. of International Symposium on Low Power Electronics and Design(ISPLED’98)*, pages 197–202, August 1998.

[4] E. Lawler. “Optimal Sequencing of a Single Machine Subject to Precedence Constraints”. Managements Science, 19, 1973.

[5] C. L. Liu and J. Layland. “Scheduling Algorithms for Multiprogramming in a Hard Real-time Environment”. *Journal of the ACM*, 20(1):46–61, 1973.

[6] P. Bratley, M. Florian, and P. Robillard. “Scheduling with Earliest Start and Due Date Constraints”. Naval Research Logistics Quarterly, 18(4), 1971.

[7] T. Ishihara and H. Yasuura. “Optimization of Supply Voltage Assignment for Power Reduction on Processor-Based Systems”. In *Proc. of SASIMI ’97*, pages 51–58, 1997.

[8] T. Ishihara and H. Yasuura. “Programmable Power Management Architecture for Power Reduction”. *IEICE Transaction on Electronics*, E81-C(9), September 1998.

[9] W. Horn. “Some Simple Scheduling Algorithms”. *Naval Research Logistics Quarterly*, 21, 1974.