Application-Driven Low-Power Techniques Using Dynamic Voltage Scaling

Author
School of Electrical Engineering and Computer Science
Seoul National University, Seoul, KOREA
id AT snu.ac.kr

Abstract

It is generally accepted that dynamic voltage scaling (DVS) is one of the most effective techniques of energy minimization for real-time applications. The effectiveness comes from the fact that the amount of energy consumption is quadractically proportional to the voltage applied to the processor.

1 Introduction

Over the past decades there have been enormous efforts to minimize the energy consumption of CMOS circuit systems. Dynamic voltage scaling (DVS), involving dynamic adjustments of the supply voltage and the corresponding operating clock frequency, has emerged as one of the most effective energy minimization techniques. A one to one correspondence between the supply voltage and the clock frequency in CMOS circuits imposes an inherent constraint to DVS techniques to ensure that voltage adjustments do not violate the target system's timing constraints.

Many previous works have focused on hard real-time systems with multiple tasks. Their primary concern is to assign a proper operating voltage to each task while satisfying the task's timing constraint. In these techniques, determination of the voltage is carried out on a task-by-task basis and the voltage assigned to the task is unchanged during the whole execution of the task, which is refered to as *inter-task voltage scheduling*. Yao *et al.* [1] proposed an optimal inter-task voltage scheduling algorithm for independent tasks in which a task is characterized by its arrival time, deadline and required CPU cycles. The proposed scheduling technique computes the speed of execution at any given time (and thus automatically determines each task's starting and ending times) so that the total energy consumption is minimized. Although they formulated the problem without the constraint that the task should be assigned to a single operating voltage, by the convexity of the power function each task is given only one 'middle' voltage that is proved to be optimal.

2 Intra-task DVS techniques

The amount of energy dissipation for the execution of a task is

$$E \propto {V_{DD}}^2 \times N_{tot} \tag{1}$$

where N_{tot} is the total number of instruction cycles executed for a task. Thus, the *intra-task voltage scheduling* problem is to assign a proper voltage to each basic block of the task so that the energy consumption in *Eq.(1)* is minimized.

The relationship between clock frequency and voltage in CMOS circuits is

$$f_{CLK} \propto (V_{DD} - V_T)^{\alpha}/V_{DD}. \tag{2}$$

where V_T is the threshold voltage and α is the velocity saturation index. If the value of V_T is small enough, the expression is reduced to $f_{CLK} \propto {V_{DD}}^{\alpha-1}$.

Since the clock frequency determines the voltage, the scheduling problem can be stated as:

(Intra-Task DVS Problem) *The intra-task voltage scheduling problem for a task's CFG is to determine the clock frequency for each node (i.e., basic block) of the CFG so that the total energy by the task is minimized while satisfying the timing constraint of the task.*

The key problem to solve is to determine what clock frequency should be set to the entry point of each basic block so that the overall energy consumption of the task is minimized. Existing intra-task DVS techniques can be classified according to the way of determining the lowest clock frequency to be set at the entry point of each basic block.

1. (*RWCEP based DVS*): This techique [2] uses the lowest clock frequency by which the remaining WCEP (worst case execution path) can be completed within the deadline of the task.
2. (*RACEP based DVS*): This techique [4] uses the lowest clock frequency by which the remaining ACEP (average case execution path) can be completed within the deadline of the task.

3. (*ROCEP based DVS*): This techique [3] uses the lowest clock frequency by which the remaining OCEP (energy-optimal case execution path) can be completed within the deadline of the task.

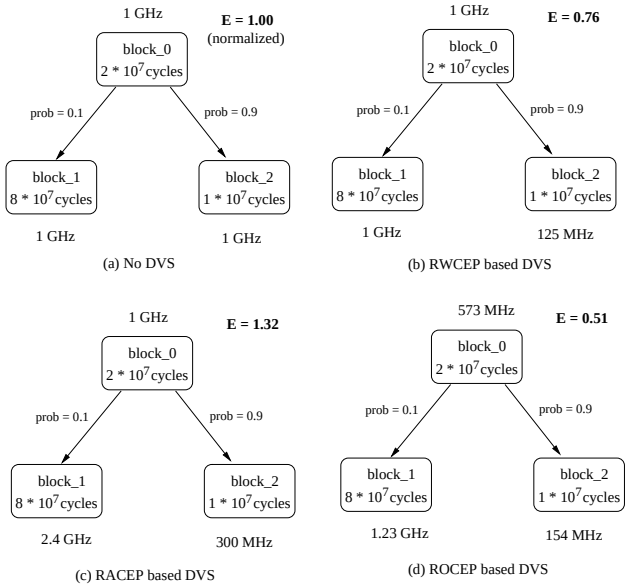

Figure 1: **Example showing the calculations of clock frequency in basic block_0.**

For example, consider a simple hard real-time task with deadline of 100 ms and three basic blocks. Its control flow graph is shown in Figure 1(a) where the number inside each node indicates the number of execution cycles of the block and the number assigned to each arc indicates the probability that the control flow follows the edge. If DVS technique is not used, the speed for the task should be set tightly to (length of the critical path) / (remaining time to deadline) = $[(2+8)10^7 cycles]/(100 \cdot 10^{-3}s) = 1GHz$. (See Figure 1(a).) If DVS technique follows the worst case execution path, the speed of block_2 will be set to (length of the (RWCEP from block_2) / (remaining time to deadline) = $[1 \cdot 10^7 cycles]/(80 \cdot 10^{-3}s) = 125MHz$. (See Figure 1(b).) On the other hand, if DVS technique follows the average case execution path, the speed is set to (length of the (RACEP from block_0) / (remaining time to deadline) = $[(2+1) \cdot 10^7 cycles]/(100 \cdot 10^{-3}s) = 300MHz$. (See Figure 1(c).) Finally, if DVS follows energy-optimal execution path, the speed to be set in block_0 is calculated based on the probabilities of its succeeding basic blocks. Here, the speed is 573 MHz. See Figure 1(d).) In summary, we can see that The ROCEP based DVS technique outperforms the others because the clock speed used at each basic block alwyas leads to the total energy consumption which is optimal on the average.

In Figure. 2(b), the thick, dotted, and regular arrows respectively indicate the increase, decrease and no change of the processor speed, and the basic blocks with changed operating speed are marked with gray-color.

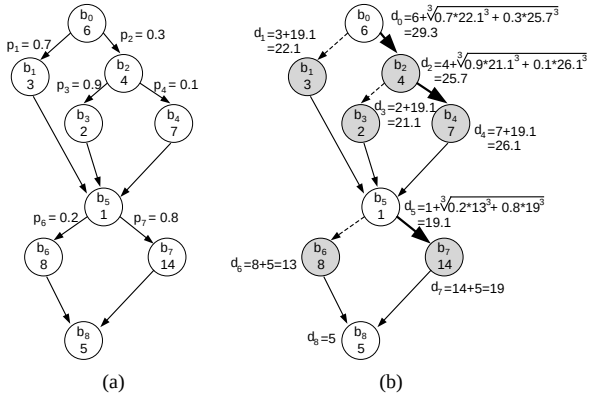

Figure 2: **(a) CFG of a task τ_{simple}; (b) Calculation of δ values.**

3 Conclusions

In this paper, we described the current status of the research works on dynamic voltage scaling (DVS) in view of the optimiality of energy minimization.

References

[1] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," *Proc. of IEEE Symposium on Foundations of Computer Science*, 1995.

[2] D. Shin, J. Kim, and S. Lee, "Intra-task voltage scheduling for low-energy hard real-time applications, *IEEE Design and Test of Computers*, Vol. 18, 2001.

[3] J. Seo, T. Kim, and K. Chung, "Profile-based optimal intra-task voltage scheduling for hard real-time applications," *Proc. of Design Automation Conference*, 2004.

[4] D. Shin and J. Kim, "A Profile-Based Energy-Efficient Intra-Task Voltage Scheduling Algorithm for Hard Real-Time Applications", *Proc. of International Symposium on Low-Power Electronics abd Design*, 2001