An Extenics-based Dynamic Voltage Scaling for Real-Time Systems

Liang-Teh Lee [1], Chih-Chieh Tang [2]

Dept. of Computer Science and Engineering, Tatung University, Taiwan
No.40, Sec. 3, Zhongshan N. Rd., Taipei City 104, Taiwan (R.O.C.)
[1] `ltlee@ttu.edu.tw`
[2] `g9606031@ms.ttu.edu.tw`

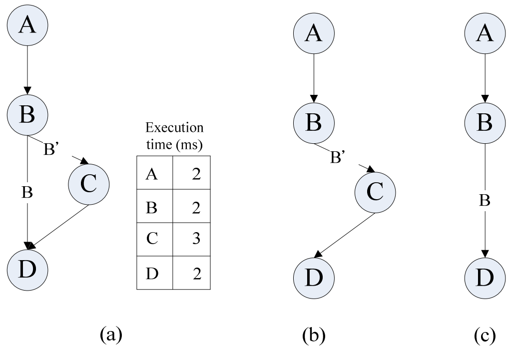
***Abstract*—With the growing of applying the embedded system technology to the mobile systems, energy efficiency is becoming an important issue for designing a real-time embedded system. One of the possible techniques to reduce energy consumption is the Dynamic Voltage Scaling (DVS). In recent years, many researchers have proposed several DVS-based algorithms for applying to hard real-time systems. However, those methods are all based on the static scheduling. In this paper, a dynamic scheduling scheme with the Excenics-based Dynamic Voltage Scaling (E-DVS) mechanism has been proposed. In the proposed method, an extenics relational function is derived to calculate and predict the execution time and the system supply voltage. The experimental results show that the proposed method can effectively decrease the energy consumption about 78.81 percent.**

***Keywords*—Real-time System, Dynamic Voltage Scaling, Extenics, Embedded System**

1. INTRODUCTION

With the growing of applying the embedded system technology to the mobile system, energy efficiency is becoming an important issue for the real-time embedded system. If the energy consumption of the mobile system can be reduced, the mobile devices will maintain more time for continuous operation. One of the possible techniques to reduce energy consumption is the Dynamic Voltage Scaling (DVS) [1]. DVS can adjust the supply voltage during runtime. DVS utilizes the slack time, i.e. the remaining time that the processor completing a job before its deadline, to reduce the energy expense.

In recent years, many researchers have proposed several DVS-based algorithms for applying to hard real-time systems [3-8]. Lin et al. proposed an algorithm that is Genetic Algorithm for the DVS scheduling [4]. Shao et al. focused on real-time loop-scheduling technique via DVS [6]. Wu et al. had presented an energy-efficient scheduling, which was utilizing and processing the schedule table before running and mapping each task to deferent processors by Genetic Algorithm to reduce the energy expense effectively [7]. However, those methods are all based on the static scheduling. Our aim is to find out a new dynamic scheduling algorithm for solving the problem in a task set with branches to reduce the energy expense.



Execution time (ms)	
A	2
B	2
C	3
D	2

(a) (b) (c)

Fig. 1 (a) A task set with branches (b) Track 1 (c) Track 2

Fig.1(a) shows an example of the relation of a task set. In this example, task B has to decide the system executing track. According to the relationship among tasks shown in Fig.1(a), there are two possible tracks which are illustrated in Fig.1(b) and Fig.1(c). If Task B is executed and the condition value B' is obtained, the system will execute Task C before Task D, otherwise the system will execute Task D directly. Note that, for simplicity, the communication costs between each task are assumed to be 0 in this paper.

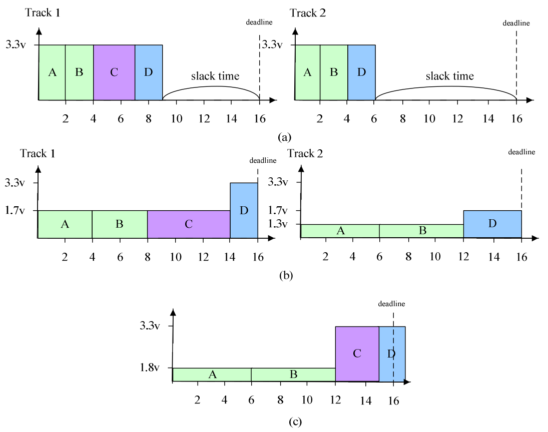


Fig. 2 (a) Schedule of task set in Fig.1a without DVS (b) Schedule of task set in Fig.1a with DVS (c) Improper scaling

Fig.2 is the schedule of the task set based on the Fig.1(a). Fig.2(a) shows the schedule without applying DVS. In track 1, the wasted slack time is 7ms, and there is 10ms wasted in track 2. The system can utilize those slack times to reduce energy consumption. The schedule with DVS is shown in Fig.2(b). Since the condition value is unknown before executing the task B, the system can not estimate which track will be executed in advance. If the system executes task A and B by using the lower voltage, it may be conflicted, the system can not meet the hard real-time system requirement as shown in Fig.2(c).

In this paper, a novel real-time scheduling technique is proposed to decrease energy consumption via Excenics [2] by applying DVS. In the proposed Excenics-based Dynamic Voltage Scaling (E-DVS), the system is allowed to add new tasks to the ready list during runtime. In addition, comparing with the traditional scheduling algorithms, the proposed method can decrease the energy consumption significantly.

The rest of the paper is organized as follows. In Section 2, the background about our researches is introduced briefly. The proposed algorithm will be presented in Section 3. We give some simulation results in Section 4, and the conclusion is made in Section 5.

2. BACKGROUND

In this section, we will describe several preliminaries in our model. At the first, we will define some power consumption equations in for estimating the energy consumption. Then we will describe some concepts about Excenics. In this section, we will describe several preliminaries in our model. At the first, we will define some power consumption equations in for estimating the energy consumption. Then we will describe some concepts about Excenics.

In a single core embedded system which has several voltage levels, the equation of power consumption can be defined as $P = V^2/Z$, where $Z = 1/(2\pi Cf)$. So the power consumption can be simplified to:

$$P = k \cdot f \cdot V^2 \tag{1}$$

Let ***k*** be a constant in formula (1), then we can find a relation between the frequency f and the supply voltage ***V*** as:

$$f = K \cdot (V - V_t)^2 / V \tag{2}$$

Based on the equations (1) and (2), we can find a relation equation between power consumption and the supply voltage. This equation is shown as follows:

$$P = C \cdot K \cdot (V - V_t)^2 \cdot V \tag{3}$$

In equation (3), ***C*** is the capacitance, ***K*** is a constant, ***V*** is used to represent the current supply voltage, and $\boldsymbol{V_t}$ means the threshold voltage in the system. Let $\boldsymbol{E_V}$ be the processing time, if the supply voltage is ***V***, then we can calculate $\boldsymbol{E_V}$ as:

$$E_V = \frac{(V_{\max} - V_t)^2}{V_{\max}} \cdot \frac{V}{(V - V_t)^2} \cdot E_{\max} \tag{4}$$

In formula (4), $\boldsymbol{V_{max}}$ is the highest supply voltage in the system and $\boldsymbol{E_{max}}$ is the execution time with the supply voltage $\boldsymbol{V_{max}}$. Thus, the energy consumption can be calculated by multiplying equation (3) and equation (4).

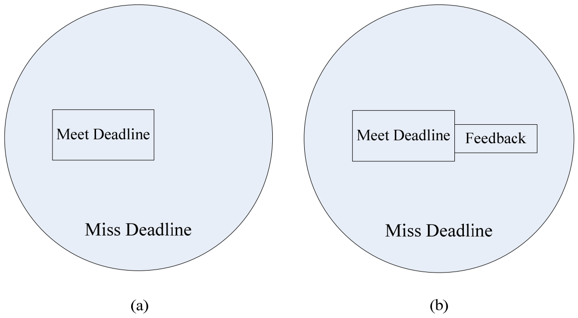


Fig. 3 (a) Traditional meet deadline, miss deadline state spaces (b) Excenics meet deadline, miss deadline, and feedback state spaces

The Excenics theory had been proposed by Cai in 1983 [2] to solve the incompatible problem through the systematic transformation. Traditionally, in a real-time system, the status of a task can be classified as meet deadline or miss deadline, as shown in Fig.3(a). By applying the

Excenics theory, parts of the task set in the miss deadline status maybe transferred to the task set of meet deadline status by a feedback algorithm. The extenics relational function is defined to the relationship between an element and a set. In our case, the range of the relationship is (-∞, +∞), we define an extenics relational function by ***K(V)* ∈ *(-∞, +∞)*** for any supply voltage ***V***.

- ***K(V) < -1***, the task will miss its deadline if it is executed in supply voltage ***V***.
- ***0 > K(V) > -1***, the task will meet its deadline if applying the feedback algorithm and it is executed in supply voltage ***V***.
- ***K(V) ≧ 0***, the task will meet its deadline, if it is executed in supply voltage ***V***.
- ***K(V) = 0***, it is called zero point, i.e., the task completes its execution just at its deadline when it is executed in supply voltage ***V***. At this point, either the supply voltage can just support the task to meet its deadline or this task is executed with the highest voltage in the proposed system.

3. EXCENICS-BASED DYNAMIC VOLTAGE SCALING ALGORITHM

In this paper, we propose an extenics-based dynamic voltage scaling (E-DVS), which will be described in this section. The extenics relational function will be shown at first, and then, the E-DVS algorithm will be presented. Finally, we will describe the feedback algorithm in the proposed method.

3.1. Extenics Relational Function

In a hard real-time system, each task must be completed before its deadline. Let ***X*** be the time domain of each task and it is between the task release time T_s and its deadline T_d. Let X_0 be the time between the minimum execution time, T_s plus the execution time in the highest supply voltage E_{max}, and the new completion time T_c. E_w is assumed to be the task execution time with the maximum supply voltage of the system. Thus, a new completion time T_c can be calculated by the following equation:

$$T_c = \frac{E_{\max}}{E_W}\left(T_d - T_S\right) + T_S \qquad (5)$$

Based on the equation (4), we can calculate the completion time T_V with supply voltage ***V*** by the following equation:

$$T_V = \frac{\left(V_{\max} - V_t\right)^2}{V_{\max}} \cdot \frac{V}{\left(V - V_t\right)^2} \cdot E_{\max} + T_S \qquad (6)$$

According to the extenics theory, the distance can be calculated between the execution time E_v and X_0 or ***X***. The equation is shown below:

$$\rho\left(T_v, X_0\right) = \left|T_v - \frac{T_s + E_{\max} + T_c}{2}\right| - \frac{1}{2}\left(T_c - T_s - E_{\max}\right) \qquad (7)$$

$$\rho\left(T_v, X\right) = \left|T_v - \frac{T_s + T_d}{2}\right| - \frac{1}{2}\left(T_d - T_s\right) \qquad (8)$$

From equations (7) and (8), the extenics relational function can be expressed as:

$$K\left(T_v\right) = \begin{cases} \dfrac{\rho\left(T_v, X_0\right)}{\rho\left(T_v, X\right) - \rho\left(T_v, X_0\right)} & T_v \notin X_0 \\ -\rho\left(T_v, X_0\right) & T_v \in X_0 \end{cases} \qquad (9)$$

```
01. For all of tasks $T_i$, i = 1... n
    {
02  If task $T_i$ is the first task in the system, a new
    created task, or its parents are branch task.
        {
03.             Re-allocate deadline $T_c$ of each
                task by considering the worst
                case.
04.             $T_{total\ slack} = 0$
        }
05.     For each voltage level $V_k$, k = 1 …
        max-1 {
06.             Calculate K($V_k$).
07.             Find a supply voltage v such
                that min(K($V_k$) ≧ 0).
        }
08.     If v does not exists {
09.             $T_c = T_c + T_{total\ slack}$
10.             Calculate K($V_{max}$).
11.             If K($V_{max}$) < 0
                        miss deadline and
                        delete task
12.             Else
                set v to $V_{max}$, set $T_{total\ slack}$ to 0
        }
13.     If $v_i$ ≧ $V_{max}$ {
14.             Change system voltage to $V_{max}$
        }
15.     Else {
16.             Change system voltage to v
        }
17.     $T_{total\ slack} = T_{total\ slack} + T_{slack}$
}
```

Fig. 4 Excenics-based Dynamic Voltage Scaling Algorithm

3.2. E-DVS algorithm

For explaining the E-DVE algorithm, T_{slack} is defined to be a slack time when the system is executed with supply voltage ***V*** and $T_{total\ slack}$ is the sum of all slack times**.** The E-DVS algorithm is described in Fig.4. In Steps 02 to 04, if task T_i

is the first task in the system, a new created task, or its parents are branch tasks, the new completion time T_c of the task will be calculated by considering the worst case. Steps 05 to 07, the relational value of each supply voltage level is calculated by equation (9), the extenics relational function $K(V)$. Furthermore, in Step 7, it is trying to find out a supply voltage level V with the minimum positive relational value for all voltage levels except the highest voltage level. Steps 08 to 12, if V does not exist, the proposed feedback algorithm will be performed to adjust the supply voltage for the task to meet its deadline. The feedback algorithm will be described in section 3.3. From Steps 13 to 16, the system will change the voltage to the new supply voltage V for executing this task.

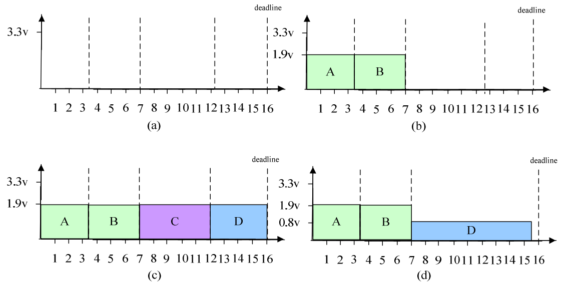


Fig. 5 The schedule of task set in Fig.1a with E-DVS

The example illustrated in Fig.1(a) will be used to explain the proposed algorithm. In the example, task A is a start task and task B is a branch task, which will decide the track to be executed according to the conditional value. Initially, the worst case completion times of task A, B, C, and D are calculated, as shown in Fig.5(a). For example, T_c of task A is 3.556*ms*. The relational value of task A with each supply voltage is calculated by applying equation (9), and find out a supply voltage is 1.9V which is the minimum positive relational value. Similarly, we can get the supply voltage, 1.9V, for task B, as shown in Fig.5(b). When task B is completed, the conational value will be generated and the system will decide which track will be executed. At this moment, the proposed method will recalculate the new completion times of remaining tasks and decide the system supply voltage. According to the conational value, different branch will be executed. Fig.5(c) shows the schedule and supply voltage used if B' branch is selected, while Fig.5(d) shows the situation if B branch is selected.

3.3 Feedback Algorithm

Since the mobile device can not supply required voltage levels for DVS, it is difficult to provide a voltage for a task to meet its deadline with zero slack time. In the proposed E-DVS mechanism, if the task is the first task in the system, a new created task, or its parents are branch tasks, the system will recalculate the new completion times of tasks in the ready list. According to the proposed feedback algorithm, the system collects the slack time of each task until the next time to recalculate new completion times. Steps 8 to 12 in Fig.(4) is to perform the feedback algorithm. From Steps 08 to 12, if V does not exist, the system will add $T_{total\ slack}$ to T_c and the new relational value with supply voltage level V_{max} will be recalculated by equation (9). If $K(V_{max}) \geqq 0$, the task will meet its deadline with the maximum supply voltage V_{max}. The system will change the supply voltage to V_{max} and set $T_{total\ slack}$ to 0.

4. SIMULATION RESULTS

In this section, the simulation of the proposed algorithm will be described. We generate eight different task sets with the number of tasks 5, 10, 15, 20, 25, 30, 35, and 40, respectively. Each set will generate ten thousand relations between tasks in each task set and the execution time of each task is set between 1*ms* and 5*ms*. There are two different values for system deadline in the simulation, one is fixed and the other is changeable according to the number of tasks. The system deadline is set to the worst case execution time. For example, in the task set with 5 tasks, since the worst case execution time of a task is 5ms, the system deadline is set to 25*ms*. Energy consumption in executing different task sets will be compared in two different simulations. The first simulation, the system deadline of each task set is set according to the number of tasks. In the second simulation, the system deadline is fixed to 200*ms*.

Table 1 shows the simulation results. We can see that the number of tasks is increased, the system energy consumption will also be increased. In the first simulation, no matter how the number of tasks increased, the energy consumption will be effectively decreased, and the energy saving is about 78.81 percent. The comparison of energy consumption with and

TABLE 1

THE SIMULATION RESULTS WITH ENERGY CONSUMPTION AND ENERGY EFFICIENCY

Number of Tasks	Energy Consumption without DVS (nJ)	E-DVS with different deadlines		E-DVS with fixed deadline	
		Energy Consumption (nJ)	Energy Saving (%)	Energy Consumption (nJ)	Energy Saving (%)
5	189.401	45.5668	75.94%	0.783964	99.59%
10	366.401	81.3615	77.79%	5.27211	98.56%
15	542.892	117.767	78.31%	16.8605	96.89%
20	722.108	155.097	78.52%	39.344	94.55%
25	900.146	192.765	78.59%	75.7505	91.58%
30	1083.69	231.03	78.68%	129.228	88.08%
35	1259.57	266.797	78.82%	205.602	83.68%
40	1436.09	304.245	78.81%	303.953	78.83%

without E-DVS in different deadline is shown in Fig.6. In the second simulation, the deadline is fixed to 200*ms*, the energy saving is maintaining above 78.83 percent. The comparison of energy consumption with and without E-DVS in fixed deadline is shown in Fig.7.

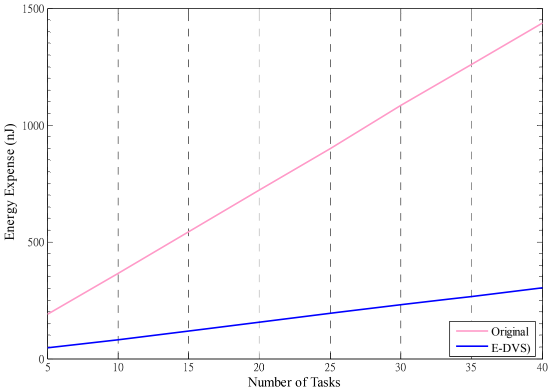


Fig. 6 Comparison of energy consumption with and without E-DVS in different deadline

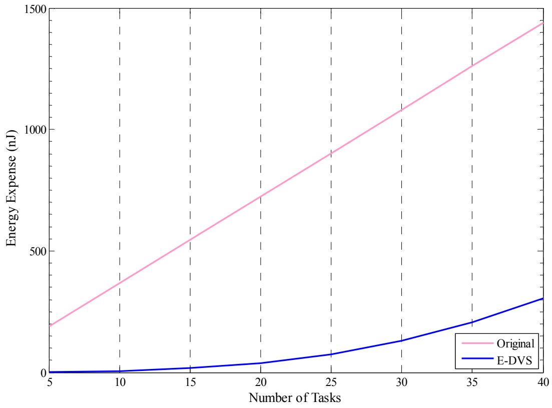


Fig. 7 Comparison of energy consumption with and without E-DVS in fixed deadline

5. CONCLUSION

We have proposed a novel dynamic scheduling algorithm for effectively reducing the energy consumption in this paper. The E-DVS algorithm is proved to be feasible to save energy by our simulation results. The E-DVS algorithm is applying the Excenics theory with the DVS algorithm. The proposed algorithm can easily be realized with a little system overhead. Furthermore, the simulation results show that the energy consumption can be decreased significantly by applying the proposed method. Comparing to the system without applying E-DVS, the energy saving with the proposed E-DVS is about 78.81 percent for running the same task set in our simulation.

REFERENCES

[1] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," *Solid-State Circuits, IEEE Journal*, Vol. 35, pp. 1571-1580, 2000.

[2] W. Cai, “The extension set and incompatible problem,” *Journal of Scientific Exploration*, Vol. 1, pp. 81-93, 1983.

[3] X. Jin Jian, H. Wang, N. Wun, D. Wu, and F. Wang Jian, "Energy Efficient Real-Time DVS based on Genetic Algorithm," in *Embedded Software and Systems, ICESS '08. International Conference*, 2008, pp. 98-104.

[4] M. Lin and C. Ding, "Parallel Genetic Algorithms for DVS Scheduling of Distributed Embedded Systems," *LECTURE*

NOTES IN COMPUTER SCIENCE, vol. 4782, pp. 180-191, 2007.

[5] G. Qu, "Power Management of Multicore Multiple Voltage Embedded Systems by Task Scheduling," in *Parallel Processing Workshops, ICPPW International Conference*, 2007, pp. 34-34.

[6] Z. Shao, M. Wang, Y. Chen, C. Xue, M. Qiu, L. T. Yang, and E. H. M. Sha, "Real-Time Dynamic Voltage Loop Scheduling for Multi-Core Embedded Systems," *Circuits and Systems II: Express Briefs, IEEE Transactions*, vol. 54, pp. 445-449, 2007.

[7] D. Wu, B. M. Al-Hashimi, and P. Eles, "Scheduling and mapping of conditional task graph for the synthesis of low power embedded systems," *Computers and Digital Techniques, IEE Proceedings*, vol. 150, pp. 262-273, 2003.

[8] Y. Zhu and F. Mueller, "Feedback EDF Scheduling of Real-Time Tasks Exploiting Dynamic Voltage Scaling," *Real-Time Systems*, vol. 31, pp. 33-63, 2005.