A Scheduling with DVS Mechanism for Embedded Multi-Core Real-Time Systems

Liang-Teh Lee, Shin-Tsung Lee, Chao-kai Tsai


A Scheduling with DVS Mechanism for Embedded Multi-Core Real-Time Systems

Liang-Teh Lee, Shin-Tsung Lee, Chao-kai Tsai

Department of Computer Science and Engineering, Tatung University, ltlee@ttu.edu.tw
doi:10.4156/jdcta.vol5. issue4.7

Abstract

With the advancement of technology, embedded systems have been widely used in portable devices. Portable embedded systems must have rather superior computing capability in order to meet real-time application demands. It is very important for the design of portable embedded system to enable minimum energy consumption while meeting real-time application demands. Dynamic voltage scaling technology enables effective reduction of energy consumption by utilizing slack time to modify operation voltage and frequency of processor in order to reduce energy consumption. Multi-core systems have gradually become mainstream products providing better throughput capacity than single-core systems under the same working clock frequency. However, in multi-core environment the real-time scheduling is an NP-hard issue. A real-time scheduling mechanism proposed for multi-core systems which in conjunction with dynamic voltage scaling can reduce energy consumption. The proposed ME-DVS (Modified EDF with Dynamic Voltage Scaling) scheduling algorithm can effectively reduce energy consumption in multi-core environment and ensure all tasks to meet their deadlines. The experimental results have shown that in 4-core environment, the energy consumption can be reduced by 3% when increasing the number of tasks to 30, 87% of processor total utilization, and by 52% when increasing the number of tasks to 5, 18% of processor total utilization. Similar results can be found in 8-core and 16-core environments.

Keywords: Scheduling Algorithm, Dynamic Voltage Scaling, Multi-core System, Real-time System, Embedded System

1. Introduction

With the advancement of technology, embedded systems have been widely used in portable devices such as PDA, Mobile Phone, and GPS. Portable embedded systems must have rather superior computing capability in order to meet real-time application demands. As the computation increases, so does the corresponding energy consumption. The energy consumption of portable embedded system is a very important issue due to limited battery capacity of the system. The working time of portable device can be prolonged if energy consumption of the system can be effectively reduced. Therefore it is very important for the design of portable embedded system to enable minimum energy consumption while meeting real-time application demands [11].

In an embedded system, processor is one of the devices which consume a lot of energy. The higher the frequency of processor, the greater the energy consumption will be. Therefore it is very important to figure out a way to reduce energy consumption of the processor. Dynamic voltage scaling technology enables effective reduction of energy consumption [3]. Nowadays most processors provide operating voltages at different levels for the purpose of reducing energy consumption. Dynamic voltage scaling technology reduces energy consumption of processor by utilizing slack time to modify the operating voltage [2]. For example, if one task is completed before the deadline in the processor, the extra slack time will be utilized by the processor to reduce the operating voltage and frequency such that energy consumption can be reduced.

Multi-core systems have gradually become mainstream products. With multiple processing units integrated on the same chip, multi-core system can provide higher throughput capability than single-core system with the same clock rate, while the energy consumption also increases. In multi-core environment the real-time scheduling is an NP-hard issue [8]. It’s important to consider the issues of all cores operating at the same voltage and frequency while using dynamic voltage scaling technology in general.
multi-core system environment [9].

The proposed real-time scheduling mechanism for multi-core systems which ensures all tasks in the core can meet their deadlines and can reduce energy consumption in conjunction with dynamic voltage scaling.

2. Background

Real-time system can be classified into hard real-time system and soft real-time system. In the soft real-time system there is no strict requirement for tasks to be completed within limited time frame, as long as they are completed as soon as possible. In the hard real-time system there is strict requirement that all tasks must be completed before their deadlines.[7][15] Most embedded systems consist of both soft and hard real-time tasks, while a portion of real-time systems concentrate on hard real-time. Therefore, the target of interest focus on is hard real-time system.

2.1. Real-time scheduler for single core processor

EDF as the optimized scheduling algorithm for single-core processor [13], EDF is a preemptive periodic real-time scheduling mechanism. When the sum of utilization of all tasks is equaled to or smaller than 1, EDF scheduling algorithm will guarantee all tasks to be completed before deadline. In EDF scheduling algorithm, the priority of task is set according to the deadline of each task, where higher priority will be assigned to the task closest to the deadline.

2.2. Dynamic voltage scaling

Dynamic voltage scaling is capable of effectively reduce the energy consumption for real-time embedded system. For processor, this is one kind of hardware technology which can dynamically modify the operating voltage and frequency of processor during operation such that the electricity consumption can be reduced. The speed of processor goes up during operation when voltage and frequency get higher, and so does the energy consumption. Conversely, the operating speed gets slower with less energy consumption when the voltage and frequency get lower. Dynamic voltage scaling is mainly about calculating and utilizing the remaining slack time of the task to reduce the operating voltage and frequency of processor. For example, if one periodic task is completed before deadline, the remaining time (slack time) can be utilized by processor to reduce the energy consumption. There are two kinds of dynamic voltage scaling technologies for hard real-time system: [6], Inter-task DVS and Intra-task DVS. The main difference between Inter-task DVS and Intra-task DVS is how the slack time is utilized. For the Inter-task DVS the slack time is utilized for subsequent tasks, while the Intra-task DVS the slack time is utilized for current tasks. With Inter-task DVS, a task is executed first, and when it is completed or preempted, the maximum operating time for the next task will be calculated and system voltage will be adjusted for it. When the actual execution time for the task is shorter than the worst execution time, the slack time will be generated and utilized by Intra-task DVS to modify the operating voltage for current task. The proposed dynamic voltage scaling mechanism in this paper is mainly focused on Intra-task DVS.

2.3. DVS problem in multi-core real-time system

Due to the fact that every core in the multi-core real-time system must operate at the same voltage and frequency, the dynamic voltage scaling mechanism can be applied to reduce the energy consumption. However, when applying the dynamic voltage scaling mechanism in the multi-core system must be carefully, sometimes it may result in the task of certain core that can not be completed before deadline.

A dynamic voltage scaling mechanism for hard real-time system will be proposed to reduce the electricity consumption for all cores, and to ensure all tasks to be completed before their deadlines.
2.4. DVS for single core processor

EDF scheduling algorithm is the optimized algorithm for single-core processor. Static EDF algorithm is mainly about arranging the operating order for tasks with EDF strategy, and reducing electricity consumption of processor by using slack time. The steps are described as follows: Step1. Determine task priority based on EDF scheduling method. Step2. Calculate utilization of all tasks to obtain the total utilization U. Step3. Determine operating frequency based on U. The following example can be used to explain this algorithm: assume there are three tasks $S_1$ ($e=3$, $p=8$), $S_2$ ($e=3$, $p=10$), $S_3$ ($e=1$, $p=14$), where $e$ is the worst case execution time, $p$ is the task period. The utilizations of three tasks and two actual execution times are shown in Table 1.

<table>
<thead>
<tr>
<th>Task</th>
<th>Utilization</th>
<th>First ACET</th>
<th>Second ACET</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>0.375</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>0.300</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>0.071</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The result of static EDF algorithm shows that when the worst case execution time equals to actual case execution time, all static slack time will be completely used to achieve the optimized effectiveness. However there is one major shortcoming of this algorithm, i.e., if the actual case execution time is less than the worst case execution time, excessive slack time will be generated [10].

2.5. DVS for multi-core processors

Most previous studies were focused on the scheduling and energy consumption issues of single-core processors and multiprocessors, while only a small number of studies were focused on multi-core processors. The dynamic voltage scaling technology for the multi-core system must modify voltages and frequencies of all cores simultaneously [2]. Even though there has been other kind of processor such as per-core DVS where each core can operate at different voltage and frequency levels, this kind of processor requires more power supply regulators and more complicated circuit transmission lines, it is more expensive and complicated [5].

An energy-aware soft real-time scheduling method for multi-core multi-threading system has been introduced by [2], and this scheduling algorithm in conjunction with dynamic voltage scaling technology can reduce energy consumption. This algorithm is targeting the minimum energy consumption of the task worst case execution time. When a task is established, a core will be activated and operated at minimum speed. If other tasks are also established, scheduling algorithm will evaluate whether or not the current core operating speed can meet the worst case execution time of the most recent task in order. If this core cannot meet the requirement, the second core will be activated and operated at the minimum speed. If this is still not enough to meet the worst operating time of the most recent task in order, then the core operating frequency will be raised to the maximum. After the completion of tasks, the core operating frequency will be lowered or each core will be shut down individually in order to reduce the energy consumption of the system.

3. DVS scheduling in multi-core real-time system

A scheduling method, ME-DVS, for multi-core real-time system has been proposed to reduce energy consumption in conjunction with dynamic voltage scaling technology, and it will be explained in this chapter. ME scheduling method is a static preemptive scheduling algorithm with the following features: (1) all tasks will be completed before deadline; (2) the sum of utilization of all tasks in multi-core environment will be used to determine the minimum number of cores required to arrange this task set; (3) utilization of each task will be applied to separate groups, and then the number of cores for each group will be determined. Firstly, to define system module, then the proposed multi-core real-time system...
scheduling method will be presented. Finally, the dynamic voltage scaling mechanism for multi-core scheduling method will be described.

3.1 System model

3.1.1. Task set

Tasks in the task set are assumed to be executed periodically where each task has its own deadline and it must be completed before deadline. Task set \( T \) is defined as shown in equation (1), where each task in the task set has its own execution time \( e_i \) and period \( p_i \), execution time \( e_i \) equals to the worst case execution time, and period \( p_i \) equals to deadline \( d_i \). Task utilization \( u_i \) is defined in equation (2), while the total utilization of task set \( U \) is defined in equation (3).

\[
T = \{t(e_1, p_1), t(e_2, p_2),...t(e_n, p_n)\}
\]

\[
u_i = \frac{e_i}{p_i}
\]

\[
U = \sum_{i=1}^{n} u_i
\]

3.1.2. Multi-core system

The processor \( P \) in a multi-core real-time system is composed of multiple homogeneous cores with the same structure, effectiveness and voltage/frequency level. As defined by equation (4), there are \( m \) cores in processor \( P \), and assumed that there is no shared resource or excessive time consumption among different cores. When the voltage and frequency of one core needs to be modified, all other cores in the system will be modified to the same voltage/frequency level. Therefore all cores in the system are at the synchronized voltage/frequency level. If there is one core in the multi-core system with no task scheduled for it, then the energy consumption of this core is zero.

\[
P = \{Core_0, Core_1, ..., Core_m\}
\]

3.1.3. Power model

The voltage, frequency, and energy consumption of the system are referred to the values measured by Transmeta Crusoe processor as shown in Table 2 [11]. There are four adjustable levels of voltage/frequency in the system where the maximum voltage is 1.5V and maximum frequency is 500MHz. If the operating frequency calculated by dynamic voltage technology is 167MHz, the system will automatically select operating voltage as the minimum value of 1.10V, and operating frequency as 200MHz.

<table>
<thead>
<tr>
<th>Voltage (V)</th>
<th>Frequency (MHz)</th>
<th>Relative Power (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.10</td>
<td>200</td>
<td>21.51</td>
</tr>
<tr>
<td>1.25</td>
<td>300</td>
<td>41.67</td>
</tr>
<tr>
<td>1.40</td>
<td>400</td>
<td>69.69</td>
</tr>
<tr>
<td>1.50</td>
<td>500</td>
<td>100</td>
</tr>
</tbody>
</table>

3.2. DVS scheduling in multi-core real-time system

In the hard real-time system, there is very strict requirement of deadline that all tasks must be completed before deadline. Thus, to determine the core to execute the next task, and ensure this task can
be completed before deadline in this core in the multi-core real-time system. A scheduling method for multi-core real-time system is proposed to accomplish the above requirement.

Task Set Partition: When the task set is established, the utilization $u_j$ of each task in this task set will be calculated first, then the utilization of each task will be accumulated according to equation (3) to obtain a value $U$, the total utilization, then determine the least amount of cores required to schedule all tasks for this task set in multi-core system based on equation (5) proven by [1]. The value $m$ will be obtained to determine the number of cores to be activated in multi-core system.

$$U \leq \frac{m^2}{3m - 2}$$

(5)

The partition of task set is mainly to obtain $u_j$ by using equation (2) and then assign the task to the corresponding group according to equations (6) and (7) [1][14]. Task group can be categorized into high weight task (HWT) and low weight task (LWT). If the utilization of a task calculated by equation (6) is less than or equal to $m/(3m-2)$, the task will be assigned to LWT; conversely, if by equation (7) the utilization is more than $m/(3m-2)$, it will be assigned to HWT. The partition of task set can reduce the chances of HWT and LWT mixed in the same core.

$$u_j \leq \frac{m}{3m - 2}$$

(6)

$$u_j > \frac{m}{3m - 2}$$

(7)

After the partition of task set, the number of cores to be assigned to two task sets will be determined. No core will be assigned to a task set if the task set is empty. If there is only one task in this task set, then one core will be assigned to it. If there are more than one tasks assigned to this set, then equation (5) will be used to determine the minimum number of cores needed for the scheduling of this task set.

![Figure 1. The flow chart of the task set scheduler](image-url)
3.2.1. Task set scheduler

After the arrangement of HWT and LWT, and the determination of core numbers for task sets, the scheduling of two task sets will be conducted. Firstly, the priority of task set will be determined according to the characteristic of EDF scheduling method, and then each task will be assigned to each core according to their priorities to complete the scheduling.[12] The flow chart of the task set scheduler is shown in Figure 1.

Assume that there are five tasks in a 4-core environment, where the period of each task equals to its deadline. The utilization, execution time, and period of each task are shown in Table 3. T1 and T4 are assigned to HWT. T1, T2, and T3 are assigned to LWT. Based on the calculation of equation (5), the minimum number of cores required for 5 tasks is 3. Two cores will be assigned to HWT while one core will be assigned to LWT. Based on EDF strategy, T3 has the higher priority over T4, so T3 is assigned to core 1 while T4 is assigned to core 2. T1, T2, and T5 are assigned to core 3 according to their priorities. The scheduling is as shown in Figure 2, the illustration of ME scheduler.

<table>
<thead>
<tr>
<th>Task</th>
<th>Execution</th>
<th>Period</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1</td>
<td>7</td>
<td>0.143</td>
</tr>
<tr>
<td>T2</td>
<td>2</td>
<td>16</td>
<td>0.125</td>
</tr>
<tr>
<td>T3</td>
<td>8</td>
<td>18</td>
<td>0.444</td>
</tr>
<tr>
<td>T4</td>
<td>10</td>
<td>23</td>
<td>0.435</td>
</tr>
<tr>
<td>T5</td>
<td>3</td>
<td>25</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Figure 2. ME scheduler in 4-core environment

Figure 3. EDF scheduler in 4-core environment
A Scheduling with DVS Mechanism for Embedded Multi-Core Real-Time Systems
Liang-Teh Lee, Shin-Tsung Lee, Chao-kai Tsai

Figure 3 is the illustration of EDF scheduling method. Comparing with Figure 2, while using EDF scheduling method to schedule 5 tasks in a 4-core environment, at least one task will be assigned to each core. With our multi-core scheduling method, ME, the scheduling of 5 tasks can be completed by only activating 3 cores. Therefore, under the same condition, the proposed scheduling method, ME, will save more core electricity consumption than that of applying EDF scheduling method.

3.2.2. ME scheduler with dynamic voltage scaling mechanism

Dynamic voltage scaling can effectively reduce the energy consumption of processor in multi-core real-time system, yet the deadline for tasks must be taken into consideration. This research combines dynamic voltage scaling mechanism with ME scheduling method. ME scheduling method will determine the number of cores to be activated based on the total task utilization for reducing the core energy consumption. Then it can be applied the dynamic voltage scaling mechanism to those activated cores for further saving the energy. Dynamic voltage scaling mechanism is mainly to determine the operating frequency and voltage, based on considering the interval between starting points of the current task and the next task. The frequency of current time slot of each core \( f_i \) (i is the core number) will be calculated by equation (8), \( N_t \) is the starting time of the next task, and \( C_t \) is the starting time of the current task. The calculated \( f_i \) must be compared to the calculated slot frequencies of other cores such that a maximum frequency can be selected from the frequency set for operation. When other cores proceed to the next slot, yet one of them still has not completed the previous slot, it must be compared to other cores. If there are 5 tasks in a 4-core environment, from equations (2), (6), and (7) these equations belong to the LWT with periods equal to their deadlines. The execution time and period of this task set are shown in Table 4.

\[
f_{\text{max, } i} = f_i \cdot (N_t - C_t)
\]  

(8)

Table 4. Task set execution time, task period

<table>
<thead>
<tr>
<th>Task</th>
<th>Execution</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>T2</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>T3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>T4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>T5</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 4. ME scheduler without DVS
The scheduling of 5 tasks in the 4-core environment by applying the multi-core scheduling method is shown in Figure 4. The operating frequencies of slots 1, 2, 3, and 4 are calculated by equation (8) as 500MHz, 200MHz, 200MHz, and 100MHz, respectively. The maximum frequency 500MHz will be selected for operation and the operating frequencies of slots 2, 3 and 4 will be tuned to the maximum frequency. When the operation of core 1 reaches time spot 2, slot 5 will be calculated because the first slots of other cores have not been completed yet. The calculated frequency of slot 5 will be compared to those frequencies of other cores (slot 2, 3, and 4), and will be operated based on the maximum frequency. Similarly, at time spot 5, slots 6, 7, and 8 will be compared to unfinished task (slot 4). However, at this time the operating frequency of slot 4 is the maximum, so slots 6, 7, and 8 will be operated at the maximum frequency. When the operation reaches time spot 10, the operating frequencies of slots 9, 10, 11 and 12 are 100MHz, 200MHz, 200MHz, and 200MHz, respectively. So 200MHz will be the operating frequency, and the maximum frequency of the system will be lowered to 200MHz. Therefore, the energy consumption can be reduced as shown in Figure 5.

Figure 5. ME scheduler with DVS

4. Simulation results

The simulation environment and the simulation results of the proposed multi-core real-time algorithm and dynamic voltage scaling mechanism will be described. The proposed scheduling algorithm, ME, will be compared with scheduling algorithm integrated with dynamic voltage scaling mechanism (ME-DVS), and will also be integrated with EDF scheduling algorithm for comparison. The simulation results show that the proposed dynamic voltage scaling mechanism can effectively reduce the system energy consumption.

4.1. Simulation environment

During the simulation of multi-core system, the system is running in the 4-core, 8-core, and 16-core environments with identical structure, voltage and frequency for each core. Every core must be operated simultaneously at the same voltage and frequency level. The energy consumption of multi-core system is based on the reference of the energy module described before. The simulation constructs six different task sets in the simulation with the number of tasks as 5, 10, 15, 20, 25, and 30, respectively. The task execution time are defined from 1ms to 5ms randomly, and the task execution period (the same as its deadline) are defined from 1ms to 50ms randomly. Tasks are assumed to be executed periodically and without independency between tasks. Total execution time is set to the least common multiple of each task period. Each task set will be operated in the same multi-core environment for 10 times, and the
average value of energy consumption will be calculated.

The energy consumption by applying different algorithms are compared and shown in the following figures. The effect of energy saving can be expressed by the following equation:

\[
\text{Energy Saving}(\%) = \frac{\text{energy}_{\text{Improved}}(\text{mJ}) - \text{energy}_{\text{Unimproved}}(\text{mJ})}{\text{energy}_{\text{Unimproved}}(\text{mJ})}
\]

4.2. Simulation result of applying ME scheduler algorithm

Figure 6 is the comparison between ME scheduling algorithm and EDF scheduling algorithm. It shows that energy consumption increases with the number of tasks. In 4-core environment, the proposed multi-core scheduling algorithm can determine how many cores to be activated according to the total task utilization. When there are 5 tasks in the task set, in the simulated multi-core environment on average only 2 cores needed to be activated to complete the task with ME scheduling algorithm, while with EDF scheduling algorithm there are 4 cores needed to be activated. Therefore, when the number of tasks is 5, 18% of processor total utilization, ME scheduling algorithm will save 65% of energy consumption comparing to the EDF scheduling algorithm, and when the number of tasks is 10, 40% of processor total utilization, the saving will be 2%. However, as the number of tasks greater than 10 in 4-core environment, the energy consumption with ME scheduling algorithm will be the same as EDF scheduling algorithm.

Figure 7. Comparison of energy consumption with different scheduling algorithm in 8-core
Figure 8. Comparison of energy consumption with different scheduling algorithm in 16-core

Figure 7 is the comparison between the energy consumption of ME scheduling algorithm and EDF scheduling algorithm in 8-core environment. When the number of tasks is 5, 12% of processor total utilization, ME scheduling algorithm will save 48% of energy consumption comparing to the EDF scheduling algorithm, and on average 2 cores are required for ME scheduling algorithm. When the number of tasks is 10, 20% of processor total utilization, ME scheduling algorithm will save 39% of energy consumption comparing to the EDF scheduling algorithm, and on average 3 cores are required for ME scheduling algorithm. Figure 8 is the comparison between the energy consumption of ME scheduling algorithm and EDF scheduling algorithm in 16-core environment. When the number of tasks is 5, 7% of processor total utilization, ME scheduling algorithm will save 53% of energy consumption comparing to the EDF scheduling algorithm, and on average 3 cores are required for ME scheduling algorithm. When the number of tasks is 10, 14% of processor total utilization, ME scheduling algorithm will save 28% of energy consumption comparing to the EDF scheduling algorithm, and on average 7 cores are required for ME scheduling algorithm. Obviously, when the number of tasks is small, multi-core scheduling algorithm (ME) is superior to EDF scheduling algorithm in terms of energy consumption of multi-core real-time system.

4.3. Simulation result of applying ME-DVS algorithm

This section will describe the simulation of the multi-core scheduling algorithm combined with dynamic voltage scaling mechanism. Figure 9 shows that the energy consumption increases as the increasing of the number of tasks. In the simulation, to compare the multi-core scheduling algorithm (ME) and the multi-core scheduling algorithm integrated with dynamic voltage scaling mechanism (ME-DVS). The result shows that in 4-core environment, there is 3% saving of energy consumption when 30 tasks are in the task set, 87% of processor total utilization; there is 52% saving of energy consumption when 5 tasks are in the task set, 18% of processor total utilization. When there are 5 tasks in the task set, 18% of processor total utilization, and no core in the system is shut down, there is 73% saving of energy consumption.
A Scheduling with DVS Mechanism for Embedded Multi-Core Real-Time Systems
Liang-Teh Lee, Shin-Tsung Lee, Chao-kai Tsai

Figure 9. Comparison of energy consumption with DVS scheduling algorithm in 4-core

Figure 10. Comparison of energy consumption with DVS scheduling algorithm in 8-core

Figure 11. Comparison of energy consumption with DVS scheduling algorithm in 16-core

Figure 10 is the simulation result of ME-DVS in the 8-core environment. Due to the number of cores increased, there is about 5% energy saving comparing to the 4-core environment when there are 30 tasks in the task set, 77% of processor total utilization. The simulation result of ME-DVS in the 16-core environment is shown in Figure 11. It shows that when there are 30 tasks in the task set, 50% of processor total utilization, ME-DVS will save 30% of energy consumption. From the simulation results, the proposed dynamic voltage scaling mechanism can effectively reduce the energy consumption in the multi-core environment when all cores are operating simultaneously at the same voltage and frequency level. And all tasks are guaranteed to be completed before their deadlines to meet the requirement of the hard real-time system.
4.4. Simulation result of applying EDF-DVS algorithm

The simulation results are shown in Figures 12, 13, and 14. In 4-core environment, the EDF scheduling algorithm combined with dynamic voltage scaling mechanism, about 13% of energy consumption can be saved when there are 30 tasks. In 8-core and 16-core environments, energy consumption can be saved 5% and 27%, respectively. From the simulation results the proposed dynamic voltage scaling mechanism can effectively reduce the energy consumption in the multi-core environment. Due to the fact, EDF scheduling algorithm is not the optimal scheduling algorithm for multiprocessor systems [4].

![Figure 12. Comparison of energy consumption with EDF-DVS scheduling algorithm in 4-core](image1)

![Figure 13. Comparison of energy consumption with EDF-DVS scheduling algorithm in 8-core](image2)

![Figure 14. Comparison of energy consumption with EDF-DVS scheduling algorithm in 16-core](image3)
4.5. System overhead

The proposed dynamic voltage scaling mechanism will examine the interval between the starting time of current task and the next task, and determine the operating frequency required for tasks to meet their deadlines. To determine the operating frequency will lead to energy consumption to be a system overhead. For applying the proposed mechanism, assume that the system is operated at the maximum voltage, i.e., 500MHz, for calculating the overhead. Simulation results show the average energy consumption of the overhead in 4-core environment is about 3301.5mJ, and in 8-core and 16-core environments are about 6199.5mJ and 13959.833mJ, respectively as shown in Figure 15. This energy consumption of system overhead is fairly small and thus can be neglected.

![Figure 15. The overhead of multi-core system](image)

5. Conclusions

This research proposes a real-time scheduling algorithm for multi-core systems which in conjunction with dynamic voltage scaling mechanism to reduce the energy consumption. In most multi-core environments all cores must be operated at the same voltage and frequency level, therefore, when applying the dynamic voltage scaling mechanism must take into consideration that after adjusting the voltage and frequency, tasks in every core must be ensured to be completed before their deadlines. The simulation results show that ME-DVS can effectively reduce energy consumption in the multi-core system and guarantee all tasks to be completed before their deadlines. The proposed algorithm can be operated with a little system overhead. From the comparison between multi-core dynamic voltage scaling scheduling algorithm (ME-DVS) with and without dynamic voltage scaling mechanism (ME), in 4-core environment, 3% of energy saving can be achieved when there are 30 tasks in the task set. For 8-core environment the energy saving is 5%, and for 16-core environment the energy saving is 30%. This mechanism can combine the dynamic voltage scaling mechanism with EDF scheduling algorithm to achieve the same energy saving effect for further proving that the proposed dynamic voltage scaling mechanism can reduce the energy consumption of the multi-core real-time system effectively. Based on the simulation results, the system overhead caused by applying the proposed mechanism is small enough to be neglected.

In the future, the proposed energy saving mechanism can be modified and extended to dynamic scheduling method to effectively reduce the energy consumption of multi-core real-time systems with non-periodic tasks and unknown number of tasks.

6. Acknowledgment

Financial support for this work was provided by the Tatung University, Taiwan, R.O.C. with grant number B98-102-065.
7. References