ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013

DVS Scheduling in Multi Core Real Time System

X.JUDE ROY JEYASEELAN

Abstract- **Embedded systems have been widely used in portable devices. To meet real time application demands computing capability of the embedded system should be high. It is very important for designing of embedded system to enable minimum energy consumption to while meeting the 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 providing better throughput capability than single-core processor working under in same clock frequency. The proposed IEDF-DVS (Improved EDF with Dynamic Voltage Scaling) scheduling algorithm can effectively reduce energy consumption in multi-core environment and ensure all tasks to meet their deadlines.**

Index Terms- **EDF Algorithm, Improved EDF Algorithm, DVS Mechanism**

I. Introduction

Embedded systems have been widely used in portable devices such as Mobile Phone, PDA and GPS. To meet real time application demands computing capability of the embedded system should be high. As the computation process increases, so does the corresponding energy consumption increases gradually. Due to limited battery capability of the embedded system it is very important issue. If energy consumption of the system is reduced then the working time of the embedded system will be increased. Therefore it is very important factor to consider while meeting real-time application demands.

In an embedded system, processor is one of the devices which consume a lot of energy. The higher the frequency of processor, the energy consumption will be high. Therefore it is very important to figure out a way to reduce energy consumption of the processor. Dynamic voltage scaling technology reduces energy consumption of processor by utilizing slack time to modify the operating voltage. Dynamic voltage scaling technology enables effective reduction of energy consumption. 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.

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

II. Related Work

In [14] a flexible task scheduling algorithm for grid computing after researching on the existing QoS Guide Min-min scheduling algorithm intensively, which is called distance-weighted measurement methodology (DWMM)[14]. QoS guide Min-min algorithm only divided the applications into two classes simply which has QoS requirement or no. In order to measure this distance accurately, we introduce a distance weighted measurement methodology (DWMM). The application can be executed on a resource while the distance is the shortest.

In [9] a Multiple clock domain (MCD) microarchitecture, which uses a globally asynchronous, locally synchronous (GALS) clocking style, permits future aggressive frequency increases, maintains a synchronous design methodology, and exploits the trend of making functional blocks more autonomous. In MCD, each processor domain is internally synchronous, but domains operate asynchronously with respect to one another. Designers still apply existing synchronous design techniques to each domain, but global clock skew is no longer a constraint. Global dynamic voltage scaling already appears in many systems and can help reduce power dissipation for rate based and partially idle workloads. MCD architecture can save power even during intensive computation by slowing domains that are comparatively unimportant to the application's current critical path, even when it is impossible to completely gate off those domains. The disadvantage is the need for inter domain synchronization, which, because of buffering, out-of-order execution, and superscalar data paths, has a relatively minor impact on overall performance, less than 2 percent [9].

III. Proposed Work

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

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. A real-time scheduling mechanism proposed for multi-core systems which in conjunction with dynamic voltage scaling can reduce energy consumption.

Improved EDF scheduling method is a static pre-emptive scheduling algorithm with the following features:

- All tasks will be completed before deadline.
- 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.
- Utilization of each task will be applied to separate groups, and then the number of cores for each group will be determined.

All Rights Reserved © 2013 IJARCET

B. System model

B.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

$$T=\{t_1(e_1,p_1), t_2(e_2,p_2),\ldots, t_n(e_n,p_n)\} \quad (1)$$

Where each task in the task set has its own execution time (e_i) and period (p_i), period (p_i) is equals to deadline (d_i).

B.2 Task utilization

Task utilization (u_i) is defined as the ratio between execution time (e_i) and the period (p_i)

$$u_i=\frac{e_i}{p_i} \quad (2)$$

Utilization U factor is equal to sum of all task utilization (u_i)

$$U=\sum_{i=1}^{n} u_i \quad (3)$$

B.3 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.

$$P= \{core_0, core_1, \ldots core_m\} \quad (4)$$

As defined by equation 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 synchronize 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.

B.4 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. There ar
e 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 2: Voltage, Frequency, and Energy consumption

Voltage (V)	Frequency (MHz)	Relative Power (%)
1.10	200	21.15
1.25	300	41.67
1.40	400	69.69
1.50	500	100

C. Earliest deadline first Algorithm

Earliest deadline first (EDF) is a dynamic scheduling algorithm used in real-time operating systems. It places processes in a priority queue. Whenever a scheduling event occurs (task finishes, new task released, etc.) the queue will be searched for the process closest to its deadline. This process is the next to be scheduled for execution.

EDF is an optimal scheduling algorithm on pre-emptive uniprocessors, in the following sense: if a collection of independent jobs, each characterized by an arrival time, an execution requirement, and a deadline, can be scheduled (by any algorithm) such that all the jobs complete by their deadlines, the EDF will schedule this collection of jobs such that they all complete by their deadlines.

C.1 Example

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.

Table 3: Task set, execution time, and period

Task	Execution	Period	Utilization
T1	1	7	0.143
T2	2	16	0.125
T3	8	18	0.444
T4	10	23	0.435
T5	3	25	0.12

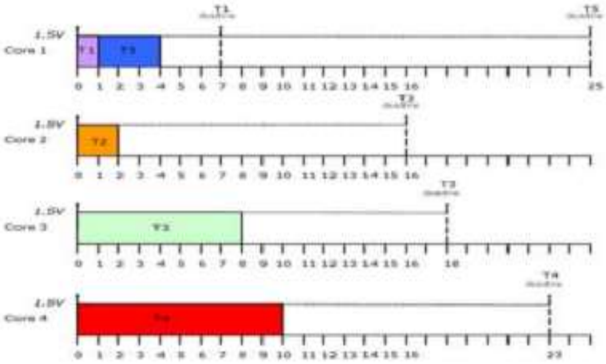

Fig 1: EDF scheduler in 5-core environment

All Rights Reserved © 2013 IJARCET

C.2 Improved EDF Algorithm

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.

C.3 Task Set Partition

When the task set is established, the utilization u(i) 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).

The value **m** will be obtained to determine the number of cores to be activated in multi-core system [1].

$$U \leq \frac{m^2}{3m-2} \quad (5)$$

C.4 Low Weight Task (LWT)

$$u_i \leq \frac{m^2}{3m-2} \quad (6)$$

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

C.5 High Weight Task (HWT)

$$u_i \geq \frac{m^2}{3m-2} \quad (7)$$

If the utilization of a task calculated by equation (7) is more than m/(3m-2), it will be assigned to HWT.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.

3.6.3.1 Example

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 4.

Task	Execution	Period	Utilization
T1	1	7	0.143
T2	2	16	0.125
T3	8	18	0.444
T4	10	23	0.435
T5	3	25	0.12

Table 4: Task set, execution time, and period

Based on equation (6) and (7) T3 and T4 are assigned to HWT and T1, T2 and T5 are assigned to LWT. The minimum number of cores required for 5 tasks is 3, based on equation (5). 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.

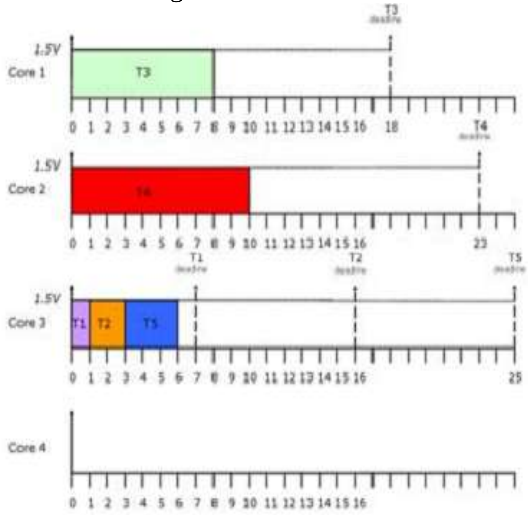

Fig 3: IE scheduler in 4-core environment

T1, T2, and T5 are assigned to core 3 according to their priorities. The scheduling is as shown in Figure 3, the illustration of IE scheduler.

Figure 1 is the illustration of EDF scheduling method. Comparing with Figure 3, 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, IE, the scheduling of 5 tasks can be completed by only activating 3 cores. Therefore, under the same condition, the proposed scheduling method, IE, will save more core electricity consumption than that of applying EDF scheduling method.

D. Improved EDF with DVS Technology

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 IE scheduling method. IE 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

All Rights Reserved © 2013 IJARCET

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,

$$f_{max}.\ e_i = f_i.(N_t - C_t) \qquad (8)$$

Where, N_t is the starting time of the next task, and C_t is the starting time of the current task. The calculated if 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 5.

Table 5: Task set, execution time, task period

Task	Execution	Period
T1	1	5
T2	4	15
T3	2	5
T4	2	5
T5	2	10

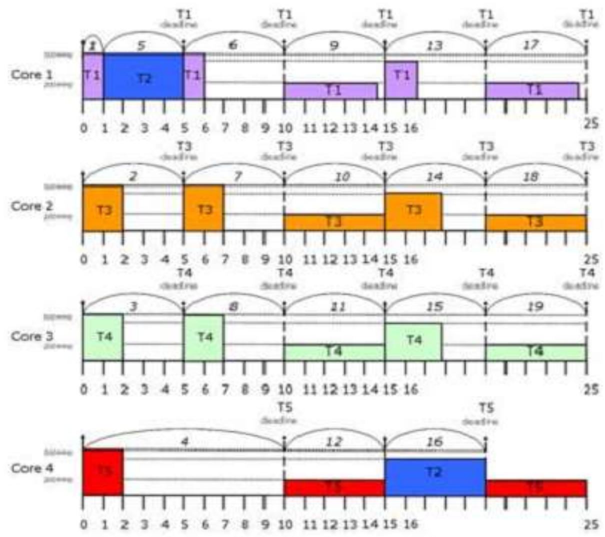

Figure 4: IEDF scheduler with DVS

The scheduling of 5 tasks in the 4-core environment by applying the multi-core scheduling method, 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 4.

IV. 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 simulation results show that scheduling of EDF and IEDF algorithm.

A. EDF scheduler in 4-core environment

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 6.

Table 6: Task set, execution time, task period

TASK	EXECUTION	PERIOD
T1	1	16
T2	3	17
T3	5	14
T4	5	12
T5	2	20
T6	2	10

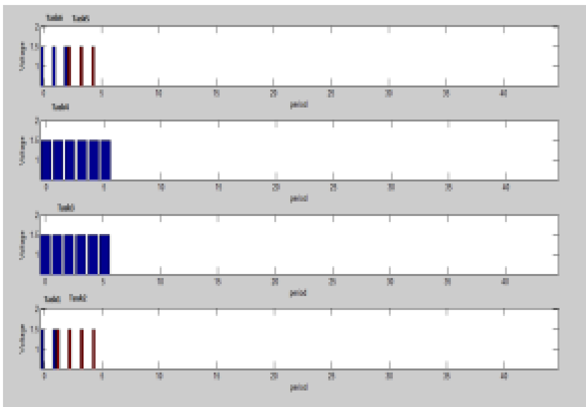

Figure 5: Simulation result of EDF scheduler 5 Task in 4-core environment

All Rights Reserved © 2013 IJARCET

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013

B. Improved EDF scheduler in 4-core environment

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 7.

Table 7: Task set, execution time, task period

TASK	EXECUTION	PERIOD
T1	5	15
T2	3	9
T3	1	12
T4	5	12
T5	4	10
T6	2	8

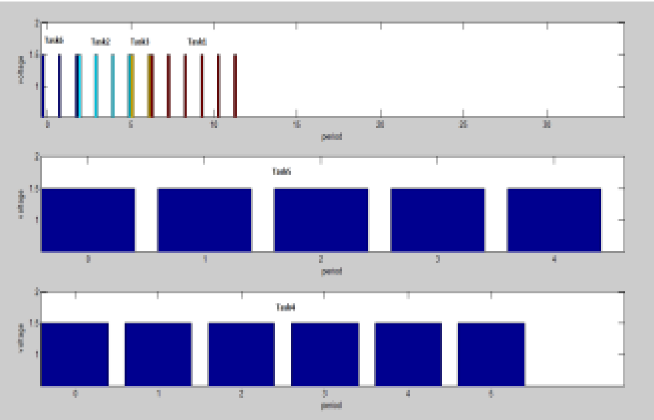

Fig 6: Simulation result of IEDF scheduler

Figure 5 is the illustration of EDF scheduling method. Comparing with Figure 6, 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, IEDF, the scheduling of 5 tasks can be completed by only activating 3 cores. Therefore, under the same condition, the proposed scheduling method, IEDF will save more core electricity consumption than that of applying EDF scheduling method.

V. CONCLUSION

This paper 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. We have studied scheduling algorithm of EDF and Improved EDF. Comparing with our simulation results of EDF and IEDF scheduling method, in 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, IEDF, the scheduling of 5 tasks can be completed by only activating 3 cores. Therefore, under the same condition, the proposed scheduling method, IEDF will save more core electricity consumption than that of applying EDF scheduling method. In future have to simulate IEDF scheduling mixed with DVS technology to reduce the energy consumption of the multi-core real-time system effectively

VI. REFERENCES

[1] Andersson, B., Baruah, S. and J. Jonsson, “Static-priority scheduling on multiprocessors”, In Proceeding of the 22nd IEEE Real-Time Systems Symposium, pp.193-202, 2001.

[2] Bautista, D., Sahuquillo, J., Hassan, H., Petit, S. and J. Duato, “A simple power-aware scheduling for multicore systems when running real-time applications”, In Proceeding of the IEEE International Symposium on Parallel and Distributed Processing, pp.1-7, 2008.

[3] Burd, T.D., Pering, T.A., Stratakos, A.J. and R.W. Broderson, “A dynamic voltage scaled microprocessor system”, Journal of IEEE Solid-State Circuits, vol. 35, no.11, pp.1571-1580, 2000.

[4] Dertouzos, M.L. and A.K. Mok, “Multiprocessor online scheduling of hard-real-time tasks”, Journal of IEEE Transactions on Software Engineering, vol.15, no.12, pp.1497-1506, 1989.

[5] Donald, J. and M. Martonosi, “Techniques for multicore thermal management: Classification and new exploration”, In Proceeding of the 33rd International Symposium on Computer Architecture, pp.78-88, 2006.

[6] Kim, W., Shin, D., Yun, H.S., Kim, J. and S.L. Min, “Performance comparison of dynamic voltage scaling algorithms for hard real-time systems”, In Proceedings of the 8th IEEE Real-Time and Embedded Technology and Application Symposium, (RETAS’02), IEEE Computer Society, pp.219-228, 2002.

[7] Labrosse, J.J., MicroC OS-II, The Real-Time Kernel, 2nd Edn., CMP Books, San Francisco, CA, 1998

[8] Leung, J.Y.T. and J. Whitehead, “On the complexity of fixed-priority scheduling of periodic, real-time tasks”, Journal of Performance Evaluation, vol.2, no.4, pp.237-250, 1982.

[9] Magklis, G., Semeraro, G., Albonesi, D.H., Dropsho, S.G., Dwarkadas, S. and M.L. Scott, “Dynamic frequency and voltage scaling for a multiple-clock-domain microprocessor”, Journal of IEEE Micro., vol.23, no.6, pp.62-68, 2003.

[10] Pillai, P. and K.G. Shin, “Real-time dynamic voltage scaling for low-power embedded operating systems”, In Proceedings of the 18th ACM

All Rights Reserved © 2013 IJARCET

Symposium on Operating Systems Principles, (SOSP'01), pp.89-102, 2001.
[11] Pouwelse, J., Langendoen, K. and H. Sips, "Dynamic voltage scaling on a low-power microprocessor", In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, (ICMCN'01), pp.251-259, 2001.
[12] Ren, X.Y., Qi, Z.H., Wang, R.C. and X.D. Ma, "Rough Set and A* based Tasks Scheduling in Gird Environment", Journal of JDCTA, vol.4, no.4, pp. 193-198, 2010.
[13] Seo, E., Jeong, J., Park, S. and J. Lee, "Energy efficient scheduling of real-time tasks on multicore processors", Journal of IEEE Transactions on Parallel and Distributed Systems, vol.19, no.11, pp.1540-1552, 2008.
[14] Song, K., Ruan, S. and M. Jiang, "A Flexible Grid Task Scheduling Algorithm Based on QoS Similarity", Journal of JCIT, vol.5, no.7, pp.161-166, 2010.
[15] Tian, H., Wang, X. and R. Henryk, "A New Approach to Computing Weighted Attributes Values in Incomplete Information Systems", Journal of JCIT, vol.2, no.2, pp. 21-29, 2007.

Author:

X.JUDE ROY JEYASEELAN,
PG SCHOLAR,
DEPTMENT OF ECE,
COIMBATORE INSTITUTE OF TECHNOLOGY,
COIMBATORE.

All Rights Reserved © 2013 IJARCET