Energy-Aware Fault Tolerance in Hard Real-Time Embedded Systems

S.Subha, N.Kumaresan

Abstract: ***Energy consumption of electronic devices has become a serious concern in recent years. Energy efficiency is necessary to lengthen the battery lifetime in portable systems, as well as to reduce the operational costs and the environmental impact of stationary systems. Dynamic power management (DPM) algorithms aim to reduce the energy consumption at the system level by selectively placing components into low-power states. Dynamic voltage scaling (DVS) algorithms reduce energy consumption by changing processor speed and voltage at run-time depending on the needs of the applications running. The proposed method is extended by integrating the DPM model DVS algorithm, thus enabling larger energy savings. The proposed methods are i) Postponement method and ii) Hybrid method. fault tolerance are also achieved by increasing transistor density and decreasing supply voltage.***

Keywords: Energy efficient; Real-time systems, DVS, DPM, Reliability.

I. INTRODUCTION

Energy minimization has always been a critical design criteria for portable embedded systems. DVS (dynamic voltage scaling) and DPM (dynamic power management) are the two main techniques to reduce the energy consumption for such systems. DVS saves energy by operating the system at a lower frequency and thus a lower voltage, while DPM saves energy by putting the system into a lower power state when the idle time is long enough. There has been lots of work in the areas of DVS and DPM. Most DVS is associated with task scheduling, and can be classified into offline task scheduling algorithms or online task scheduling algorithms.

Work in DPM has centered around prediction of future idle periods aggregation of small idle times to get longer idle durations stochastic control techniques based on Markov chain models etc. Dynamic Voltage Scaling (DVS),have negative impacts on the system reliability. As a result, in designing safety-critical hard real-time systems, it is inevitable to consider fault-tolerance and energy consumption jointly.

This paper describes how to to reduce energy consumption and achieve fault tolerance at the same time in a real-time system. The aim of this paper is comparing the proposed1. Postponement method and 2. Hybrid method with existing energy management methods. The postponement method is the most preferable when reliability is main. Hybrid method is most appropriate during energy consumption.

Revised Manuscript Received on 30 July 2012
*****Correspondence Author(s)
S. Subha*, PG Student, Department of ECE, Anna University of Technology, Coimbatore, India.
N. Kumaresan, Assistant Professor, Department of ECE, Anna University of Technology, Coimbatore, India.

II. RELATED RESEARCHES

In this section, we review several related researches in this area. There has been lots of work in the areas of DVS and DPM. Hard real-time systems often require hardware redundancy (replication) [1], but the imposed redundancy leads to more energy consumption. On the other hand, most popular system level energy-management techniques, e.g., Dynamic Voltage Scaling (DVS), have negative impacts on the system reliability [2], [3]. As a result, in designing safety-critical hard real-time systems, it is inevitable to consider fault-tolerance and energy consumption jointly. Recently, researchers have considered both fault-tolerance and energy consumption in real-time scheduling. Melhem et al. and Zhang and Chakrabarty consider the use of checkpointing schemes for fault tolerance, when DVS is used for low energy consumption [4], [5]. These works have only focused on checkpointing for fault-tolerant and have not considered replication methods that are often used in hard real-time systems. Furthermore, they have not considered DPM. Ejlali et al. propose to use a combination of information redundancy and time redundancy in order to decouple the system fault tolerance from slack time and leave more slack time for DVS and another system level energy management technique (i.e., adaptive body biasing) [6]. However, this work also has not considered replication methods, and DPM. Pop et al. propose methods for scheduling processes in low power and fault-tolerant hard real-time distributed embedded systems [7].

III .MODELS DESCRIPTION

A. System and Application

In this paper, we consider a periodic real-time application which consists of a set of periodic processes with deadline, D, which is also the length of period. throughout the paper, we will focus only on a single period of execution. For safety-critical applications we consider a system with two identical processors which are DVS-enabled. One processor is used as a primary unit to execute an application and the other one operates as a secondary processor and executes the same application as a backup. the system supply voltage may vary between Vmin and Vmax continuously.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

B.Error Detection

We use replication for fault tolerance, i.e., when an error is detected the replicated unit replaces the faulty one. Many error detection techniques can only detect errors at the end of the process execution. Some examples are: result checking arithmetic codes, instruction counting, etc. Therefore, in this paper we also assume that errors are only detected when a process finishes.

C.Energy Model

In addition to switching power each processor dissipates speed independent power e.g., subthreshold leakage power. After finishing the execution of a period of an application, until the beginning of the next period, the processor can be hibernated, aiming at reducing the energy consumption (DPM). Even in the hibernation mode a fraction of power is consumed to maintain the system circuit on and keep clock running. This fraction of power is removed, only by turning the processor off. Thus, in the active mode with the supply voltage S, the total power consumption can be determined by,

$$P=P_{hib}+P_{si}+{}_{\varphi}.S^3 \qquad (1)$$

Consider an application running at the supply voltage S for $\Gamma(S_{max/}S)$ time units, since energy is the integral of power consumption over time, the system total energy consumption can be determined by

$$E(S)=P_{hib}.D+(p_{Si}+{}_{\varphi}.\ S^3)\ \Gamma(S_{max}/s) \qquad (2)$$

By using normalized values $S_{max=1,\ \varphi=1,\ \text{and}}$ D=1 ,and using the factors of m and n, (2) can be converted to a simpler form which is shown in

$$E(S)=n+(m+s^3)\ \Gamma/s \qquad (3)$$

D. Reliability Model

In hard real-time systems, errors might be caused by different kinds of fault, such as permanent, transient, or intermittent faults. In this paper, we are interested in tolerating transient faults, which are more important and critical in nowadays systems. The most popular kind of transient faults are caused by cosmic rays or high energy particles, which are also called soft-errors. In this paper we consider systems that use duplication to tolerate up to one transient fault (one-fault-tolerant is a common assumption. . If the primary and secondary copies of the application run at the supply voltage S1 and S2 respectively, and the execution time of the application (at the maximum supply voltage Smax) be Γ, the reliability of the whole system can be expressed as

$$R=1-(1-e^{-\Lambda(s_1)\Gamma Smax/S_1})\,(1-e^{-\Lambda(s_2)\Gamma Smax/S_2}) \qquad (4)$$

E. Markov Model

Markov model is used in order to analyze fault occurrence probability as well as average energy consumption. Under this model, a period of an application execution is considered to have a number of different states (or scenarios), each representing a different level of energy consumption. For a Markov model consisting of n states, let E(statei) be the total energy consumption in the ith state, and P(statei) be the probability of being in the ith state. the average energy consumption can be determined by

$$Eavg=\sum_{i=1}^{n} P(Statei)E(Statei) \qquad (5)$$

IV.ENERGY CONSUMPTION USING CLASSIC DPM AND CLASSIC DVS

A. Classic DPM

In this method, both the primary and secondary copies of a duplicated process are run at the maximum supply voltage and operational frequency (Smax) from the start of the period. In other words, this method does not use the dynamic voltage scaling capability of processors but instead uses the slack time for putting the processors in a hibernation mode to reduce the energy consumption. This method has a valuable advantage for hard real-time systems with high level of reliability requirements, that is processing at the maximum possible supply voltage results in minimum failure rate and consequently minimum probability of error occurrence. Since both the primary and secondary copies start at the same time and run at the maximum supply voltage, the total energy consumption is simply two times the energy of equation (2).

B. Classic DVS

This method proposes to use the DVS capability of processors in order to achieve the minimum possible energy consumption for each process execution. similar to classic DPM, in this method both the copies of the duplicated process start at the same time (i.e., at the start of the period), and use the same supply voltage. Hence, the total energy consumption is simply two times the energy of (2) with optimal normalized supply voltage.

C. Problems using classic DVS and Classic DPM

Classic DVS is widely used and well understood making implementation relatively straight forward .But Care must be taken to avoid slowing the system down too much and causing missed deadlines.

Classic DPM highly beneficial for energy savings for devices that have low duty cycles. Requires accurate prediction of duration of sleep state to determine if a transition is worth it, called the break-even time.

V. PROPOSED METHOD

To explain the proposed technique, we consider a periodic real- time application consists of a set of periodic processes with deadline, D.

C. Postponement Method

Like the classic DPM method, in the Postponement method the processors are run at the maximum supply voltage and we can tolerate one transient fault. Hence, the reliability of the Postponement method is the same as that of the classic DPM.To analyze the energy consumption of the Postponement method;

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

we need to consider the following cases with respect to the system workload.

Case (A)- Low workload

In this case, the workload is low enough to postpone the secondary (replicated) copy of the process to a time after finishing the primary process. This means that, in a period of the application, if the primary copy finishes successfully, we do not require executing any portion of the secondary process. However, if the primary copy of the process fails, the secondary copy must be executed completely. Markov model technique is used in this method.

The average energy consumption is,

$$E=2P_{hib}.D+(P_{si}+\varphi.S^3_{max})\tau +\lambda_0(P_{si}+\varphi.S^3_{max})\ \tau \qquad (6)$$

Case (B)- High workload

When no fault happens, the primary copy is executed completely and the secondary copy is executed for (2τ-D) units of time.

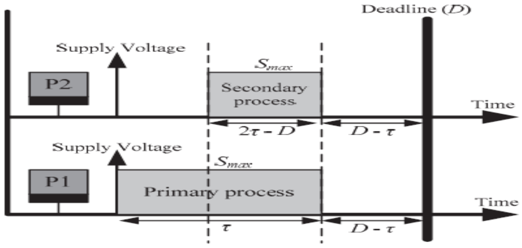

Figure 4. Postponement Method

There is no need to execute the remaining (D- τ)of the secondary copy, unless the primary copy becomes faulty.When using using the Markov model technique the average energy consumption is

$$E=2P_{hib}.D+(P_{si}+\varphi.S^3_{max})\tau+(P_{si}+\varphi.S^3_{max})(2\tau-1+\lambda_0(P_{si}+\varphi.S^3_{max})(D-\tau) \qquad (7)$$

D.Hybrid Method

Hybrid method aims at improving the Postponement method from the energy consumption viewpoint by exploiting the DVS technique. In this method the primary copy of the process is run at a reduced supply voltage and the secondary copy is postponed and also executed at a reduced supply voltage. Here, selecting the supply voltages of the primary and secondary copies, and selecting the best time for starting the secondary copy should be done carefully to achieve maximum energy saving.

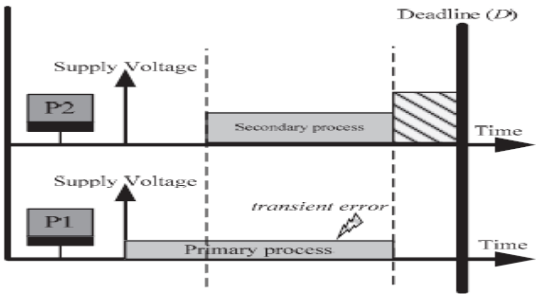

Figure 5. Hybrid Method

Case A)- Low workload

In this case, if no fault occurs, it is always possible to put the second processor into the hibernation mode all the time. But if a fault occurs during the execution of the primary copy, the secondary copy will be started right after finishing the faulty execution of the primary copy. This case is similar to Case A of the Postponement method. The average energy consumption and reliability of this method are

$$E=2P_{hib}.D+(P_{si}+\varphi(\varsigma.S^3_{max})\tau/\varsigma+\lambda(\varsigma.S_{max})(P_{si}+\varphi.S^3_{max})\ \tau \qquad (8)$$

$$R=1-(1-e^{-\lambda(\varsigma.S_{max})\tau/\varsigma})(1-e^{-\lambda\tau}) \qquad (9)$$

Case B)- High workload

Like Case A, if no fault occurs, it is always possible to put the second processor into the hibernation mode all the time. The average energy consumption and

$$E=2P_{hib}.D+(P_{si}+\varphi(\theta S^3_{max})^3)\tau/\theta$$
$$+(P_{si}+\varphi(\psi.S_{max})^3\ )\tau-(D-\tau/\theta)/\psi$$
$$+\lambda(\theta.S_{max})(P_{si}+\varphi.S^3_{max})(D-\tau/\theta) \qquad (10)$$

reliability are determined by,

$$R=1-(1-e^{-\lambda(\theta.S_{max})\ \tau/\theta})\ (1-e^{-\lambda(\psi.S_{max})\ \tau/\psi})$$
$$(1-\ e^{-\lambda 0(D-\tau/\theta)}) \qquad (11)$$

V. ESULTS AND DISCUSSION

A. Energy consumption –Comparison

We compare the energy consumption and reliability of all the four methods. Postponement method provides the same reliability as the classic DPM method, it consumes much less energy. By comparing the energy consumption of the classic DVS and Hybrid methods. we conclude that the energy consumption of the Hybrid methods less than or equal to that of the classic DVS for all system workloads.

Also, it can be seen that for lower system workloads the difference between the energy consumption of the classic DVS and Hybrid methods is relatively greater. The reason is that in lower system workloads, the replicated copy of the process can be postponed for relatively greater durations, and consequently the more energy conservation is achieved.

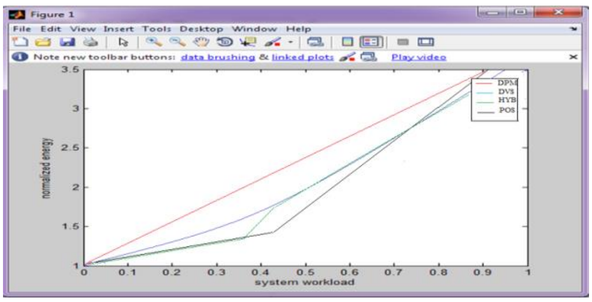

Figure 6.Simulation output for Energy consumption

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

B.Reliability--Comparison

By comparing the reliabilities of the Hybrid and classic DVS methods, we can conclude that the reliability loss in the Hybrid method is much less than that of the classic DVS. The reason is that in the Hybrid method we do not only use voltage scaling and we also use postponement.. systems, instead of using the classic DVS method. For very low and very high system workloads, there is no significant difference between the energy consumptions and also the reliabilities of the four methods.

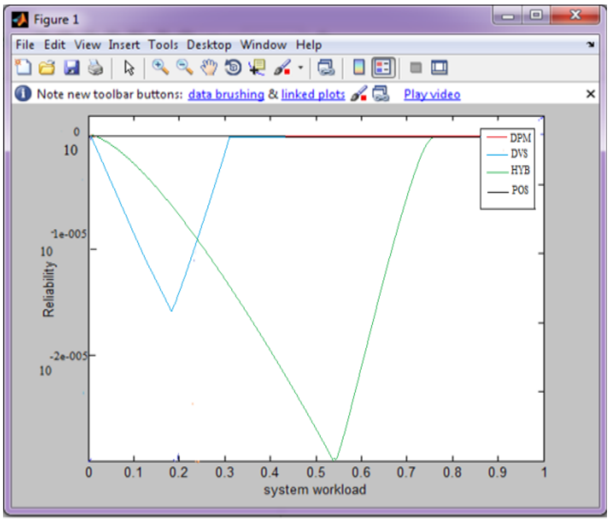

Figure 7.Simulation output for Reliability

VI. CONCLUSION

Low energy consumption and fault-tolerance are two major objectives in designing hard real-time embedded systems. In this paper, we use analytical models to explore and compare four different energy management methods used for reducing the energy consumption of replicated systems. Our analytical results show that, generally the Postponement and Hybrid methods are better than Classic DVS and DPM methods. Although, the Postponement method provides higher levels of reliability, in most cases the Hybrid method is better from the viewpoint of energy consumption.

REFERENCES

1. S. Poledna, Fault-Tolerant Real-Time Systems: The Problem of Replica Determinism. Kluwer Academic Publishers, 1996
2. A. Ejlali, B.M. Al-Hashimi, M.T. Schmitz, P. Rosinger, and S.G. Miremadi, "Combined Time and Information Redundancy for SEU-Tolerance in Energy-Efficient Real-Time Systems," IEEE Trans. Very Large Scale Integration Systems, vol. 14, no. 4, pp. 323- 335, Apr. 2006.
3. D. Zhu, R. Melhem, and D. Moose, "The Effects of Energy Management on Reliability in Real-Time Embedded Systems," Proc. Int'l Conf. Computer Aided Design (ICCAD '04), pp. 35-40, Nov. 2004
4. R. Melhem, D. Mosse, and E. Elmootazbella, "The Interplay of Power Management and Fault Recovery in Real-Time Systems," IEEE Trans. Computers, vol. 53, no. 2, pp. 217-231, Feb. 2004
5. Y. Zhang and K. Chakrabarty, "Dynamic Adaptation for Fault Tolerance and Power Management in Embedded Real-Time Systems," ACM Trans. Embedded Computing Systems, vol. 3, no. 2, pp. 336-360, May 2004
6. A. Ejlali, B.M. Al-Hashimi, M.T. Schmitz, P. Rosinger, and S.G. Miremadi, "Combined Time and Information Redundancy for SEU-Tolerance in Energy-Efficient Real-Time Systems," IEEE Trans. Very Large Scale Integration Systems, vol. 14, no. 4, pp. 323- 335, Apr. 2006
7. P. Pop, K.H. Harbo, V. Izasimov, and P. Eles, "Scheduling and Voltage Scaling for Energy/Reliability Trade-Offs in Fault- Tolerant Time-Triggered Embedded Systems," Proc. Fifth IEEE/ ACM Int'l Conf. Hardware/Software Codesign and System Synthesis (CODES+ISSS '07), pp. 233-238, 2007

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)