Ain Shams University
Ain Shams Engineering Journal
www.elsevier.com/locate/asej
www.sciencedirect.com

Dynamic power management techniques in multi-core architectures: A survey study

Khaled M. Attia *, **Mostafa A. El-Hosseini**, **Hesham A. Ali**

Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

Received 10 May 2015; accepted 3 August 2015

KEYWORDS

Chip multiprocessors;
Multi-core;
Power management

Abstract Multi-core processors support all modern electronic devices nowadays. However, power management is one of the most critical issues in the design of today's microprocessors. The goal of power management is to maximize performance within a given power budget. Power management techniques must balance between the demanding needs for higher performance/throughput and the impact of aggressive power consumption and negative thermal effects. Many techniques have been proposed in this area, and some of them have been implemented such as the well-known DVFS technique which is used in nearly all modern microprocessors. This paper explores the concepts of multi-core, trending research areas in the field of multi-core processors and then concentrates on power management issues in multi-core architectures. The main objective of this paper is to survey and discuss the current power management techniques. Moreover, it proposes a new technique for power management in multi-core processors based on that survey.

1. Introduction

The evolution of multi-core processors led to the evolution of many research areas. Before the appearance of multi-core processors, the speed of microprocessors increased exponentially over time. More speed requires more transistors. Moore [1] observed that the number of transistors doubles approximately every two years. With the rapid increase in speed, the number of transistors in processors increased in a way that it can't scale to Moore's law anymore as an extremely huge number of transistors switching at very high frequencies means extremely high power consumption. Also, the need for parallelism increased and the instruction level parallelism [2] was not sufficient to provide the demanding parallel applications. So the concept of multi-core was introduced by Olukotun et al. [3], to design more simple cores on a single chip rather than designing a huge complex one. Now all modern microprocessor designs are implemented in a multi-core fashion. Multi-core advantages can be summarized as follows:

- A chip multiprocessor consists of simple-to-design cores
- Simple design leads to more power efficiency
- High system performance in parallel applications where many threads need to run simultaneously

* Corresponding author. Mobile: +20 1000736160.
E-mail addresses: khaled.m.attia@mans.edu.eg (K.M. Attia), melhosseini@mans.edu.eg (M.A. El-Hosseini), h_arafat_ali@mans.edu.eg (H.A. Ali).

Peer review under responsibility of Ain Shams University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.asej.2015.08.010
2090-4479 © 2015 Faculty of Engineering, Ain Shams University. Production and hosting by Elsevier B.V.

Introducing multi-core processors aroused many related areas of research. Dividing code into threads, each can run independently is very important to make use of the power of the multi-core approach. However, not all code can be divided in such a manner. That issue was described by Amdhal in [4] which concludes that maximum speedup is limited by the serial part and that is called the serial bottleneck. Serialized code reduces performance expected by the processor; it also wastes lots of energy. Also, the parallel portion of the code is not completely parallel because of many reasons such as synchronization overhead, load imbalance and resources contention among cores. The serial bottleneck research led to the evolution of asymmetric multi-core processors [5].

The concept of asymmetric multi-core processors implies that the design would include one large core and many small cores. The serial part of the code will be accelerated by moving it to the large core and the parallel part is executed on the small cores. This accelerates both the serial part by using the large core and the parallel part as it will be executed simultaneously on the small cores and the large core to achieve high throughput. Using asymmetric cores can be more energy efficient too. In [5] Mark et al. described how asymmetry can be achieved. They divided it into static and dynamic methods. For static methods, cores may be designed at different frequencies or a more complex core with completely different micro-architecture may be designed. In dynamic methods, frequencies can be boosted dynamically on demand or small cores may be combined to form a dynamic large core and this is described in detail in [6]. Other research topics related to multi-core processors that emerged include the following: power management, memory hierarchies in multi-core processors, the design of interconnection networks in multi-core processors, heterogeneous computing in multi-core processors, reliability issues in multi-core processors and parallel programming techniques. In power management, the main objective is to reach the maximum performance of the processors without exceeding a given total power budget for the chip. There has been lots of research on power management in chip multiprocessors. Here we are going to discuss most of those techniques [7] and some modern works that try to optimize the efficiency of these techniques. In this paper we examine all popular techniques in detail and how they work to minimize performance losses while saving power. We investigate the suitable technique for each case (workloads, power budget available, critical systems) and how to make these techniques even more suitable for their cases.

This paper makes the following contributions:

- Listing almost all the used techniques for power management in multi-core processors, discussing them in terms of advantages and disadvantages (performance loss, power saving, suitable cases) and providing a comparison between them.
- Examining some of the improvements added to each of these techniques to make them even better.
- Proposing a new adaptive control mechanism for power management in asymmetric multi-core processors.
- Suggesting further research to be done in some of the investigated techniques/scenarios.

The rest of this paper is organized as follow: Section 2 introduces the historical improvements in the microprocessor design, explains how we have reached the multicore era and mentions the main issues associated with multicore processors. Section 3 starts to focus on the power management issue, showing the importance of handling such a problem and providing a proper problem formulation. It continues to explain almost all the current techniques used in the power management field in modern processors, showing the advantages and disadvantages of each one and the research done to try to solve each shortage. Section 4 proposes a new mechanism for power management in asymmetric multicore processors. Finally, we conclude in Section 5 by reviewing the most important ideas that were presented in the paper.

2. Background

The performance of microprocessors has increased exponentially over years. Techniques have been devised to achieve parallelism, starting from pipelining, passing by super-scalar architectures and finally the chip multiprocessors or multicore processors. Here we shed light on the various levels of parallelism and how consequent technologies tried to exploit each level.

2.1. Levels of parallelism

Each one of these techniques exploits some levels of parallelism which can be listed as follows:

(1) Instruction level parallelism

In this level, architectures make use of independent instructions (the operands of one instruction do not depend on the result of another one) that exist in the instructions streams to execute them concurrently.

(2) Basic Block level

A block can be considered a set of instructions that end with a branch. Modern architectures were able to exploit this level of parallelism among basic blocks with the help of advanced branch predictors

(3) Loop iterations

Some types of loops work on independent data in each iteration of the loop. So, it is possible in these loops to run different iterations concurrently in superscalar architectures for example.

(4) Tasks

A task signifies an independent function extracted from one application. It can also be called a thread. Software developers have to divide their code into independent threads to make use of this level of parallelism in multiprocessors systems, where each thread can run independently on a dedicated core.

2.2. Advances in processor microarchitecture

Over the years, there have been many trials to exploit better parallelism as shown in Fig. 1; advances in architecture can be viewed as follows: