Microprocessors and Microsystems 32 (2008) 460–473

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Modeling hard real-time systems considering inter-task relations, dynamic voltage scaling and overheads ☆

Eduardo Tavares *, Paulo Maciel, Bruno Silva

Universidade Federal de Pernambuco, Centro de Informática, 50732-970 Recife, Pernambuco, Brazil

A R T I C L E   I N F O

Article history:
Available online 8 August 2008

Keywords:
Hard real-time system
Scheduling
Formal methods
Dynamic voltage scaling
Energy consumption

A B S T R A C T

Dynamic voltage scaling (DVS) has been adopted as an effective technique for reducing energy consumption in embedded systems. Although several scheduling approaches have been developed to address voltage scaling together with stringent timing constraints, inter-task relations have been neglected. This work presents a pre-runtime method for hard real-time systems scheduling considering dynamic voltage scaling, overheads and inter-task relations. The proposed work considers time Petri nets as a formal model in order to provide a basis for precise schedule generation as well as to allow property analysis and verification. Experimental results depict the proposed approach feasibility, in the sense that energy consumption is minimized as well as system constraints are met.

1. Introduction

Whenever designing embedded systems, several constraints, such as size, reliability, energy consumption and timing constraints, may have to be considered to satisfy system requirements. Lately, considerable special attention has been devoted to energy consumption, mainly due to the great expansion of the mobile device market.

During the last decade, DVS (dynamic voltage scaling) has been adopted as one of the most effective techniques for reducing energy consumption in embedded systems. Adjusting CPU supply voltage has great impact on energy consumption, since the consumption is proportional to the square of supply voltage in CMOS microprocessors [11]. However, lowering the supply voltage linearly affects the maximum operating frequency. Therefore, DVS may be seen as a technique for trading-off energy consumption and performance.

When considering hard real-time systems, DVS needs to be adopted with caution, since stringent timing constraints may be affected. In this case, equipment damage or even loss of human lives may occur due to timing constraint violations. Thus, several scheduling approaches, mainly based on runtime techniques, have been devised to cope with DVS in time-critical systems. However, system specifications often either oversimplify tasks' relations such as precedence and exclusion relation or do not consider them at all. Furthermore, overheads, such as preemptions and voltage/frequency switching, are issues that must be considered during schedule generation. Indeed, if overheads are neglected, tasks' constraints may be affected and even the gains obtained with DVS may be significantly reduced [15].

This paper presents a pre-runtime scheduling method for hard real-time systems that considers DVS, inter-task relations and overheads. More specifically, the contributions are: (i) the proposition of a formal model based on time Petri nets (TPN) that provides the basis for precise schedule generation as well as for the verification/analysis of behavioral and structural properties; (ii) the explicit modeling of inter-task relations and overheads (e.g. voltage/frequency switching), so as to allow considering them for the schedule generation; and (iii) a pre-runtime scheduling algorithm that finds out feasible schedules that satisfy timing and energy constraints. Besides, a technique for dealing with dynamic slack times is presented in order to take advantage of new opportunities to further reduce energy consumption during system execution.

One challenge designers have to face when dealing with embedded hard real-time systems is the modeling power of tools. In order to be of practical usability, they might provide means for representing concurrent communicating tasks, synchronization mechanisms and communication primitives as well as they should describe timing constraints and requirements. Furthermore, the availability of precise methods for analysis and verification of systems' representation is a requirement of remarkable importance. Petri nets [16] are a very well suited model for representing real-time embedded systems, since concurrency, synchronization and communication mechanisms are naturally represented. It should

☆ The authors would like to thank the anonymous reviewers for their valuable comments, which greatly improved the quality of the paper.

* Corresponding author. Tel.: +55 81 2126 8430.
E-mail addresses: eagt@cin.ufpe.br (E. Tavares), prmm@cin.ufpe.br (P. Maciel), bs@cin.ufpe.br (B. Silva).

0141-9331/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2008.07.001

also be emphasized the sound mathematical basis related to Petri net analysis methods.

The rest of the paper is organized as follows: Section 2 summarizes related works. Section 3 presents some preliminaries with the purpose of providing a better comprehension of the proposed approach. Section 4 describes the computational model and Section 5 introduces the formal modeling. Section 6 describes the proposed pre-runtime schedule synthesis as well as discussing the respective algorithm complexity. Section 7 presents a technique for dealing with slack times that may appear during system execution. Section 8 describes experimental results, and Section 9 concludes this paper and introduces future works.

2. Related works

Many scheduling methods [1,2,7,10,15,21,26] have been developed to cope with voltage scaling in time-critical systems. Works, such as [2,21], are based on runtime scheduling policies, which can greatly improve energy consumption as shown by their experimental results. Some of them apply a preprocessing for defining an initial voltage for each task before runtime. This can be viewed as a hybrid approach, which mixes runtime and pre-runtime approaches. However, some of these works do not properly tackle overheads related to voltage/frequency switching, preemption, and runtime calculations, and neglect precedence and exclusion relations. A common approach in dealing with runtime overheads is considering them in tasks' worst-case execution cycles (WCEC). Nevertheless, this approach may be too pessimistic, since the total overhead is not known before schedule generation. In this context, [15,1] explicitly take into account overheads related to voltage/frequency switching during scheduling generation without relying on the previous statement. Nevertheless, dispatcher/scheduler and preemption overheads are disregarded. In [10], the authors propose a technique for reducing the impact of preemptions in system energy consumption. Although interesting results are provided, the technique does not consider inter-task relations and assumes CPUs with continuously variable voltage.

In the literature, few works cope with inter-task relations. The work described in [8] proposes a runtime method for dealing with exclusion relations. Nevertheless, precedence relations are neglected and preemption and voltage/frequency switching overheads are adopted in tasks' computation time. In [3], the authors describe a DVS scheduling method for a distributed environment considering precedence relations, but they ignore mutual exclusions and consider the scheduler overhead in tasks' worst-case execution time (WCET). Cortés [4] proposes a scheduling method considering precedence relations, assuming that all tasks are non-preemptable. Besides, the adopted task model assumes tasks with mandatory and optional parts, in the sense that optional parts can be left incomplete in order not to violate timing constraints. In relation to formal methods, some works have been proposed over the years to tackle real-time systems with energy constraints. However, in general, they consider soft timing constraints (e.g. [19]) or adopt fixed-priority scheduling (e.g. [13]). In the latter situation, feasible schedules may not be properly generated when considering arbitrary inter-task relations [25].

As an alternative, this work proposes a pre-runtime scheduling method that considers DVS, inter-task relations and runtime overheads. A formal model based on time Petri nets is adopted to provide a basis for precise schedule generation as well as to allow property analysis and verification.

3. Preliminaries

This section aims at presenting fundamental concepts and a motivational example in order to show the feasibility of the proposed method.

3.1. Problem formulation

Before providing details related to the proposed methodology, it is important to describe its context. The software specification is represented by a set of periodic hard real-time tasks ($\mathcal{T}$) with inter-task relations, such as precedence and mutual exclusion. Additionally, it is considered a processor that can vary its voltage/frequency level within a discrete range (*vff*). It is worth noting the overheads that may occur during system execution, more specifically, voltage/frequency switching, preemption, and runtime calculations (e.g. dispatcher execution). This paper concerns the problem of scheduling those tasks on a DVS-capable processor, such that timing constraints as well as inter-task relations are met and energy consumption is minimized, respecting a given energy constraint ($e_{\max}$). Besides, overheads are considered during scheduling process in order to provide a more realistic system behaviour. Throughout this paper, the term overhead encompasses, as stated previously, voltage/frequency switching, preemption, and dispatcher executions.

3.2. Methodology

Fig. 1 provides an overview of the design methodology for implementing embedded software synthesis, in which the proposed scheduling method is a fundamental activity.

Initially, the designer defines the system specification, which consists of a set of concurrent tasks with their respective constraints, behavioral descriptions, information related to the hardware platform (e.g. voltage/frequency levels and energy consumption) as well as the system energy constraints. A measurement activity may be required whether the designer does not possess the tasks' timing information or the information regarding the hardware energy consumption. Next, the specification is translated into an internal model able to represent concurrent activities, timing information, inter-task relations, such as precedence and mutual exclusion, as well as energy constraints. The adopted internal model is a time Petri net extension, labeled with energy consumption values. After generating the internal model (TPN),

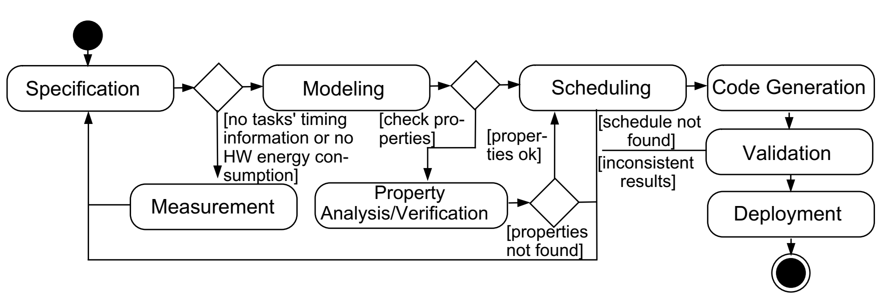

Fig. 1. Methodology activity diagram.