

Applying Multi-Core Model Checking to Hardware-Software Partitioning in Embedded Systems

Alessandro Trindade · Hussama Ismail · Renato Degelo · Edilson Galvao · Lucas Cordeiro

Received: date / Accepted: date

Abstract We present an alternative approach to solve the hardware (HW) and software (SW) partitioning problem, which uses Bounded Model Checking (BMC) based on Satisfiability Modulo Theories (SMT) in conjunction with a multi-core support using Open Multi-Processing. The multi-core SMT-based BMC approach allows initializing many verification instances based on processors cores numbers available to the model checker. Each instance checks for a different optimum value until the optimization problem is satisfied. The goal is to show that multi-core model-checking techniques can be effective, in particular cases, to find the optimal solution of the HW-SW partitioning problem using an SMT-based BMC approach. We compare the experimental results of our proposed approach with Integer Linear Programming and the Genetic Algorithm.

Keywords hardware-software co-design · hardware-software partitioning · optimization · model checking · multi-core · OpenMP

1 Introduction

Nowadays, with the strong development of embedded systems, the design phase plays an important role. At early stages, the design is split into separated flows: hardware and software. Consequently, the partitioning decision process, which deals with the decisions upon which parts of the application have to be designed in

hardware (HW) and which in software (SW), must be supported by any well-structured methodology. If not, this leads to a number of issues (design flow interruptions, redesigns, and undesired iterations) which affects the overall development process, the quality and the lifecycle of the final system. Starting at the 1990s, intensive research was performed, and several approaches proposed, as shown in [1] and [2]. In any HW and SW design of complex systems, more time is spent on verification than on construction [3]. Formal methods based on model checking offer great potential to obtain a more effective and faster verification in the design process. Programs may be viewed as mathematical objects with behavior that is, in principle, well determined. This makes it possible to specify programs using mathematical logic, which constitutes the intended (correct) behavior. Then, one can try to give a formal proof or otherwise establish that the program meets its specification [4]. Research in formal methods has led to the development of very promising verification techniques, which facilitate the early detection of errors. Model-based verification techniques use models that describe the possible system behavior in a mathematically precise and unambiguous manner. The system models are accompanied by algorithms that systematically explore all the states of the system model. In [5] and [6] was shown that it is possible to use Bounded Model Checking (BMC) based on Satisfiability Modulo Theories (SMT) to perform HW-SW partitioning in embedded systems. The present work extends those studies since there is a substantial improvement in terms of the genetic algorithm and the SMT-based verification method, which has been extended with a multi-core architecture. Multi-core processors have been used in all segments of industry to implement high-performance computing [7]. In particular, hardware platforms, to-

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

gether with multi-processing platforms, have allowed verification algorithms to distribute tasks executions across multiple processors, which generate an increase in performance if compared to single-core solution. However, most verification algorithms still disregard the limitations of the CMOS technology, which limits the increase of the chips frequency after it reaches 4 GHz. Here, we exploit the availability of multi-core processors. In particular, a multi-core SMT-based BMC method is applied to the HW-SW partitioning and then is compared to the results with classical integer linear programming (ILP) and genetic algorithm (GA) using a multi-core tool as well. To the best of our knowledge, this is the first work to use a multi-core SMT-based verification to solve a HW-SW partitioning problem in embedded systems. We implement our ideas with the Efficient SMT-based Bounded Model Checker (ESBMC) tool [14]. As its main contribution, this paper shows that it is possible to take advantage of an SMT-based BMC tool in a multi-core architecture to solve optimization problems. This paper is organized as follows: Section II gives a background on optimization, model checking, and multi-core support with Open MultiProcessing. Section III describes informal and formal mathematical modeling. Section IV describes briefly the binary integer programming and GA algorithms. The SMT-based BMC method is presented in Section V. Section VI presents the experimental evaluation. Section VII discusses related work. Section VIII presents the conclusion and future work.

2 BACKGROUND

2.1 Optimization

Optimization is the act of obtaining the best result (i.e., the optimal solution) under given circumstances [9]. In the design, construction, and maintenance of any engineering system, engineers have to make many technological and managerial decisions at several stages. The ultimate goal of all such decisions is either to minimize the effort required or to maximize the desired benefit. Because the effort required or the benefit desired in any practical situation can be expressed as a function of certain decision variables, optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function [9]. There is no single method available for solving all optimization problems efficiently [9]. The most known technique is linear programming, which is a method applicable for the solution of problems in which the objective function and the constraints appear as linear functions of

the decision variables. A particular case of linear programming is ILP, in which the variables can assume just integer values. Eq. (1) shows a typical linear programming problem, where \mathbf{c} and \mathbf{b} are vectors or matrixes that describe the constraints.

2.2 Optimization with Vz

2.3 Bounded Model Checking with ESBMC

2.4 Multi-core ESBMC with OpenMP

2.5 Multi-core ESBMC with OpenMP using Binary Search

3 Mathematical modeling

The mathematical modeling was taken from [1], [2].

3.1 Informal Model (or Assumptions)

3.2 Formal Model

4 Partitioning problem using ILP-based, Genetic Algorithms

The ILP and GA were taken from [5] and [6]. Both use slack variables in order to be possible to represent the constraints and to use commercial tools. However, GA had improvements from the parameters of related studies in order to increase the solution accuracy without producing timeout. The tuning was performed by empirical tests and resulted in changing of three parameters, which are passed to function `ga` of MATLAB [18]: the population size was set from 300 to 500, the Elite count changed from 2 (default value) to 50, and the number of Generations changed from $100 * \text{NumberOfVariables}$ (default) to 75.

5 Analysis of the partitioning problem using ESBMC

6 Analysis of the partitioning problem using vZ

7 Experimental Evaluation

8 Related Work

9 Conclusions

10 References