Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Electrical Engineering 30 (2004) 471–489

Computers and Electrical Engineering

www.elsevier.com/locate/compeleceng

Hardware-software co-synthesis of hard real-time systems with reconfigurable FPGAs

Faridah M. Ali *, A. Shoba Das

Department of Computer Engineering, Kuwait University, P.O. Box: 5969, Safat 13060, Kuwait

Received 30 October 2002; received in revised form 1 April 2004; accepted 14 September 2004
Available online 30 December 2004

Abstract

Real-time systems cover a wide application domain. This paper presents an efficient heuristic algorithm for enforcing the schedulability of aperiodic hard real-time tasks arriving simultaneously with precedence constraints and individual deadlines. The proposed co-synthesis algorithm integrates partitioning and non-preemptive scheduling. Reconfigurable FPGAs are incrementally added when schedulability suffers in a uniprocessor system. Initially, a schedule that minimizes the maximum lateness and satisfies the precedence constraints is made. If individual timing constraints are not met in this schedule, some tasks are selected and transferred to dynamically reconfigured FPGAs. The proposed algorithm was implemented and tested with a large number of task graphs with task size as high as 700 nodes. The algorithm could not only achieve schedulability but also could reduce the total completion time of the task graph. Moreover, incremental addition of reconfigurable FPGAs yielded a cost effective solution.

Keywords: Hardware–software co-synthesis; Hard real-time; Reconfigurable FPGA; Precedence constraint

* Corresponding author. Tel.: +965 48 11188x5826; fax: +965 48 39461.
E-mail address: faridah@eng.kuniv.edu.kw (F.M. Ali).

0045-7906/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2004.09.001

1. Introduction

Hard real-time systems are characterized by strict constraints on task completion time, referred to as deadline. In such systems, which are found in flight control, telecom, defense control, nuclear power plants, automotive electronics and process control etc., missing the deadline may lead to catastrophic sequences. If software implementation of the tasks cannot fulfill the timing requirement of the real-time system, concurrent execution of part of the tasks by hardware components can be used to remedy this problem.

An embedded system is composed of hardware (HW) and software (SW) computing components executing concurrently and cooperatively in order to benefit from the strength of each technology. The software component consists of one or more general purpose programmable processors, while the Field Programmable Gate Arrays (FPGAs) or the Application Specific Integrated Circuits (ASICs) are the hardware components. In the last decade, a systematic system design approach referred to as HW/SW co-design has been receiving a great attention in the research community. It is only recently that some attention has been given to the co-design of embedded real-time systems [3,8,9]. Moreover, timing constraints are well defined in real-time systems which make them more suitable for HW/SW co-design.

The co-design process includes three main activities, namely, (1) allocation of appropriate electronic components such as processors, FPGAs/ASICs, buses, memory etc., (2) partitioning the task set to be implemented among software and hardware component, and (3) scheduling the tasks on the electronic components [1,2]. Due to recent improvements which contributed to lower prices, higher gate counts and reconfiguration speed, embedded systems with dynamically reconfigurable FPGAs have become a reality, which provides a great deal of functional flexibility and execution speed [12]. Most of the existing approaches perform partitioning and scheduling in two separate steps. The tasks are first partitioned and then scheduling is performed to check the feasibility and the performance of the system. Integer programming, greedy heuristics and graph partitioning have been used to solve the partitioning problem in single CPU systems [3–6].

In the last few years, developing an integrated approach which combines scheduling and partitioning became an active area of research [7–9,16,18,19]. In this approach, tasks are initially assumed to be implemented in software and then selected tasks are moved from software to hardware until the specified constraints are met. The scheduling algorithms which determine the order in which the tasks are executed are classified based on the assumptions made on the system and on the tasks. Liu and Wong [7] used the integrated approach for partitioning and scheduling tasks with precedence constraint and uniform execution time. Two homogeneous CPUs and k hardware components were used to achieve reduction in completion time and hardware cost. Shin and Chico [9] reduced the execution time of tasks by assigning part or whole of the task from software to a coprocessor which is a hardware implementation of code segments of tasks. For optimal partitioning and comparing different target architectures, schedulability measurement was used as the criteria by Axelsson [8]. In both approaches [8,9], initial scheduling in software is done by rate monotonic scheduling algorithm, which is optimal for the class of tasks which are independent and periodic, where task dependency is not a criterion.

HW/SW co-synthesis with FPGAs as the hardware component has been gaining attention of researchers [11,13,18,19]. Dick and Jha [13] introduced dynamically configurable FPGAs for HW/SW co-synthesis with an overall objective of achieving low cost systems. Jeong et al. [11],