Scheduling of Conditional Process Graphs for the Synthesis of Embedded Systems

Petru Eles\textsuperscript{1,2}, Krzysztof Kuchcinski\textsuperscript{1}, Zebo Peng\textsuperscript{1}, Alexa Doboli\textsuperscript{2}, and Paul Pop\textsuperscript{2}

Abstract We present an approach to process scheduling based on an abstract graph representation which captures both data-flow and the flow of control. Target architectures consist of several processors, ASICs and shared busses. We have developed a heuristic which generates a schedule table so that the worst case delay is minimized. Several experiments demonstrate the efficiency of the approach.

1 Introduction

In this paper we concentrate on process scheduling for systems consisting of communicating processes implemented on multiple processors and dedicated hardware components. In such a system in which several processes communicate with each other and share resources, scheduling is a factor with a decisive influence on the performance of the system and on the way it meets its timing constraints. Thus, process scheduling has not only to be performed for the synthesis of the final system, but also as part of the performance estimation task.

Optimal scheduling, in even simpler contexts than that presented above, has been proven to be an NP complete problem \cite{13}. In our approach, we assume that some processes can be activated if certain conditions, computed by previously executed processes, are fulfilled. Thus, process scheduling is further complicated since at a given activation of the system, only a certain subset of the total amount of processes is executed and this subset differs from one activation to the other. This is an important contribution of our approach because we capture both the flow of data and that of control at the process level, which allows an accurate and direct modeling of a wide range of applications.

Performance estimation at the process level has been well studied in the past years \cite{10, 12}. Starting from estimated execution times of single processes, performance estimation and scheduling of a system containing several processes can

---

\textsuperscript{1}Department of Computer and Information Science Linköping University Sweden
\textsuperscript{2}Computer Science and Engineering Department Technical University of Timisoara Romania
be performed. In [14] performance estimation is based on a preemptive scheduling strategy with static priorities using rate-monotonic-analysis. In [11] scheduling and partitioning of processes, and allocation of system components are formulated as a mixed integer linear programming problem while the solution proposed in [8] is based on constraint logic programming. Several research groups consider hardware/software architectures consisting of a single programmable processor and an ASIC. Under these circumstances deriving a static schedule for the software component practically means the linearization of a dataflow graph [2, 6].

Static scheduling of a set of data-dependent software processes on a multiprocessor architecture has been intensively researched [3, 7, 9]. An essential assumption in these approaches is that a (fixed or unlimited) number of identical processors are available to which processes are progressively assigned as the static schedule is elaborated. Such an assumption is not acceptable for distributed embedded systems which are typically heterogeneous.

In our approach we consider embedded systems specified as interacting processes which have been mapped on an architecture consisting of several processors and dedicated hardware components connected by shared busses. Process interaction in our model is not only in terms of dataflow but also captures the flow of control under the form of conditional selection. Considering a nonpreemptive execution environment we statically generate a schedule table for processes and derive a worst case delay which is guaranteed under any conditions.

The paper is divided into 7 sections. In Section 2 we formulate our basic assumptions and introduce the graph-based model which is used for system representation. The schedule table and the general scheduling strategy are presented in Sections 3 and 4. The algorithm for generation of the schedule table is presented in Section 5. Section 6 describes the experimental evaluation and Section 7 presents our conclusions.

2 Problem Formulation and the Conditional Process Graph

We consider a generic architecture consisting of programmable processors and application specific hardware processors (ASICs) connected through several busses. These busses can be shared by several communication channels connecting processes assigned to different processors. Only one process can be executed at a time by a programmable processor while a hardware processor can execute processes in parallel. Processes on different processors can be executed in parallel. Only one data transfer can be performed by a bus at a given moment. Computation and data transfer on several busses can overlap.

In [4] we presented algorithms for automatic hardware/software partitioning based on iterative improvement heuristics. The problem we are discussing in this paper concerns performance estimation of a given design alternative and scheduling of processes and communications. Thus, we assume that each process is assigned to a (programmable or hardware) processor and each communication channel which connects processes