Scheduling, Mapping and Communication Synthesis for Distributed Real-Time Systems

Paul Pop
Dept. of Computer and Information Science,
Linköping University, Sweden
http://www.ida.liu.se/~paupo

1. Introduction

This thesis concentrates on aspects related to the scheduling, mapping and synthesis of distributed embedded real-time systems consisting of programmable processors and application specific hardware components.

We have investigated the impact of particular communication infrastructures and protocols on the overall performance and how the requirements of such an infrastructure have to be considered for process and communication scheduling. Not only have particularities of the underlying architecture to be considered during scheduling, but the parameters of the communication protocol should also be adapted to fit the particular embedded application.

Moreover, we consider an incremental design process that starts from an already existing system running a set of applications. For our mapping strategy, we are interested to implement new functionality so that the already running applications are disturbed as little as possible and there is a good chance that, later, new functionality can easily be added to the resulted system.

The approaches to scheduling, mapping and system synthesis are based on an abstract graph representation which captures, at process level, both dataflow and the flow of control.

2. Thesis Topics

We model real-time systems using a set of *conditional process graphs*, and we consider a generic architecture consisting of *programmable processors* and application specific *hardware processors* connected through several *buses*. As the communication infrastructure for our distributed real-time system we consider the *time-triggered protocol* (TTP). TTP is well suited for safety critical distributed real-time embedded systems and represents one of the emerging standards for several application areas like, for example, automotive electronics. In our approach, process scheduling can use either a *non-preemptive static cyclic* or a *static priority preemptive* scheduling approach while the bus communication is statically scheduled according to the TTP.

The scheduling problems discussed in this thesis concern the performance estimation of a given design alternative, where mapping has already been decided. Thus, we assume that each process has been assigned to a (programmable or hardware) processor and each communication channel which connects processes assigned to different processors has been assigned to a bus. In this context, the goals of scheduling of processes and communication are the following:

- derive a schedulability analysis for systems with both control and data dependencies,
- derive a schedulability analysis for systems where the communication takes place using the time-triggered protocol,
- determine an as small as possible worst case delay by which the system completes its execution and generate the static schedule such that this delay is guaranteed, and
- determine the parameters of the communication protocol so that the overall system performance is optimized and, thus, the imposed time constraints can be satisfied.

A characteristic of research efforts concerning the codesign of embedded systems is that authors concentrate on the design, from scratch, of a new system optimized for a particular application. For many application areas, however, such a situation is extremely uncommon and only rarely appears in design practice. It is much more likely that one has to start from an already existing system running a certain application and the design problem is to implement new functionality (including also upgrades to the existing one) on this system. In such a context it is very important to make as few as possible modifications to the already running applications. The main reason for this is to avoid unnecessarily large design and testing times. However, this is not the only aspect to be considered. Such an incremental design process, in which a design is periodically upgraded with new features, is going through several iterations. Therefore, after new functionality has been implemented, the resulting system has to be structured such that additional functionality, later to be mapped, can easily be accommodated.

Thus, in this thesis we perform mapping and scheduling of new functionality so that timing and other design constraints are satisfied and:

- already running applications are disturbed as little as possible;
- there is a good chance that new functionality can, later, easily be mapped on the resulted system.

In this context, we have:

- introduced two design criteria with their corresponding metrics that drive our mapping strategy to solutions supporting an incremental design process, and
- proposed several algorithms to produce a minimal subset of applications which have to be remapped and scheduled in order to implement the new functionality.

3. Contributions

In this thesis, an embedded system is viewed as a set of interacting processes mapped on an architecture consisting of several programmable processors and ASICs interconnected by a communication channel.

Process interaction is not only in terms of dataflow but also captures the flow of control, since some processes can be activated depending on conditions computed by previously executed processes.

We have considered both the non-preemptive static cyclic scheduling and the static priority preemptive scheduling approaches for the scheduling of processes, while the communications are statically scheduled according to the TTP.

The scheduling strategies are based on a realistic communication model and execution environment. We take into consideration the overheads due to communications and to the execution environment and consider the requirements of the communication protocol during the scheduling process.

The mapping strategies have been developed within an incremental design process that tries to minimize the modifications performed to the existing applications, while at the same time supporting the addition, in the future, of new applications.

Our approaches have considered at process level a non-preemptive static cyclic or a static priority preemptive scheduling approach while the bus communication was statically scheduled according to the TTP.